



Merchant API Documentation

Official Account & Mini Program
Interface Specification

Document version: 1.5.3

Interface version: 2.0

TABLE OF CONTENTS

1	Introduction.....	4
1.1	Abstract.....	4
1.2	Audience.....	4
2	Program Overview.....	4
2.1	Industry Background.....	4
2.2	Business realization process.....	5
2.2.1	Quick Pay business.....	5
3	Data Format.....	7
3.1	Request data.....	7
3.2	XML data format.....	7
4	Digital Signature.....	9
4.1	Original string of signature.....	9
4.2	Method of signature.....	11
4.2.1	MD5 signature.....	11
4.2.2	SHA256 signature.....	12
4.2.3	RSA signature.....	14
5	Mechanism To Supplement Order.....	17
6	Payment Interface.....	18
6.1	Initial Request API.....	18
6.1.1	Business function.....	18
6.1.2	Interactive mode.....	18
6.1.3	Request Parameters.....	18
6.1.4	Response result.....	20
6.2	Primitive js payment API.....	21
6.2.1	Usage example.....	21
6.2.2	Interactive mode.....	22
6.2.3	Request Parameters.....	22
6.3	Official accounts JS payment API.....	23
6.3.1	Business function.....	23
6.3.2	Interactive mode.....	23
6.3.3	Request Parameters.....	23
6.4	Notification of JS pay.....	24
6.4.1	Notification request parameters.....	24
6.4.2	Response of notification.....	27
6.5	Retrieve transaction result interface.....	27
6.5.1	Business function.....	27
6.5.2	Interactive mode.....	28
6.5.3	Request Parameters.....	28
6.5.4	Response result.....	28

6.6 Refund interface.....	30
6.6.1 Business function	30
6.6.2 Interactive mode.....	32
6.6.3 Request Parameters	32
6.6.4 Response result.....	33
6.7 Retrieve refund result interface.....	34
6.7.1 Request Parameters	34
6.7.2 Response result.....	35
6.8 Close Order Interface	37
6.8.1 Business function	37
6.8.2 Interactive mode.....	38
6.8.3 Request Parameters	38
6.8.4 Response result.....	38
7 Notes	39

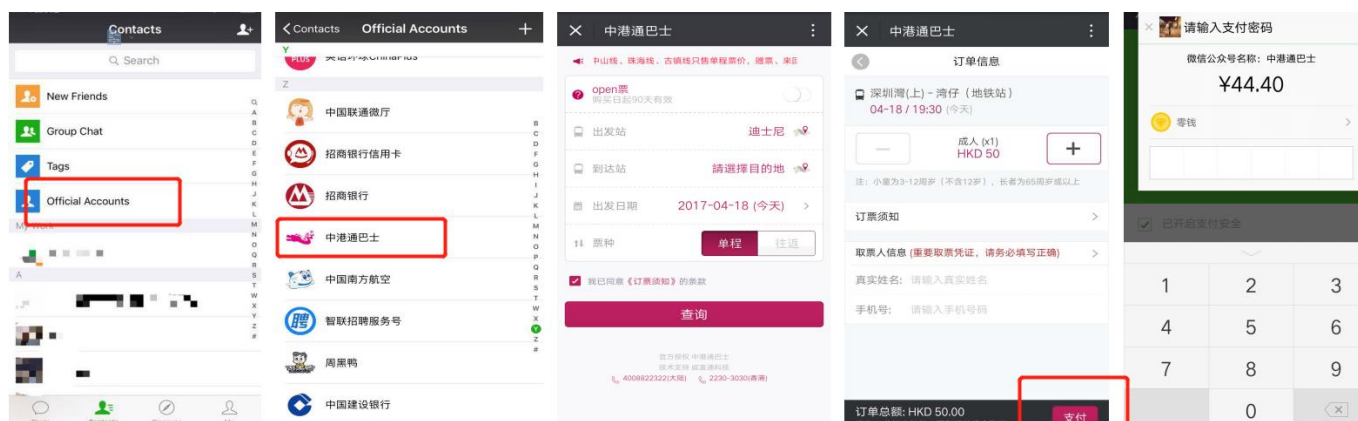
Document Changes		
Date	Version	Description
2018-02-24	1.0	First draft
2018-06-01	1.1	Request URL is changed to https://gateway.wepayez.com/pay/gateway
2019-06-13	1.2	Update SHA256 & RSA signature
2023-02-24	1.5.3	Updated the diagram, signature sample message of SHA256. Added MD5 signature method; Updated the following fields: fields charset, sign_type, out_trade_no, device_info, attach, notify_url, total_fee, err_code, err_msg, out_trade_no, transaction_id, refund_id, out_trade_refund;

1 Introduction

1.1 Abstract

JS Pay is a payment method which requires merchants to have official account. WeChat users open the H5 page link in WeChatApp and complete the payment.

