



API Interface

Payment interface document



Document version: 1.0

Interface version: 1.0

Table of Contents

1	Introduction.....	2
1.1	Payment mode.....	2
2	Program overview.....	2
2.1	Interface call sequence diagram.....	2
3	Data Format.....	3
3.1	submit data.....	3
3.2	XML data format.....	3
4	digital signature.....	4
4.1	Signature original string.....	4
4.2	Signature algorithm.....	5
4.2.1	SHA256 signature.....	5
4.2.2	RSA signature.....	6
5	Payment interface.....	9
5.1	Payment interface.....	9
5.2	Order query interface.....	12
5.3	Refund query interface.....	15
5.4	Refund interface.....	18
5.5	Cancel interface.....	21
5.6	Scan code notification interface.....	24
5.7	Binding card interface.....	27
5.8	Unbind interface.....	29
6	Precautions.....	31

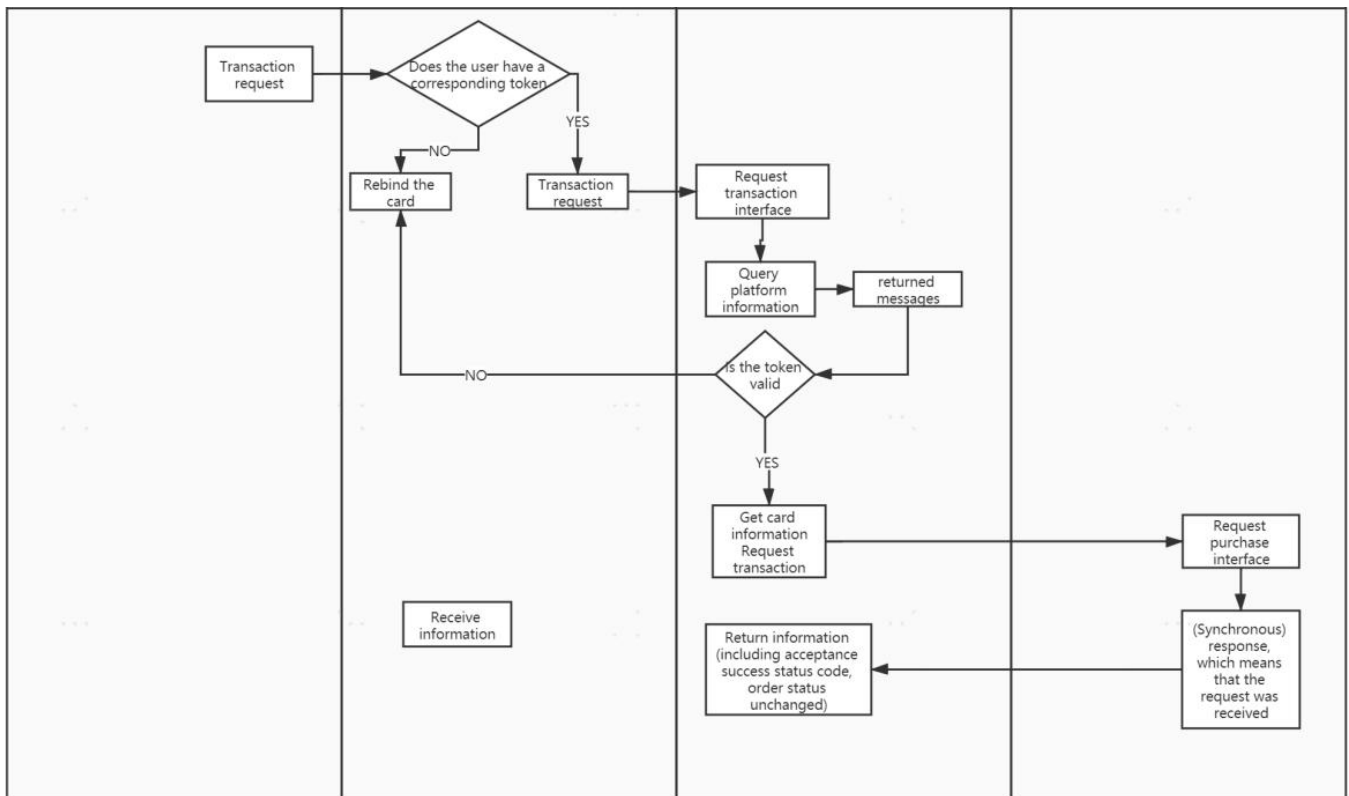
1 Introduction

1.1 Payment mode

Scan code payment is a mode in which merchant systems generate payment QR codes according to wallet payment protocols such as WeChat, Alipay, and UnionPay, and users then use WeChat, Alipay, and support UnionPay payment apps to "scan and scan" to complete the payment. This mode is suitable for PC website payment, physical store single product or order payment, media advertising payment and other scenarios.

2 Program overview

2.1 Interface call sequence diagram



3 Data Format

3.1 submit data

Using the HTTPS standard POST protocol, in order to ensure that the receiver receives the data correctly, the transmitted data must be signed. (Take the WeChat scan code interface as an example, please fill in the specific interface type for the specific service field transmission reference)

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

3.2 XML data format

Using standard XML protocol, all parameters only exist in the first-level node, do not use multi-level node nesting, and need to be included in CDATA

Protocol-level error return:

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

Return data correctly:

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

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

Business-level error return:

```
<xml>
<status>0</status>
<message><![CDATA[OK]]></message>
<appid><![CDATA[wx2421b1c4370ec43b]]></appid>
<mch_id><![CDATA[10000100]]></mch_id>
<device_info><![CDATA[1000]]></device_info>
<nonce_str><![CDATA[sthBJ9QyUG6vkrjJ]]></nonce_str>
<sign><![CDATA[6277A96D7875D4FF23AA7B6A4C3046AB]]></sign>
<result_code>1</result_code>
<err_code><![CDATA[AUTHCODE_EXPIRE]]></err_code>
<err_code_des><![CDATA[The QR code has expired, please refresh and try again]]></err_code_des>
</xml>
```

4 digital signature

In order to ensure the authenticity and integrity of the data during the data transmission process, it is necessary to perform signature verification after receiving the signed data.