- WeChat users go in a merchants' official account and open one of the home page then complete the payment.
- WeChat users open a link of merchants official account and complete the payment.



JS Pay can be applied in On-line payment scene. E.g on-line tickets booking, on-line shopping mall and etc.

1.2 Audience

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

2 Program Overview

2.1 Industry Background

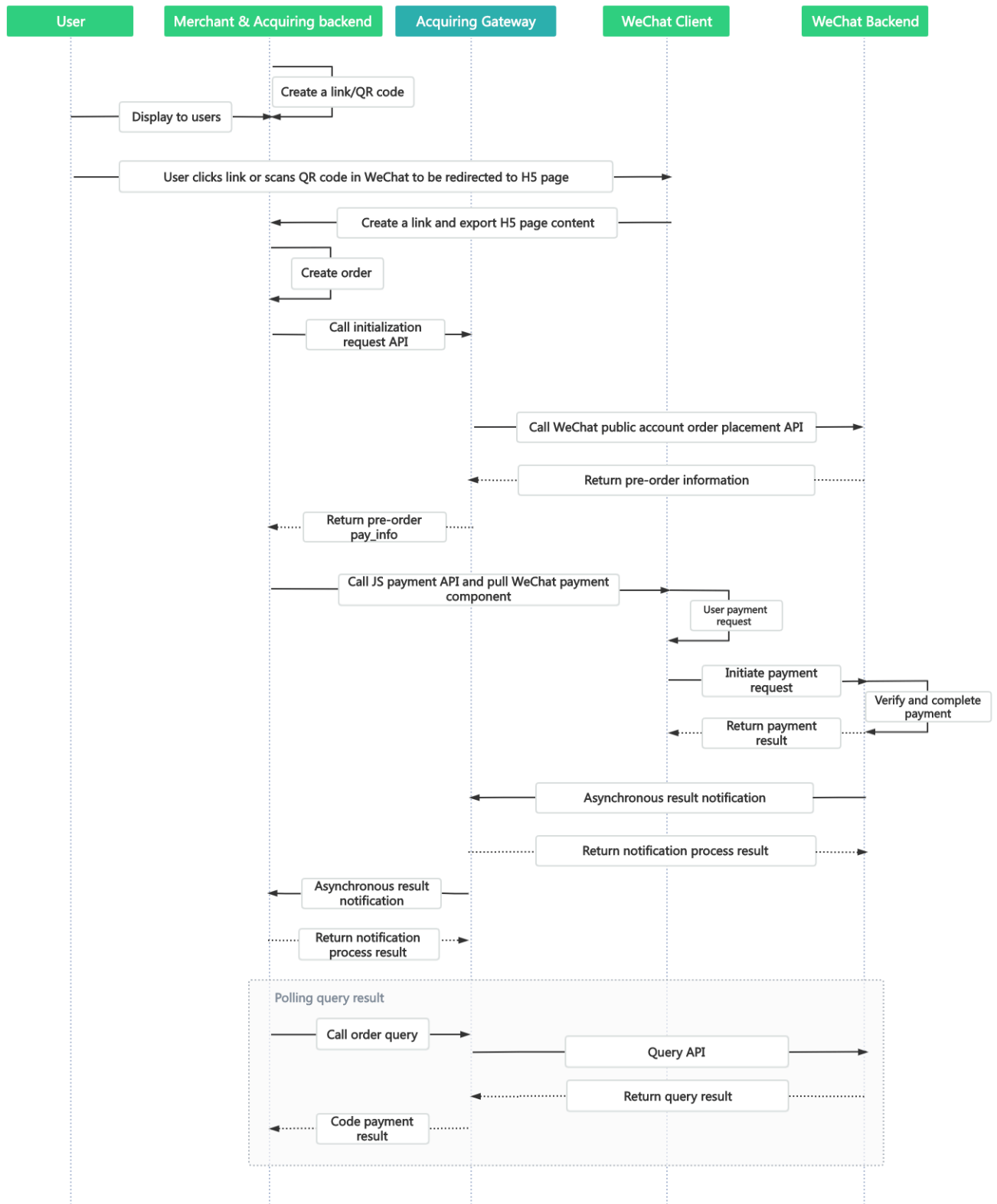
WeChat payment is a payment service function which is provided based on WeChat application, and meanwhile it provides the commercial tenants with such support functions as sales,

marketing analysis and management of account. Customers could complete a payment by scanning QR code, being scanned payment QR code or other payment methods.