4.1 Signature original string

Whether it is a request or a response, the original signature string is assembled into a string in the following way:

1. Except for the sign field, all parameters are sorted according to the ascii code of the field name from small to large and then spliced using the format of QueryString (ie key1=value1&key2=value2...). Null values are not passed and do not participate in the signature string.
2. In the original signature string, both the field name and the field value adopt the original value, without URL Encode.
3. The response or notification message returned by the platform may add parameters due to the upgrade. Please pay attention to allow this situation when verifying the response signature.

```
<xml>
<body><![CDATA[TestPay]]></body>
<device_info><![CDATA[100]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[127520000042]]></mch_id>
<nonce_str><![CDATA[2209196862]]></nonce_str>
<notify_url><![CDATA[https://int.payable.lk:8088/mposcbc/callback]]></notify_url>
<out_trade_no><![CDATA[127590000128]]></out_trade_no>
<service><![CDATA[pay.weixin.native.intl]]></service>
<sign><![CDATA[84F40D5FA4BD7FE3BE9A0A88F564D789]]></sign>
<total_fee><![CDATA[250]]></total_fee>
</xml>
```

The correct order of signature fields is:

```
body=TestPay&device_info=100&mch_create_ip=127.0.0.1&mch_id=127520000042&nonce_str=2209196862&notify_url=https://int.payable.lk:8088/mposcbc/callback&out_trade_no=127590000128&service=pay.weixin.native.intl&total_fee=250
```

4.2 Signature algorithm

4.2.1 SHA256 signature

SHA256 signature calculation formula:

sign = SHA256 (original string &key=merchant key). toUpperCase

Such as: assuming that the following are the XML incoming parameters

```
<xml>
<auth_code>135187250012923035</auth_code>
<body>test</body>
<charset>UTF-8</charset>
<mch_create_ip>127.0.0.1</mch_create_ip>
<mch_id>127530000052</mch_id>
<nonce_str>1542940680925</nonce_str>
<out_trade_no>1542940643087</out_trade_no>
<service>unified.trade.micropay</service>
<sign>10F2F6DC0D5E008B967CC3C86FC58179686B4EE42F4F68B4A7668501B6030C29</sign>
<sign_type>SHA256</sign_type>
<total_fee>2</total_fee>
<version>2.0</version>
</xml>
```

Suppose the merchant key is: 18e0a2ad5d5571af14b855fcf33091f4

i: The string1 after the URL key value is sorted lexicographically after the process of a is:

```
auth_code=135187250012923035&body=test&charset=UTF-8&mch_create_ip=127.0.0.1&mch_id=127530000052
&nonce_str=1542940680925&out_trade_no=1542940643087&service=unified.trade.micropay&total_fee=1
```

ii: After the process of b, the sign obtained is:

```
sign
=SHA256(string1&key=18e0a2ad5d5571af14b855fcf33091f4).toUpperCase
=SHA256(auth_code=135187250012923035&body=test&charset=UTF-8&mch_create_ip=127.0.0.1&mch_id=1275
30000052&nonce_str=1542940680925&out_trade_no=1542940643087&service=
unified.trade.micropay&total_fee=1&key=
18e0a2ad5d5571af14b855fcf33091f4).toUpperCase()
="10F2F6DC0D5E008B967CC3C86FC58179686B4EE42F4F68B4A7668501B6030C29"
```

4.2.2 RSA signature

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

Note: The encoding character set specified when the character string is converted into a byte stream when signing should be consistent with 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 to unlock the response.

RSA signature calculation formula:

$$\text{sign} = \text{RSA}(\text{original string} \& \text{key} = \text{merchant RSA private key}). \text{toUpperCase}$$

Suppose the following are the XML incoming parameters:

```
<xml>
<out_trade_no><![CDATA[6057113230875088]]></out_trade_no>
<nonce_str><![CDATA[wNzpaD0sN17KI80yBQwINNhf0IeNqap]]></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 purchase]]></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.alipay.native]]></service>
<sign>
<![CDATA[SE008JDir0uw0uBy8d48SmdfG37PyGvhtqu8pDT00DpdNkgkLuegNsb6SaL/dEfzu035bwAVwiKSc9m9xBKohGJ
EMtzRm3tLNNpQ0BDpL1YNWQkr0JN3JAjy6wk1icSmfbjXgEvWCcx17Mue59NfI10JRR3MgFg/ySYq2cT4U/o6WJisxtIbre0
ZPM66WL815gAzosJW3Gwr+B8fkRyOckF4w64i0TM20xoSHGEa8w1utOFik5wvxyPHb/JcjhRyrBnb2LFZy5Rq3XzZLW6FaK1
gunWnjK5+4NiNuHDgm7CKZPp8Bp
hA/qnJtAMGfexEZ8J4z9ktyywDK8b8VWJ5ow==]]>
</sign>
</xml>
```

Assume that the merchant's RSA private key is:

```
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEp08nMyk+xuYSatHwnU4//
m47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64RJraOrZq123dvPW4v
VomMRgRcoP0n0YVWp+M6T5PaFgE4M8dh41MZz57gVwOodd08F99Z92f3QgZtEjI+/EXvMenXxb/aRoFnt+Wdk2ELJ6MIP0d9
UU5v3WgLUuNv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5Y1Ry2EIN504cuJYVKU1wOSZgq9oJCynGR0aPuQW
x58IHxEtAgMBAECggEAHfEFd8qm2PTE21Tavec7F+TcgD84IUAz0dZnURtx6YIOoZ5+LH/zVG6juYLJU/Oo5RPac+iMVS68
u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQuXN8PkDc0xqDmZ2kPjz14PZvBZRE0abdug+tMatGzpGAuJzrWcB/NOoVivrXp9Pn
Oqfo/Y5nxmp0FCImJppIS3AL1pftNtQZ09G15CPHDYtpUbXPtD2MjW40LxKuPRoHSwUgo6LW9XSwNXfcuK+1bZLL0Bh1WD9
IV/+yCEUEb1N87yxxfhpQFaAhXj5W+B3YsMOZuK93+XMOpYmw8EpUDM0bOnvbw0NSHURV2RUAQKBgQDTojlnNS1e7+tjPzFt
OhGpjluCBPAEIEhAcnPgD80bEiuJxMLcNgaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsfr3CAAu7CadQ49TW+SovAvcy9AZuSVVI
wynu6QdYgFyPke1LZYAEq5k+mB1Vh5q0RoxMNA5pGYK8g+4MmmsJi7X7QKBgQDAunCOqIiH128bs/1VRIhDpzuRW5Qr/SRb
02saVg5RSHn0/nGT20uxSTTkc8yrx7qd9SmAxX15kR238DhMOQOnRBomldmVtAJuJgrdQyt0wXfeVQqshqCUaE/xhEbpSCd
bPSZbKZZdp1V0y605vXIhxw+1qAvXLcxw46s3R92QQKBgQC1Q+eJywkVPDILHMwSSehwvThufkCYWYubbcVDowp0e5AMoZid
tNju7MNjg2rLHTsCx/kBzOr+7THNw14R7kTiEmg09c0+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpGvtIGB
Ch6BW5JZcSImT+ykIoYfvDRuQKBgCgwOHxnBGFf0RoLxE3dhpSk8LT05cbueIBVuZW6UC3+8PeK82AjIbLMUy04QHupoG6D
yu3BP/1r10jd3L94PBzLBLD7Gm4vJTqWODknYo5sMXS1JrnofcKjBv7nbHXZTx3EtJSxpVa0dpcA/HpsCuCP3AH2e1yk9sZ3
wu61BYSBAoGACYM60j1CVRNSZxUNRgiwFwzS69qIleezPc7xQEganpVBI9SZcTnp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su
3ruNqXvg1XiUPbH6A6nxd5B3QX0mS9+wDm60NysPLRdKbfF00mdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLIOB7gBE=
```

i: The string1 after the URL key value is sorted lexicographically after the process of a is:

```
attach=Additional information&body=Test purchase product
&mch_create_ip=127.0.0.1&mch_id=102532336411&nonce_str=wNzpaD0sN17KI80yBQwINNhfM0IeNqapnotifi_ur
l=http://www.baidu.cn/notify.aspx&out_trade_no=6057113230875088&service=pay.alipay.native&sign_t
ype=RSA_1_256&total_fee=1&version=1.0
```

ii: After the process of b, the sign obtained is:

```
sign
=RSA(string1, RSA private key of the merchant)==RSA(attach=additional information&body=test
purchased product
&mch_create_ip=127.0.0.1&mch_id=102532336411&nonce_str=wNzpaD0sN17KI80yBQwINNhfM0IeNqapnotifi_ur
l=http://www.baidu.cn/notify.aspx&out_trade_no=6057113230875088&service=pay.alipay.native&sign_t
ype=RSA_1_256&total_fee=1&version=1.0)
=MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEp08nMyk+xuYSatHwnU4//
/m47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64RJraOrZq123dvPW4
```


vVomMRgRcoPOn0YWVp+M6T5PaFgE4M8dh41MZz57gVwOdd08F99Z92f3QgZtEjI+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d
9UU5v3WgLuuNv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5Y1Ry2EIN504cuJYVKU1wOSZgq9oJCynGR0aPuQ
Wx58IHxEtAgMBAECggEAHfEFd8qm2PTE21TAVec7F+TcgD84IUAz0dZnURtx6YIOoZ5+LH/zVG6juYLJU/Oo5RPAc+iMVS6
8u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQuXN8PkDc0xqDmZ2kPjz14PZvBZRE0abdug+tMatGzpGAuJzrWcB/NOoVIvrXp9P
n0qfo/Y5nxmp0FCImJppIS3AL1pftNtQZo9G15CPHDYtpUbXPtD2MjjW40LxKuPRoHSwUgo6LW9XSwNXfcuK+1bzLLOBh1WD
9IV/+yCEUEb1N87yxxfhpQFAhXj5W+B3YsMOZuK93+XM0pYmw8EpUDM0b0nvwb0NSHUrV2RUAQKBgQDTTojlnNS1e7+tjPzF
tOhGPj1uCBPAEIEHAcnPgd80bEiujxMLCnGaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsfr3CAAu7CadQ49TW+SovAvciy9AZuSVV
Iwynu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNAA5pGYKg8+4MmmsJi7X7QKBgQDAunCOqIiH128bs/1VRIhDpzuRW5Qr/SR
b02saVg5RSHn0/nGT20uxSTTkc8yrx7qd9SmAxX15kR238DhMOQOnRBomldmVtAJuJgrdQyt0wXfeQVQqshqCUAE/xhEbpSC
dbPSZbKZZdp1V0y605vXIhxw+1qAvXLcxw46s3R92QKKBgQC1Q+e jywKVPDILHMwSsehvwThufkCYWYUbbcVDowp0e5AMoZi
dtNju7MNjg2rLHTsCx/kBz0r+7THNw14R7kTiEmg09c0+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvTIG
BCh6BW5JZcSImt+ykIOoYfvDRuQKBgCgwoHxnBGFf0RoLxE3dhpSk8LT05cbueIBVuZW6UC3+8PeK82AjiBLMUy04QHupoG6
Dyu3BP/1r10jd3L94PBzLBD7Gm4vJTqWODknYo5sMXS1JrnofcKjBv7nbHXZTx3EtJSxpVa0dpcA/HpsCuCP3AH2e1yk9sZ
3wu61BYSBAoGACYM60j1CVRNSzxUNRgiwfwzS69qIleezPc7xQEganpVBI9SZcTNp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6S
u3ruNqvxvg1XiUPbH6A6nxd5B3QX0mS9+wDm60NysPLRdKbFF00mdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLI0B7gBE=")= SE
008JDir0uwOuBy8d48SmdfG37PyGvhtqu8pDT00DpdNkgkLuegNsb6SaL/dEfzu035bwAVwiKSc9m9xBKohGJEMtzRm3tLNN
pQ0BDpL1YNWQkr0JN3JAjy6wk1icSmfbjXgEvWCcx17MUe59NfI10JRR3MgFg/ySYq2cT4U/o6WJisxtIbre0ZPM66WL815g
AzosJW3Gwr+B8fkRyOckF4w64i0TM20xoSHGEa8w1utOFik5wvxyPhB/JcjhRyrBnb2LFZy5Rq3XzZLW6FaK1gunWnjK5+4N
iNuHDgm7CKZPp8BphA/qnJtAMGfexEZ8J4z9ktyywdK8b8VWJ5ow==

WeChat signature verification process:

Please use this tool to verify: https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=20_1

1. Please open the URL https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=20_1, and enter xml in the "XML source string";
2. Please put your signature key into "merchant key";
3. Click the "Verify Signature" button; 4. We can get the correct Sing key in the "New Sing Value"

5 Payment interface

5.1 Payment interface

Pay attention to the interface

1. Due to the characteristics of the payment channel, after initiating a payment, you need to initiate an inquiry or wait for the notification result to determine the payment result.