2.2 Business realization process

2.2.1 Quick Pay business

The payer scans a QR



3 Data Format

3.1 Request data

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

```
<xml>

  <appid><![CDATA[wx9011114452]]></appid>

  <body><![CDATA[Paymen]]></body>

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

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

  <mch_create_ip><![CDATA[209.52.88.23]]></mch_create_ip>

  <mch_id><![CDATA[755100001]]></mch_id>

  <nonce_str><![CDATA[2309463563]]></nonce_str>

  <notify_url><![CDATA[https://xxx/payment/]]></notify_url>

  <openid><![CDATA[orWS61bl81lh7u-MfBpHv-uN9qxQ]]></openid>

  <out_trade_no><![CDATA[202302137551000001Kcawr]]></out_trade_no>

  <service><![CDATA[pay.weixin.jsipay]]></service>

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

  <time_expire><![CDATA[20230213140704]]></time_expire>

  <time_start><![CDATA[20230213140604]]></time_start>

  <total_fee><![CDATA[3362]]></total_fee>

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

  <sign><![CDATA[50B32EF88760EE8E3EEB8F6A3FB5371B]]></sign>

</xml>
```

3.2 XML data format

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

Protocol error return:

```
<xml>
```



```
<status>500</status>

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

</xml>
```

Successful response:

```
<xml>

<status>0</status>

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

<appid><![CDATA[wx2421b1c4370ec43b]]></appid>

<mch_id><![CDATA[10000100]]></mch_id>

<device_info><![CDATA[1000]]></device_info>

<nonce_str><![CDATA[FvYSnPuFFPkAr77M]]></nonce_str>

<sign><![CDATA[63238039D6E43634297CF2A6EB5F3B72]]></sign>

<result_code>0</result_code>

<openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>

<is_subscribe><![CDATA[Y]]></is_subscribe>

<trade_type><![CDATA[pay.weixin.jspay]]></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>
```

Normal 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[二维码已过期，请刷新再试]]></err_code_des>

</xml>
```

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

4 Digital Signature

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

There are two steps in digital signature.

1. Follow the rules to connect the original string that needs to be signed;
2. Calculate the result of signature according to specific algorithm and key.

Generally, the failed result will not be signed.

4.1 Original string of signature

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

1. Besides the sign field, all parameter fields will be ranked in ascending order according to the ASCII of the field name and then connected in the format of QueryString (i.e.

key1=value1&key2=value2...), and the null value will not transfer and will not participate in formation of string of signature.

2. In the original string of signature, both the field name and field value will adopt original values and will not conduct URL Encode.
3. The response or notification information returned by platform might increase parameters due to upgrading, and this case should be allowed when the response signature is verified.

Example:

Calling an interface with following fields:

```
<xml>

  <appid><![CDATA[wx9011114452]]></appid>

  <body><![CDATA[Paymen]]></body>

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

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

  <mch_create_ip><![CDATA[209.52.88.23]]></mch_create_ip>

  <mch_id><![CDATA[755100001]]></mch_id>

  <nonce_str><![CDATA[2309463563]]></nonce_str>

  <notify_url><![CDATA[https://xxx/payment/]]></notify_url>

  <openid><![CDATA[orWS61bl81lh7u-MfBpHv-uN9qxQ]]></openid>

  <out_trade_no><![CDATA[202302137551000001Kcawr]]></out_trade_no>

  <service><![CDATA[pay.weixin.jspay]]></service>

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

  <time_expire><![CDATA[20230213140704]]></time_expire>

  <time_start><![CDATA[20230213140604]]></time_start>

  <total_fee><![CDATA[3362]]></total_fee>

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

  <sign><![CDATA[28CF57F2C1AB21CB49758B1301228EB3]]></sign>

</xml>
```

The signature field sequence:

```
body=TestPay&mch_create_ip=127.0.0.1&mch_id=7551000001&nonce_str=1409196838&notify_url=http://227.0.0.1:9001/javak/&out_trade_no=141903606228&service=pay.weixin.jspay.intl&total_fee=1
```

4.2 Method of signature

MD5 & SHA256 & RSA signature are supported

Signature is a kind of abstract generation algorithm, and if the content of communication keys of the merchant is added to the back of the original string of signature and then signature operation is made, then the abstract character string formed is the signature result. In order to facilitate the comparison, the signature result is uniformly converted to the uppercase letter.

Notes: the coded character set appointed when converting the character string into the byte stream at signing should be in accordance with parameter charset.

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>
  <appid><![CDATA[wx9011114452]]></appid>
  <body><![CDATA[Paymen]]></body>
  <charset><![CDATA[UTF-8]]></charset>
  <is_raw><![CDATA[1]]></is_raw>
  <mch_create_ip><![CDATA[209.52.88.23]]></mch_create_ip>
  <mch_id><![CDATA[7551000001]]></mch_id>
  <nonce_str><![CDATA[2309463563]]></nonce_str>
  <notify_url><![CDATA[https://xxx/payment/]]></notify_url>
  <openid><![CDATA[orWS61bl81lh7u-MfBpHv-uN9qxQ]]></openid>
  <out_trade_no><![CDATA[202302137551000001Kcawr]]></out_trade_no>
  <service><![CDATA[pay.weixin.jspay]]></service>
```

```

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

<time_expire><![CDATA[20230213140704]]></time_expire>

<time_start><![CDATA[20230213140604]]></time_start>

<total_fee><![CDATA[3362]]></total_fee>

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

<sign><![CDATA[28CF57F2C1AB21CB49758B1301228EB3]]></sign>

</xml>