Note: payment can be initiated only after the card is tied

Request type: POST

Request format: XML

Request parameters: as follows

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

Field Name	Required	Type	Description
service	YES	String(32)	Interface Type: pay.expresspay.pay
version	No	String(8)	Version number. default value : 2. 0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	SHA256 : SHA256; RSA : RSA_1_256
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
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	Yes	String(32)	Specifies a Terminal device id.
device_type	No	String(32)	Equipment type
body	Yes	String(127)	Description of merchants' goods.
attach	No	String(127)	Merchant additional information.
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
time_start	No	String(14)	Order created date. Format : yyyyMMddHHmmss. i.e.20091225091010. GMT+8 Beijing Time
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 the section 4 'Digital Signature'.
channel_type	Yes	String	INTERNET MOBILE Payment in app transaction 08
Token_id	Yes	String	Token_id
phone_no	No	String	phone numbereg: 852-11112222
cvn2	No	String(3)	Card cvn2

Example

```
<xml><body><![CDATA[Test product]]></body>
<channel_type><![CDATA[INTERNET]]></channel_type>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608383533]]></nonce_str>
<notify_url><![CDATA[www.baidu.com]]></notify_url>
<out_trade_no><![CDATA[1608383533]]></out_trade_no>
<service><![CDATA[pay.expresspay.pay]]></service>
<sign><![CDATA[DE3A9D99534203D64D03E351B6D453DC]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<token_id><![CDATA[100550002127Cz95tz8]]></token_id>
<total_fee><![CDATA[100]]></total_fee>
</xml>
```

Return parameters:

Field Name	Required	Type	Description
version	No	String (8)	Version number. The default value of version is 2.0.
charset	Yes	String (8)	Character set, value: UTF-8
sign_type	Yes	String (16)	MD5 signature: MD5;
status	Yes	String (16)	SHA256 signature: SHA256
message	No	String (128)	RSA signature: RSA_1_256

The following fields are returned when status is 0

result_code	Yes	String (16)	0 means success, not 0, processing or failure, please call the query interface to get the final status
mch_id	Yes	String (32)	Merchant' s unique identification code, assigned by the platform

device_info	No	String(32)	Terminal device number
nonce_str	Yes	String(32)	Random string, no longer than 32 bits
err_code	No	String(32)	error code
err_msg	No	String (128)	Error message
sign	Yes	String(344)	Signature result, see "Chapter 4 Signature Rules" for details
trade_type	Yes	String(32)	pay.expresspay.pay
pay_result	Yes	Int	Payment result: 0—success; others—processing or failure, please call the query interface to get the final status
transaction_id	Yes	String(32)	Platform transaction number
out_transaction_id	Yes	String(32)	Third-party order number
out_trade_no	Yes	String(32)	The order number in the merchant system, 5 to 32 characters, can only contain alphanumeric or underscores, is case sensitive, to ensure that it is unique in the merchant system
total_fee	Yes	Int	The total amount of the order, the currency is the channel currency
fee_type	No	String(8)	Order channel currency, E.g AUD
attach	No	String(127)	Merchant data package, returned as is
bank_type	No	String(16)	Payment Types
time_end	Yes	String(14)	The payment completion time, in the format yyyyMMddhhmmss, such as 20091227091010 at 9:10:10 on December 27, 2009. The time zone is GMT+8 beijing.
uuid	No	String(64)	Order uuid
local_fee_type	No	Int	Merchant price currency, E.g AUD
local_total_fee	No	String(16)	Merchant bid amount
order_fee	Yes	Int	Order amount (currency is merchant' s price currency)
need_query	No	String	Do you need to call the query interface to determine the order status Y: Required N: Not required
accepted	Yes	String	Indicates whether the request is accepted successfully, 0: success 1: failure

Example

```
<xml><bank_type><![CDATA[07]]></bank_type>
<charset><![CDATA[UTF-8]]></charset>
<fee_type><![CDATA[AUD]]></fee_type>
<local_fee_type><![CDATA[AUD]]></local_fee_type>
<local_total_fee><![CDATA[100]]></local_total_fee>
<mch_id><![CDATA[100550002127]]></mch_id>
<need_query><![CDATA[Y]]></need_query>
<nonce_str><![CDATA[1608383533]]></nonce_str>
<order_fee><![CDATA[100]]></order_fee>
<out_trade_no><![CDATA[1608383533]]></out_trade_no>
<out_transaction_id><![CDATA[902012192112152302758]]></out_transaction_id>
```

```

<pay_result><![CDATA[1]]></pay_result>
<result_code><![CDATA[1]]></result_code>
<sign><![CDATA[6A859C769308854FB895F441B620FE4C]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<status><![CDATA[0]]></status>
<time_end><![CDATA[20201219211215]]></time_end>
<total_fee><![CDATA[0]]></total_fee>
<trade_type><![CDATA[pay.expresspay.pay]]></trade_type>
<transaction_id><![CDATA[100550002127202012191076647141]]></transaction_id>
<uuid><![CDATA[0068d9b6b84808dece1f08800ba16c06a]]></uuid>
<version><![CDATA[2.0]]></version>
</xml>
    
```

5.2 Order query interface

Request type: POST

Request format: XML

Request parameters: as follows

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

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)	Value : UTF-8
sign_type	Yes	String(16)	SHA256 : SHA256; RSA : RSA_1_256
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
out_trade_no	No	String(32)	The unique trade reference of merchant system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. out_trade_no and transaction_id at least one required. 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 the section 4 'Digital Signature'.

Example

```
<xml><body><![CDATA[Test order query]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608383774]]></nonce_str>
<out_trade_no><![CDATA[1608383533]]></out_trade_no>
<service><![CDATA[unified.trade.query]]></service>
<sign><![CDATA[838BB3FABAC3B980E850A2E7C6EB06EB]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
</xml>
```

返回參數:

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	SHA256 : SHA256; RSA : RSA_1_256
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 returned when status is "0"

result_code	Yes	String(16)	"0" : success. Others value : fail.
mch_id	Yes	String(32)	Merchant ID, Specifies an official account id assigned by platform.
device_info	No	String(32)	Specifies a Terminal device id.
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
err_msg	No	String (128)	Error information description
sign	Yes	String(344)	Please refer to the section 4 'Digital Signature'.
reverse_id	No	String	Parameters returned when the merchant calls the cancel interface
reverse_status	No	String	Indicates whether the undo operation was successful
trade_state	Yes	String	The status of the order
The following fields are returned when the trade_state is SUCCESS			
trade_type	Yes	String (32)	pay.expresspay.pay
trade_state_desc	No	String(32)	Transaction status description
transaction_id	Yes	String(32)	platform order id
out_transaction_id	Yes	String(32)	order number provided by the third-party (WeChat or Alipay)
out_trade_no	Yes	String(32)	Specifies an order number created by a merchant's system, which is consistent with request.
total_fee	Yes	Int	Specifies the total amount. Integral number is allowed only. The unit of the fee is the minimal unit of the local currency.
fee_type	No	String(8)	Complies with ISO 4217 standards.
attach	No	String(127)	Specifies merchant's data package, which is returned as it is.
bank_type	No	String(16)	String states bank_type
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.
local_fee_type	No	String(8)	Merchant price currency, E.g AUD
local_total_fee	No	Int	Merchant price comparison amount
order_fee	Yes	Int	Order amount (price currency)

Example

```

<xml><bank_type><![CDATA[07]]></bank_type>
<charset><![CDATA[UTF-8]]></charset>
<fee_type><![CDATA[AUD]]></fee_type>
<local_fee_type><![CDATA[AUD]]></local_fee_type>
<local_total_fee><![CDATA[100]]></local_total_fee>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608383774]]></nonce_str>
<order_fee><![CDATA[100]]></order_fee>
<out_trade_no><![CDATA[1608383533]]></out_trade_no>
<out_transaction_id><![CDATA[902012192112152302758]]></out_transaction_id>

```

```

<pay_result><![CDATA[0]]></pay_result>
<result_code><![CDATA[0]]></result_code>
<sign><![CDATA[EE17A8318E6A4DBA353421E78AF64451]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<status><![CDATA[0]]></status>
<time_end><![CDATA[20201219211215]]></time_end>
<total_fee><![CDATA[100]]></total_fee>
<trade_state><![CDATA[SUCCESS]]></trade_state>
<trade_type><![CDATA[pay.expresspay.pay]]></trade_type>
<transaction_id><![CDATA[100550002127202012191076647141]]></transaction_id>
<version><![CDATA[2.0]]></version>
</xml>
    
```

5.3 Refund query interface

Request type: POST

Request format: XML

Request parameters: as follows

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

Field Name	Required	Type	Description
service	Yes	String(32)	Value :unified.trade.refundquery
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	SHA256 : SHA256; RSA : RSA_1_256
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
out_trade_no	No	String(32)	The unique trade reference of merchant system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
out_refund_no	No	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id	No	String(32)	Specifies the internal refund number, which is unique in the platform system. out_refund_no and refund_id at least one required. refund_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 the section 4 'Digital Signature'.

Example

```
<xml><body><![CDATA[Test order query]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608536843]]></nonce_str>
<out_refund_no><![CDATA[1608536636]]></out_refund_no>
<out_trade_no><![CDATA[1608536491]]></out_trade_no>
<service><![CDATA[unified.trade.refundquery]]></service>
<sign><![CDATA[92105838ADB1AE8F55057E090B814FCC]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
</xml>
```

Response:

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2. 0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	SHA256 : SHA256; RSA : RSA_1_256
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 returned when status is "0"

result_code	Yes	String(16)	"0" : success. Others value : fail.
mch_id	Yes	String(32)	Specifies an unique id assigned by platform.
device_info	No	String(32)	Specifies a Terminal device id.
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

err_msg	No	String (128)	Error information description
sign	Yes	String(344)	Please refer to the section 4 'Digital Signature'.

The following fields will returned when status and result_code both are "0"			
transaction_id	Yes	String (32)	The unique trade reference of platform system.
out_trade_no	Yes	String (32)	Order number in the merchant system
out_refund_no_0	Yes	String (32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id_0	Yes	String (32)	Platform refund order number
refund_channel_0	Yes	String (16)	Value: ORIGINAL
refund_fee_0	Yes	Int	The total amount of the refund, in cents, partial refunds can be made
total_fee	No	Int	Order channel amount
fee_type	No	String (8)	Order channel currency
order_fee	No	Int	order amount
local_total_fee	No	Int	Merchant bid amount
local_fee_type	No	String (8)	Merchant price currency, E.g AUD
out_transaction_id	No	String (32)	Third-party merchant number
trade_type	No	String (32)	pay.expresspay.pay
refund_time_0	Yes	String (32)	Refund time
refund_status_0	Yes	String	Refund status
trans_time	No	String	Refund time
refund_count	No	String	Expresspay Only one refund record can be checked

Example

```
<xml><charset><![CDATA[UTF-8]]></charset>
<fee_type><![CDATA[AUD]]></fee_type>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608536843]]></nonce_str>
<out_refund_no_0><![CDATA[1608536636]]></out_refund_no_0>
<out_trade_no><![CDATA[1608536491]]></out_trade_no>
<out_transaction_id><![CDATA[662012211541542749008]]></out_transaction_id>
<refund_channel_0><![CDATA[ORIGINAL]]></refund_channel_0>
<refund_count><![CDATA[1]]></refund_count>
<refund_fee_0><![CDATA[10]]></refund_fee_0>
<refund_id_0><![CDATA[100550002127202012211276647223]]></refund_id_0>
<refund_status_0><![CDATA[SUCCESS]]></refund_status_0>
<refund_time_0><![CDATA[20201221154711]]></refund_time_0>
<result_code><![CDATA[0]]></result_code>
<sign><![CDATA[9F571EF77919C0F7EA4FAFFB0B383953]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<status><![CDATA[0]]></status>
```

```
<total_fee><![CDATA[10]]></total_fee>
<trade_type><![CDATA[pay.expresspay.pay]]></trade_type>
<trans_time><![CDATA[20201221154711]]></trans_time>
<transaction_id><![CDATA[100550002127202012211276647221]]></transaction_id>
<version><![CDATA[2.0]]></version>
</xml>
```

5.4 Refund interface

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

1. Refund method

Currently only supports the original return refund

Note: The refund to the bank card, due to the different processing time of each bank, generally the refund request will be received within 7 working days after the initiation of the refund request.

For partial refunds of the same order, the same order number and different out_refund_no need to be set. If a refund is failed and resubmitted, the original out_refund_no shall be used. The total refund amount cannot exceed the actual amount paid by the user (the cash coupon amount cannot be refunded)

2. Refund restrictions

Merchants should pay attention to refund restrictions during refund operations, and avoid initiating unsuccessful refund requests. The following are the main refund restrictions:

In the platform, the refund application number (there is this parameter in the refund interface) uniquely determines a refund, rather than the transaction ticket number. The refund application form number is generated by the merchant, so the merchant must ensure the uniqueness of the refund application form.