```

Suppose merchant key is: 9f72151b6592fab3e0c63a1ab3c0877b

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

```

body= Paymen&appid= wx9011114452&mch_create_ip=209.52.88.23&mch_id=7551000001&nonce_str=2309463563&notify_url=
https://xxxx/payment/&out_trade_no=202302137551000001Kcawr &service=pay.weixin.raw.app&total_fee=3362&charset=UTF-8&is_raw=1&openid=
orWS61b181lh7u-MfBpHv-uN9qxQ&sign_type=MD5&time_expire=20230213140704&time_start=20230213140604& version=1.0

```

ii: sign after process b:

```

sign

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

=md5(body= Paymen&appid= wx9011114452&mch_create_ip=209.52.88.23&mch_id=7551000001&nonce_str=2309463563&notify_url=
https://xxxx/payment/&out_trade_no=202302137551000001Kcawr &service=pay.weixin.raw.app&total_fee=3362&charset=UTF-8&is_raw=1&openid=
orWS61b181lh7u-MfBpHv-uN9qxQ&sign_type=MD5&time_expire=20230213140704&time_start=20230213140604& version=1.0
&key=9d101c97133837e13dde2d32a5054abb).toUpperCase()="28CF57F2C1AB21CB49758B1301228EB3 "

```

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>

```

```
<appid><![CDATA[wx9011114452]]></appid>

<body><![CDATA[Paymen]]></body>

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

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

<mch_create_ip><![CDATA[209.52.88.23]]></mch_create_ip>

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

<nonce_str><![CDATA[2309463563]]></nonce_str>

<notify_url><![CDATA[https://xxx/payment/]]></notify_url>

<openid><![CDATA[orWS61b181h7u-MfBpHv-uN9qxQ]]></openid>

<out_trade_no><![CDATA[202302137551000001Kcawr]]></out_trade_no>

<service><![CDATA[pay.weixin.jsPAY]]></service>

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

<time_expire><![CDATA[20230213140704]]></time_expire>

<time_start><![CDATA[20230213140604]]></time_start>

<total_fee><![CDATA[3362]]></total_fee>

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

<sign><![CDATA[90409EC50C3B9380112EF7A8DBA7BC983EDEB7E01EDB7554EAA10BB07056023F]]></sign>

</xml>
```

Merchant signature key: 9f72151b6592fab3e0c63a1ab3c0877b

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

```
body= Paymen&appid= wx9011114452&mch_create_ip=209.52.88.23&mch_id=7551000001&nonce_str=2309463563&notify_url=
https://xxx/payment/&out_trade_no=202302137551000001Kcawr &service=pay.weixin.raw.app&total_fee=3362&charset=UTF-8&is_raw=1&openid=
orWS61b181h7u-MfBpHv-uN9qxQ&sign_type=SHA256&time_expire=20230213140704&time_start=20230213140604& version=1.0
```

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

```
sign
```

```
=SHA256(string1&key=18e0a2ad5d5571af14b855fcf33091f4).toUpperCase  
  
=SHA256(body= Paymen&appid= wx9011114452&mch_create_ip=209.52.88.23&mch_id=7551000001&nonce_str=2309463563&notify_url=  
https://xxxx/payment/&out_trade_no=202302137551000001Kcawr &service=pay.weixin.raw.app&total_fee=3362&charset=UTF-8&is_raw=1&openid=  
orWS61bl81lh7u-MfBpHv-uN9qxQ&sign_type=SHA256&time_expire=20230213140704&time_start=20230213140604& version=1.0&key=  
18e0a2ad5d5571af14b855fcf33091f4).toUpperCase()  
  
="90409EC50C3B9380112EF7A8DBA7BC983EDEB7E01EDB7554EAA10BB07056023F"
```

4.2.3 RSA signature

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>  
  
<appid><![CDATA[wx9011114452]]></appid>  
  
<body><![CDATA[Paymen]]></body>  
  
<charset><![CDATA[UTF-8]]></charset>
```

```
<is_raw><![CDATA[1]]></is_raw>

<mch_create_ip><![CDATA[209.52.88.23]]></mch_create_ip>

<mch_id><![CDATA[755100001]]></mch_id>

<nonce_str><![CDATA[2309463563]]></nonce_str>

<notify_url><![CDATA[https://xxxx/payment/]]></notify_url>

<openid><![CDATA[orWS61bl81h7u-MfBpHv-uN9qxQ]]></openid>

<out_trade_no><![CDATA[20230213755100001Kcawr]]></out_trade_no>

<service><![CDATA[pay.weixin.jsipay]]></service>

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

<time_expire><![CDATA[20230213140704]]></time_expire>

<time_start><![CDATA[20230213140604]]></time_start>

<total_fee><![CDATA[3362]]></total_fee>

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

<sign><![CDATA[Ym4W6KkaRps6ODFQHonpZbFZg/HXo7fAZhPE5g/SDv53hl0m5VtyDzEGnjVGWzdH/Ry+Ijni9YsWoCkmj9JMAPN+fTcSZbhvYiu17u
PE7ijvgNFfUMGEkrRpwXKzc3pDpJWRJG+CTySgr3FZQABXwnOoPPdZ08fgWWhHu6wOpBydp74Co46HVr2l/tAKUDjIWJixtYmrfv1PdtWHgmP1gA
Pja4C/Abp0zaBx+ebbwPZn/it7tt8+I/BOszrYNidzi0iKxUNaM2miXEhL9urd/pe9wbk+C9E3ThKak+VGDvMoEGGborRylb43MkOYyBAI5lr0BjEnyH+HRcW
ZylA==]]></sign>

</xml>
```