Request type: POST

Request format: XML

Request parameters: as follows

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

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)	Value : UTF-8
sign_type	No	String(16)	SHA256 : SHA256; RSA : RSA_1_256
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
out_trade_no	No	String(32)	The unique trade reference of merchant system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. out_trade_no and transaction_id at least one required. 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 of the transaction. The unit of the fee is the minimal unit of the local 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	String(32)	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 the section 4 'Digital Signature'.

Example

```
<xml><mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608384472]]></nonce_str>
<op_user_id><![CDATA[11]]></op_user_id>
<out_refund_no><![CDATA[1608384472]]></out_refund_no>
<out_trade_no><![CDATA[1608384211]]></out_trade_no>
<refund_channel><![CDATA[ORIGINAL]]></refund_channel>
```

```

<refund_fee><![CDATA[2]]></refund_fee>
<service><![CDATA[unified.trade.refund]]></service>
<sign><![CDATA[721E2E68744E9D0966A36294B5CB583B]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<total_fee><![CDATA[10]]></total_fee>
</xml>

```

返回結果：

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	SHA256 : SHA256; RSA : RSA_1_256
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 are returned when status is 0

result_code	Yes	String(16)	0 means success, non-zero means failure
mch_id	Yes	String(32)	Merchant's unique identification code, assigned by the platform
device_info	No	String(32)	Terminal device number
nonce_str	Yes	String(32)	Random string, no longer than 32 bits
err_code	No	String(32)	error code
err_msg	No	String(128)	Error message
sign	Yes	String(344)	Signature result, see "Chapter 4 Signature Rules" for details

The following fields are returned when both status and result_code are 0

transaction_id	Yes	String(32)	Platform transaction number
out_trade_no	Yes	String(32)	Order number in the merchant system
out_refund_no	Yes	String(32)	Merchant refund order number
refund_id	Yes	String(32)	Platform refund order number
refund_channel	Yes	String(16)	Value: ORIGINAL
refund_fee	Yes	Int	The total amount of the refund, in cents, partial refunds can be

			made
total_fee	No	Int	Order channel amount
fee_type	No	String (8)	Order channel currency
order_fee	No	Int	order amount
local_total_fee	No	Int	Merchant bid amount
local_fee_type	No	String (8)	Merchant price currency, E.g AUD
out_transaction_id	No	String (32)	Third-party merchant account
trade_type	No	String (32)	pay.expresspay.pay

Example

```

<xml><charset><![CDATA[UTF-8]]></charset>
<fee_type><![CDATA[AUD]]></fee_type>
<local_fee_type><![CDATA[AUD]]></local_fee_type>
<local_total_fee><![CDATA[10]]></local_total_fee>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608384472]]></nonce_str>
<order_fee><![CDATA[2]]></order_fee>
<out_refund_no><![CDATA[1608384472]]></out_refund_no>
<out_trade_no><![CDATA[1608384211]]></out_trade_no>
<out_transaction_id><![CDATA[902012192124322311508]]></out_transaction_id>
<refund_channel><![CDATA[ORIGINAL]]></refund_channel>
<refund_fee><![CDATA[2]]></refund_fee>
<refund_id><![CDATA[100550002127202012191276647145]]></refund_id>
<result_code><![CDATA[0]]></result_code>
<sign><![CDATA[AA5A8F0787FDD6BB6F044CB8BBBDF5EC]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<status><![CDATA[0]]></status>
<total_fee><![CDATA[10]]></total_fee>
<trade_type><![CDATA[pay.expresspay.pay]]></trade_type>
<transaction_id><![CDATA[100550002127202012191076647144]]></transaction_id>
<version><![CDATA[2.0]]></version>
</xml>

```

5.5 Cancel interface

1. Points to note for the withdrawal of the interface

1. The cancellation interface can only be called within 5 minutes after the payment order is paid, and only full cancellation is allowed. Please call the refund interface for subsequent refunds.

Note: The cancellation can only be initiated if the payment is successful

Request type: POST

Request format: XML

Request parameters: as follows

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

Field Name	Required	Type	Description
service	Yes	String(32)	Value : unified.micropay.reverse
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	Signature type SHA256: 'SHA256'
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
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
transaction_id	No	String(32)	Platform transaction number, at least one of out_trade_no and transaction_id is required, and transaction_id takes precedence if it exists at the same time.
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'.

Example

```
<xml><body><![CDATA[Test cancellation query]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608385032]]></nonce_str>
<out_trade_no><![CDATA[1608385012]]></out_trade_no>
<service><![CDATA[unified.micropay.reverse]]></service>
<sign><![CDATA[18A42328AE9FF8B6B0586A65E77DC22F]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
</xml>
```

Return parameter:

Field Name	Required	Type	Description
------------	----------	------	-------------

ed			
version	Yes	String(8)	Version number. default value : 2. 0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	Signature type MD5: 'MD5' (default) SHA256: 'SHA256'
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 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. If the payment is completed, a refund is initiated. 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.
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
device_info	No	String(32)	Terminal device number
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.
err_msg	No	String(128)	Error information description
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
transaction_id	Yes	String(32)	Platform transaction number
out_transaction_id	Yes	String(32)	Third-party transaction number
out_trade_no	Yes	String(32)	The order number in the merchant system, 5 to 32 characters, can only contain alphanumeric or underscores, is case sensitive, to ensure that it is unique in the merchant system
accepted	Yes	String	Indicates whether the request is accepted successfully, 0: success 1: failure

The following fields are returned when both status and result_code are 0			
trade_state	Yes	String	Order Status

Example

```
<xml><accepted><![CDATA[0]]></accepted>
<charset><![CDATA[UTF-8]]></charset>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608563554]]></nonce_str>
<out_trade_no><![CDATA[1608563490]]></out_trade_no>
<out_transaction_id><![CDATA[162012212311560835268]]></out_transaction_id>
<result_code><![CDATA[0]]></result_code>
<sign><![CDATA[A1C7839C1D604C7EC94B6D7074014B8D]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<status><![CDATA[0]]></status>
<trade_state><![CDATA[REVERSE]]></trade_state>
<transaction_id><![CDATA[100550002127202012211276647265]]></transaction_id>
<version><![CDATA[2.0]]></version>
</xml>
```

5.6 Scan code notification interface

The notification URL is the parameter `notify_url` submitted in Section 5.1. After the payment is completed, the platform will send the relevant payment and user information to this URL, and the merchant needs to receive the processing information. When interacting with background notifications, if the platform receives a merchant's response that is not a pure string of success or returns after more than 5 seconds, the platform considers that the notification has failed, and the platform will pass a certain strategy (the notification frequency is 0/15/15/30/180 /1800/1800/1800/1800/3600, unit: second) Indirectly re-initiate the notification to increase the success rate of the notification as much as possible, but there is no guarantee that the notification will eventually succeed.