Merchant RSA Private key:

MIIeVqIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEpO8nMyk+XuYsatHwnU4//m47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64Rjra0rZql23dvPW4vVomMRgRcoPon0YwVp+M6T5PaFgE4M8dh4IMZz57gVwOdd08F99Z92f3QgZtEj+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9Uu5v3WgLuuv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5YIRy2EIN504cujYVKU1wOSZgq9oJCynGR0aPuQWx58IHxEtAgMBAECCggEAHfEfD8qm2PTE2ITAVEC7F+TcgD84IUaz0dZnURtx6YIOoZ5+LH/zVG6juYLJU/Oo5RPAC+iMVS68u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQuXN8PkDcOxqDmZ2kPjzI4PZvBZRE0abduq+MatGzPgaUJzrWcB/N0oVlvrXp9PnOqfo/Y5nxmpOFClmJppIS3AL1pftNtQz09G15CPHDYtpUbXpT2MjjW4OLxKuPRoHSwUgo6LW9XSwNXfCuK+IbzLL0BhIWD9IV/+yCEUEbIN87yxxfhpQFaAhXj5W+B3YsMOZuK93+XMOpYmw8EpUDMOBOnvwb0NSHUrv2RUAKBqQDTojlnNS1e7+tjPzFtOhGPj1uCBPAEIEHAcnPgd80bEiujxMLCnGaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsr3CAAu7CadQ49TW+SovAvciy9AZuSVVlWyny6QdYfYpKe1LZYAEq5k+mB1Vh5q0RoxMNA5pGYKg8+4MmmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSHnO/ngT2OuxSTTkc8yrx7qd9SmAxXl5kR238DhMOQOnRBomldmVtAJuJg rdQyt0wXfeVQVqshqCUae/xhEbpSCdbPSZbKZZdplV0y6O5vXlhxw+1qAvXLcxw46s3R92QQKBgQCIQ+ejykwVPDILHMwSSehwvThufkCYWYUbbcvDowpOe5AMoZidTnju7MNjg2rLHTsCx/kBzOr+7THNw4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvgVtIGBCh6BW5JZcSlmT+ykIoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbuelBVuZW6UC3+8PeK82AjlLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBzLBD7Gm4vJtQW0DknYo5sMXS1JmofcKjBv7nbHXZTx3EtJSxpVaodpcA/HpsCuCP3AH2e1yk9sZ3wu6lBYsBAoGACYM60j1CVRNSZxUNRgiwVzS69q1eezPc7xQEganpVBI9SZcTnp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqxvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbfFO0mdP4CeyuG

PdvDIMXP4dJdLhMUL4pcJLI0B7gBE=

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

body= Paymen&appid= wx9011114452&mch_create_ip=209.52.88.23&mch_id=7551000001&nonce_str=2309463563¬ify_url=
https://xxxx/payment/&out_trade_no=202302137551000001Kcawr &service=pay.weixin.raw.app&total_fee=3362&charset=UTF-8&is_raw=1&openid=
orWS61bl81lh7u-MfBpHv-uN9qxQ&sign_type=RSA_1_256&time_expire=20230213140704&time_start=20230213140604& version=1.0

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

```
sign=RSA(string1 , Merchant RSA Private key)==RSA(body= Paymen&appid=  
wx9011114452&mch_create_ip=209.52.88.23&mch_id=7551000001&nonce_str=2309463563&notify_url=  
https://xxxx/payment/&out_trade_no=202302137551000001Kcawr &service=pay.weixin.raw.app&total_fee=3362&charset=UTF-8&is_raw=1&openid=  
orWS61bl81lh7u-MfBpHv-uN9qxQ&sign_type=RSA_1_256&time_expire=20230213140704&time_start=20230213140604&  
version=1.0,MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBAcwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEpO8nMyk+XuYSatHwnU4//m47R+4G2  
YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkq5G271g+PmrzFp7f40/E9m0qjeL64Rjra0rZql23dvPW4vVomMRgRcoPOn0YwVp+M6T5PaFgE  
4M8dh4IMZz57gVwOdd08F99Z92f3QgZtEjI+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9UU5v3WgLuuv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggs  
umhd4eYY5YIRy2EIN504cuYVKU1wOSZgq9oJCynGR0aPuQWx58IHxEtAgMBAAEcggEAHfEFd8qm2PTE2ITAVEC7F+TcgD84IUaZ0dZnURtx6YIOoZ  
5+LH/zV6GjuYLUJ/Oo5RPAC+iMVS68u2JMCp7zm8Ft7B3JkrbuHLNHGU6R7PQuXN8PkDcOxqDmZ2kPjzI4PZvBZRE0abdug+tMatGzpGAUjzrWcB/  
N0oVlvrXp9PnOqfo/Y5nxmpOFCImJpplS3AL1pftNtQZo9G15CPHDYtpUbXpTD2MjjW4OLxKuPRoHSwUgo6LW9XSwNXfcuK+lzbLL0BhIWD9IV/+yCEU  
EblN87yxxfhpQFaAhXj5W+B3YsMOZuK93+XMOpYmw8EpUDMOBOnvwb0NSHUrV2RUAQKBgQDTojlnNS1e7+tjPzFtOhGPJ1uCBPAEIEHAcnPgD80b  
EiujxMLCnGaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsfr3CAAu7CadQ49TW+SovAvciy9AZuSVVlwyu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNA5p  
GYKg8+4MmmsJi7X7QKBgQDAunCOqliH128bs/1VRihDpzuRW5QR/SRbO2saVg5RSHnO/nGT2OuxSTTkc8yrx7qd9SmAxXI5kR238DhMOQOnRBoml  
dmVtAJuJgrdQyt0wXfeVQqshqCUaE/xhEbpSCdbPSZbKZZdpIV0y6O5vXlhxw+1qAvXLcxw46s3R92QQKBgQCICQ+ejywkVPDILHMwSSehwThufkCY  
WYUbbcvDowpOe5AMoZidTnju7MNjgr2rLHTsCx/kBzOr+7THNwL4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpgVtlGB  
Ch6BW5JZcSlmT+ykIOoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbueIBVuzW6UC3+8PeK82AjlBLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBz  
LBDL7Gm4vJtqW0DknYo5sMXS1JmofckjBv7nbHXZT3EtJSxpVaOdpca/HpsCuCP3AH2e1yk9sZ3wu6IBYSBAoGACYM60j1CVRNSZxUNRgiwWzS  
69q1eezPc7xQEganpVBI9SZcTnp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqxvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbFfO  
OmdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLI0B7gBE=")=  
Ym4W6KkaRps6ODFQHonpZbFZg/HXo7fAZhPE5g/SDv53hI0m5VtyDzEGnjVGWzdH/Ry+Ijni9sWoCkmj9JMAPN+ftcSZbhvYiu17uPE7ijvgNfUMG
```

EkrRpwXKzcz3pDpJWRJG+CTySgr3FZQABXwnOoPPdZ08fgWhhHu6wOpBydp74Co46HvR2I/tAKUDjIWJixtwYmrfV1PdtWHgmP1gAPja4C/Abp0zaBx

+ebbwPZn/it7tt8+I/BOszrYNidzi0iKxUNaM2miXEhL9urd/pe9wbk+C9E3ThKak+VGDvMoEGGborRylb43MkOYyBAI5lr0BjEnyH+HRcWZylA==

5 Mechanism To Supplement Order

Notes: in respect of the backstage notification interactive mode, if the response of the commercial tenant received by platform is not **pure character string “success”** or if the response is given **after 5 seconds**, then the notification will be deemed as unsuccessful, and platform will adopt certain strategies (**the interval of notification: 0/15/15/30/180/1800/1800/1800/1800/3600 Unit: seconds**) to re-initiate notification **intermittently**, to improve the success rate of notification, but platform will not guarantee the final success of notification. Because there is the case where the backstage notification will be resent, the same notification might be sent to the commercial tenant system many times. The commercial tenant system must be able to process the repeated notification in a right way. The method recommended by platform is to first examine the status of the corresponding business data upon receiving the notification to judge whether the notification has been disposed, and in case it hasn't been dispose, it will be disposed otherwise, and if it has been disposed, **the pure character string “success”** will be returned directly. Before status examination and disposal of business data are conducted, data lock should be adopted for concurrency control to avoid data chaos caused by **repeated data interposition** in the function.

6 Payment Interface

6.1 Initial Request API

6.1.1 Business function

Initial JSAPI request, generate token_id for interactive validation.