Due to the situation of resending background notifications, the same notification may be sent to the merchant system multiple times. The merchant system must be able to correctly handle duplicate notifications. The recommended approach is to first check the status of the corresponding business data when receiving a notification for processing, to determine whether the notification has been processed, and if it has not been processed, then process it again, and if it is processed, return the result directly as a success. Before the status check and processing of business data, data locks should be used for concurrency control to avoid data confusion caused by function reentry.

Special attention: After receiving the notification parameters, the merchant must verify the order number `out_trade_no` and `total_fee` in the received notification parameters with the order and amount of its own business system, and update the database order status after the verification is consistent. The background notification is carried out through the `notify_url` in the request, and the post method is sent to the merchant system (the content of the notification parameter is a string of xml)

Return format: XML

Swiftpass notification parameters: as follows

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	SHA256 : SHA256; RSA : RSA_1_256
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 <code>result_code</code> .
message	No	String(128)	Return message. Only return when the signature verification invalid.

The following fields will returned when status is “0”			
result_code	Yes	String(16)	“0” : success. Others value : fail.
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform.
device_info	No	String(32)	Specifies a Terminal device id.
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
err_msg	No	String (128)	Error information description
sign	Yes	String(344)	Please refer to the section 4 ‘Digital Signature’.

The following fields will returned when status and result_code both are “0”			
trade_type	Yes	String(32)	pay.expresspay.pay
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 number
out_transaction_id	Yes	String(32)	order number provided by the third-party
out_trade_no	Yes	String(32)	Specifies an order number created by a merchant's system, which is consistent with request.
total_fee	Yes	Int	Specifies the total amount. Integral number is allowed only. The unit of the fee is the minimal unit of the local currency.
fee_type	Yes	String(8)	Complies with ISO 4217 standards.
attach	No	String(127)	Specifies merchant's data package, which is returned as it is.
bank_type	Yes	String(16)	String states bank_type
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.
local_total_fee	No	Int	Merchant bid amount
local_fee_type	No	String (8)	Merchant price currency, E.g AUD

Example

```
<xml>
  <bank_type><![CDATA[CFT]]></bank_type>
  <charset><![CDATA[UTF-8]]></charset>
  <device_info><![CDATA[SPAY_AND]]></device_info>
  <discount_detail><![CDATA[{"discountAmt":"30.00","discountNote":"Uplan discount"}, {"discountAmt":"10.00","discountNote":"Instant Discount"}]]>
  </discount_detail>

  <fee_type><![CDATA[AUD]]></fee_type>
  <local_fee_type><![CDATA[AUD]]></local_fee_type>
  <local_total_fee><![CDATA[21850]]></local_total_fee>
  <mch_id><![CDATA[102520000212]]></mch_id>
  <nonce_str><![CDATA[1573186381348]]></nonce_str>
  <order_fee><![CDATA[21850]]></order_fee>
  <out_trade_no><![CDATA[10252000021270741546158683729]]></out_trade_no>
  <out_transaction_id><![CDATA[4200000437201911083557508581]]></out_transaction_id>
  <pay_result><![CDATA[0]]></pay_result>
  <platform_rate><![CDATA[2040399.9184]]></platform_rate>
  <rate><![CDATA[699040000]]></rate>
  <result_code><![CDATA[0]]></result_code>
  <sign><![CDATA[0FEC037A2D48654C0FC8AEA96F1C4B55]]></sign>
  <sign_type><![CDATA[MD5]]></sign_type>
  <status><![CDATA[0]]></status>
```

```

<time_end><![CDATA[20191108121300]]></time_end>
<total_fee><![CDATA[446]]></total_fee>
<trade_type><![CDATA[pay.expresspay.pay]]></trade_type>
<transaction_id><![CDATA[102520000212201911081086849655]]></transaction_id>
<unsettled_discount_fee><![CDATA[3000]]></unsettled_discount_fee>
<version><![CDATA[2.0]]></version>
</xml>
    
```

Background notification result feedback

The platform server sends notifications, the post sends XML data streams, the merchant notify_Url address receives the notification results, and the receiving method demo is written (such as the callback method in php, the notify.aspx file in c#, the TestPayResultServlet method in java), and the merchant does business After processing, the processing result needs to be returned in the form of a pure string, as follows:

Return result description	Return result description
success	The processing is successful, and the platform will not provide follow-up notifications after receiving this result
fail or other characters	If the processing is unsuccessful, the platform receives this result or does not receive any result, the system will notify again through the replenishment mechanism (see section 5 for details)

5.7 Binding card interface

Request type: POST

Request format: XML

Request parameters: as follows

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

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type: pay.card.bindcard
version	No	String(8)	Version number. The default value of version is 2.0.
charset	No	String(8)	Value: UTF-8.
sign_type	No	String(12)	SHA256 signature: SHA256 RSA signature: RSA_1_256
mch_id	Yes	String(32)	Merchant' s unique identification code, assigned by the platform
nonce_str	Yes	String(32)	Random string, no longer than 32 bits
sign	Yes	String(344)	Signature result, see "Chapter 4 Signature Rules" for details

pan	Yes	String(19)	card number
expired	Yes	String	Expiration time of the card MM/yy
request_time	Yes	String	Request time yyyyMMdd
token_expired	No	String	Token expiration time yyyyMMdd

Example
<pre><xml> <expired><![CDATA[12/33]]></expired> <mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip> <mch_id><![CDATA[100550002127]]></mch_id> <nonce_str><![CDATA[1608537335]]></nonce_str> <pan><![CDATA[625094*****0014]]></pan> <request_time><![CDATA[20201221]]></request_time> <service><![CDATA[pay.card.bindcard]]></service> <sign><![CDATA[5E0D9024DDE72F8B29B2A479E85DBF72]]></sign> <sign_type><![CDATA[MD5]]></sign_type> </xml></pre>

Return result:

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. The default value of version is 2.0.
charset	Yes	String(8)	Value: UTF-8
sign_type	Yes	String(16)	SHA256 signature: SHA256 RSA signature: RSA_1_256
status	Yes	String(16)	0 means success, non-zero means failure. This field is a communication identifier, not a transaction identifier, you need to check result_code to judge whether the transaction is successful
message	No	String(128)	Return information. If it is not empty, it is the reason for the error. Signature failure. Parameter format verification error

The following fields are returned when status is 0			
result_code	Yes	String(16)	0 means success, non-zero means failure
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by 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
err_msg	No	String(128)	Error information description

sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
------	-----	-------------	--

The following fields are returned when both status and result_code are 0			
pan	Yes	String	card number
expired	Yes	String	Card expiration time
request_time	Yes	String	Request time yyyyMMdd
token_expired	Yes	String	Token expiration time yyyyMMdd
token_id	Yes	String	tokenId

Example
<pre> <xml><charset><![CDATA[UTF-8]]></charset> <expired><![CDATA[12/33]]></expired> <mch_id><![CDATA[100550002127]]></mch_id> <msg><![CDATA[成功]]></msg> <nonce_str><![CDATA[1608537335]]></nonce_str> <pan><![CDATA[625094*****0014]]></pan> <request_time><![CDATA[20201221]]></request_time> <sign><![CDATA[9FC1390DB98C52778FA1A272DBF5B946]]></sign> <sign_type><![CDATA[MD5]]></sign_type> <status><![CDATA[0]]></status> <status_code><![CDATA[000006]]></status_code> <token_expired><![CDATA[20210621]]></token_expired> <token_id><![CDATA[100550002127GCca9vO]]></token_id> <version><![CDATA[2.0]]></version> </xml> </pre>

5.8 Unbind interface

Request type: POST

Request format: XML

Request parameters: as follows

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

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type: pay.card.unbindcard
version	No	String(8)	Version number. The default value of version is 2.0.
charset	No	String(8)	Value: UTF-8.

sign_type	No	String(12)	MD5 signature: MD5 (default);
mch_id	Yes	String(32)	SHA256 signature: SHA256
nonce_str	Yes	String(32)	RSA signature: RSA_1_256
sign	Yes	String(344)	Merchant' s unique identification code, assigned by the platform
pan	Yes	String	Random string, no longer than 32 bits
expired	Yes	String	Signature result, see "Chapter 4 Signature Rules" for details
token_id	Yes	String	Token_id

Example

```
<xml>
<expired><![CDATA[12/33]]></expired>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100550002127]]></mch_id>
<nonce_str><![CDATA[1608537247]]></nonce_str>
<pan><![CDATA[625*****00014]]></pan>
<service><![CDATA[pay.card.unbindcard]]></service>
<sign><![CDATA[3F7F4F96CCA6BE0BCD95827FF719D770]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<token_id><![CDATA[100550002127Cz95tz1]]></token_id>
</xml>
```

返回結果:

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. The default value of version is 2.0.
charset	Yes	String(8)	Value: UTF-8
sign_type	Yes	String(16)	SHA256 signature: SHA256
status	Yes	String(16)	RSA signature: RSA_1_256
message	No	String(128)	0 means success, non-zero means failure. This field is a communication, non-transaction mark, you need to check result_code to judge whether the transaction is successful

The following fields are returned when status is 0

result_code	Yes	String(16)	0 means success, non-zero means failure
mch_id	Yes	String(32)	Merchant' s unique identification code, assigned by the platform
nonce_str	Yes	String(32)	Random string, no longer than 32 bits
err_code	No	String(32)	error code
err_msg	No	String(128)	Error message

sign	Yes	String (344)	Signature result, please refer to Chapter 4 Signature Rules for details
------	-----	--------------	---

The following fields are returned when both status and result_code are 0			
pan	Yes	String	card number
expired	Yes	String	Expiration time of the card MM/yy
token_status	Yes	String	Token status
status_code	Yes	String	Unbundled state
msg	Yes	String	description of status_code

Example
<pre> <xml><charset><![CDATA[UTF-8]]></charset> <expired><![CDATA[12/33]]></expired> <mch_id><![CDATA[100550002127]]></mch_id> <msg><![CDATA[成功]]></msg> <nonce_str><![CDATA[1608537247]]></nonce_str> <pan><![CDATA[625094*****0014]]></pan> <sign><![CDATA[984BFAD030DF6CEF779950549ED055FD]]></sign> <sign_type><![CDATA[MD5]]></sign_type> <status><![CDATA[0]]></status> <status_code><![CDATA[000006]]></status_code> <token_status><![CDATA[UNBINDING]]></token_status> <version><![CDATA[2.0]]></version> </xml> </pre>

6 Precautions

1. All the units involved in the amount of money are points, the smallest unit is 1 point, and there must be no decimals.
2. The notify_url is that the platform server directly initiates a request from the background to the merchant server. The merchant cannot check the user's cookie or session during processing; the merchant must update the DB and other shipping processes after the notify_url is completed to ensure that the platform replenishes the order when the order is dropped. Can successfully make up
3. The notify_url may be notified repeatedly, and merchants need to do de-duplication to avoid multiple shipments
4. For the notification received by notify_url, if the merchant has successfully processed or checked that the order has been processed, it needs to return the pure string success indicating the successful processing, and the string success is not case-sensitive; if we do not receive the returned success,

our server will continue Send a notification to you, and there will be no more notification in three hours; assuming that all orders do not return success, it will increase the notification load of our server. In the worst case, it may cause the normal notification of the merchant to be delayed; in addition, we will Urge you to improve. If you do not improve for a long time, R&D or operation and maintenance technology will take control measures on the payment interface opened by your company.

5. The parameters that must be filled in as yes in the return parameters will definitely be returned, and the parameters that must be filled in as no will not necessarily be returned. The actual return parameters may not be exactly the same as the document due to upgrade or configuration, and must be actually received The parameters shall prevail

6. Other matters needing attention

(1) Parameter case

Please pay attention to the capitalization of the characters required in the document, such as "after the signature operation, the characters of the string should be converted to uppercase".

(2) Parameter format problem

For all incoming parameters, please pay attention to the specific requirements everywhere in the document.

(3) Time stamp issue

Please use Linux timestamp, note that it is a string format.

(4) The payment problem of the order number of the same merchant

The merchant' s out_trade_no must be globally unique, and a unique order number is required for both debugging and production environments