6.1.2 Interactive mode

Request: Background request interaction mode

Return&Notification: Background request interaction mode + Background notifies interaction mode

6.1.3 Request Parameters

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

POST request with content of XML

Field Name	Required	Type	Description
Normal Parameters			
service	Yes	String(32)	Interface type. Value : pay.weixin.jspay
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.
is_raw	No	String(1)	"1" : original ecology mode; "0" or Null : non-original ecology mode

			The field value is '1' if its mini-program
out_trade_no	Yes	String(32)	The unique trade reference(merchant order id of 5-32 bits) in merchant' s system. Letter, number and underline are allowed. Case-sensitive
device_info	No	String(32)	Specifies a Terminal device id.
body	Yes	String(127)	Commodity description.
sub_openid	Yes	String(128)	It is the only user identification under the current sub_appid.(Only required with production environment. For more information, see section 9 Notes no.7)
sub_appid	Yes	String(32)	Specifies an Official Account ID or Mini program ID assigned by WeChat.
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.aspThe url only effective in non-original ecology mode.
callback_url	No	String(255)	The redirect URL after payment action. Should be absolute path and ensure platform accessible. i.e. http://wap.tenpay.com/callback.asp. Notes: The URL only used for redirect, the final payment result should be received from notify_url.The url only effective in non-original ecology mode.
time_start	No	String(14)	Order created date. Format : yyyyMMddHHmmss. i.e.20091225091010. GMT+8 BeijingTime
time_expire	No	String(14)	Order timeout date. Format : yyyyMMddHHmmss. i.e.20091225091010. GMT+8 BeijingTime
goods_tag	No	String(32)	Specifies the label of goods, which is a parameter in the coupon feature for businesses.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
limit_credit_pa	No	String(32)	Restrict consumer pay in credit card.

y			<p>“1” : forbidden using credit card.</p> <p>“0” or null : permit using credit card.</p>
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.

6.1.4 Response result

Data return in real time with XML format

Field Name	Required	Type	Description
appid	Yes	String(32)	Specifies Official Account ID assigned by WeChat
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	<p>SHA256: 'SHA256'</p> <p>RSA: 'RSA_1_256'</p>
status	Yes	String(16)	<p>“0” : success. Others value : fail.</p> <p>Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.</p>
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 section 4 'Digital Signature'.

The following fields will returned when status and result_code both are "0"			
token_id	Yes	String(64)	The pre-pay ID reference of platform system. Using in interactive API.
pay_info	No	String	Return by character string with json format. When the is_raw is '1'. Using in primitive js payment. And when the is_minipg is '1'. Using in mini program payment.

6.2 Primitive js payment API

6.2.1 Usage example

The API needs to be paid attention to: All request parameters are character string type! It needs to be paid attention to using in the weakly typed languages like JavaScript, PHP.

The sample code is as follows:

```

WeixinJSBridge.invoke('getBrandWCPayRequest',{

"appId" : "wx2421b1c4370ec43b", // name of the official accounts, to be introduced by the commercial tenant

"timeStamp":" 1395712654", //time stamp, the number of seconds since 1970

"nonceStr" : "e61463f8efa94090b1f366cccfbbb444", //random string

"package" : "prepay_id=u802345jgfjsdfgsdg888",

"signType" : "SHA256", //WeChat signature mode

"paySign" : "70EA570631E4BB79628FBCA90534C63FF7FADD89" //WeChat signature

},function(res){

if(res.err_msg == "get_brand_wcpay_request:ok" ) {}

// The above mode is adopted to judge the front-end return, and the WeChat team solemnly reminds that: res.err_msg will return ok after the payment of

the user is successful, but it couldn't be guaranteed that it is absolutely dependable.

});

```

Definition of getBrandWCPayRequest parameters and returned value

6.2.2 Interactive mode

Request:Background request interaction mode

Return&Notification:Background request interaction mode + Background notifies interaction mode

6.2.3 Request Parameters

Field Name	Required	Type	Description
appld	Yes	String	Reference to the value of request field in pay_info
timeStamp	Yes	String	Reference to the value of request field in pay_info
nonceStr	Yes	String	Reference to the value of request field in pay_info
package	Yes	String	Reference to the value of request field in pay_info
signType	Yes	String	Reference to the value of request field in pay_info
paySign	Yes	String	Reference to the value of request field in pay_info

Response result

Return	Description
err_msg	get_brand_wcpay_request:ok --Payment successful
	get_brand_wcpay_request:cancel --Payment cancel
	get_brand_wcpay_request:fail --Payment failure

Notes: get_brand_wcpay_request:ok, the returned result of JS API shall only be returned when users successfully complete payment. Because of the complex front-end interaction, get_brand_wcpay_request:cancel or get_brand_wcpay_request:fail could be uniformly be disposed as errors occurring to users or initiative abandonment of users, without the need of refined differentiation.

Notes: The URL address of request for primitive page from the commercial user has to provide payment authorization catalogue to be configured well by the service provider, and the payment couldn't be invoked on the test-version official accounts provided by WeChat (when test is being made, the invocation of payment could be made in the mobile WeChat document transmission assistant).

6.3 Official accounts JS payment API

6.3.1 Business function

Initiate JSAPI request, and conduct interactive verification by generating token_id.

If the primitive js payment is used in invoking, this interface could ignored.

6.3.2 Interactive mode

Request:Background request interaction mode

Return&Notification:Background request interaction mode + Background notifies interaction mode

6.3.3 Request Parameters

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

The request parameters should be http queryString, like :

https://gateway.wepayez.com/pay/jsIntl?token_id=xxx,

i.e. :

https://gateway.wepayez.com/pay/jsIntl?token_id=9a0610bc519e782e6275e8c7dd94a445

Click this link in the service account could invoke payment (the user's clicking the WeChat payment button in the page is actually clicking this link, and this mode also needs to configure the payment authorization catalogue: <https://gateway.wepayez.com/pay/>. But it not needs to obtain the following operations of those parameters like the primitive jsapi payment, and when

test is being made, this assembled link could be placed in the mobile WeChat document transmission assistant to be clicked for invocation of payment page).

Field Name	Required	Type	Description
token_id	Yes	String(64)	The pre-pay ID reference of platform system. Using in interactive API.

Notes : <https://gateway.wepayez.com/pay/jspay> the request address has to be opened in the official account supporting WeChat payment to invoke WeChat payment, and the test-version official account provided by WeChat couldn't invoke payment.

6.4 Notification of JS pay

6.4.1 Notification request parameters

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

When interaction of the backstage notification is being made, if the response received by platform from the commercial tenant is not pure character string "success" or if the response is given after 5 seconds, then it will be deemed as unsuccessful by platform, and platform will adopt certain strategies (the interval of notification: 0/15/15/30/180/1800/1800/1800/1800/3600 Unit: seconds) to intermittently re-initiate notification to do its best to improve the rate of success of notification, but the final success of notification will not be guaranteed.

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

The method recommended by platform is to first examine the status of the corresponding business data upon receiving the notification to judge whether the notification has been disposed, and in case it hasn't been dispose, it will be disposed otherwise, and if it has been

disposed, the pure character string success will be returned directly. Before status examination and disposal of business data are conducted, data lock should be adopted for concurrency control to avoid data chaos caused by repeated data interposition in the function.

Notices: After the backstage of the commercial tenant has received the parameters of notification. It's required to verify out_trade_no and total_fee in notification received according to the order number and amount of order of their own business system. And the status of order in the database will only be allowed to update after the verification is consistent.

The backstage notification will return by POST mode through notify_url which sent in payment request.

(Notify in XML format)

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. 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)	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 section 4 'Digital Signature'.
The following fields will returned when status and result_code both are "0"			
openid	Yes	String(128)	It is the only user identification under the current appid.
trade_type	Yes	String(32)	Value :pay.weixin.jspay
is_subscribe	Yes	String(1)	Specifies whether the payer follows the associated official account or not, with "Y" meaning 'follows' and "N" meaning 'not follows'.
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)	Vendor order number
sub_is_subscribe	No	String(1)	Specifies whether the payer follows the associated official account or not, with "Y" meaning 'follows' and "N" meaning 'not follows'.
sub_appid	No	String	Specifies an Official Account ID assigned by WeChat.
sub_openid	No	String(128)	The user id of the Payer provided by the WeChat system in OpenID format as unique tag on vendor's appid. Also it is unique to each appid instance.
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 for a transaction. The unit of the fee is the minimal unit of the currency
coupon_fee	No	Int	Coupon amount. coupon_fee <= total_fee. total_fee - coupon_fee = cash pay amount
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	Yes	String(16)	String states bank_type
bank_billno	No	String(32)	Bank order number. Null for wechat payment.
time_end	Yes	String(14)	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. (GMT+8 Beijing).
cash_fee	Yes	Int	Cash amount of order
cash_fee_type	No	String(16)	Currency type, Complies with ISO 4217 standards, CNY
rate	Yes	String(16)	Exchange rate between user payment currency and merchants settlement currency

6.4.2 Response of notification

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

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

6.5 Retrieve transaction result interface

6.5.1 Business function

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

6.5.2 Interactive mode

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
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	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.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.

6.5.4 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)	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 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 section 4 'Digital Signature'.
The following fields will returned when status and result_code both are "0"			
trade_state	Yes	String(32)	SUCCESS:Payment successful REFUND:Order to be refunded NOTPAY:Order not paid CLOSED:Order closed REVERSE:order reversed REVOK:order revoked
The following fields will returned when trade_state is "SUCCESS"			
trade_type	Yes	String(32)	Value :pay.weixin.jspay

openid	Yes	String(128)	It is the only user identification under the current appid.
is_subscribe	Yes	String(1)	Specifies whether the payer follows the associated official account or not, with "Y" meaning 'follows' and "N" meaning 'not follows'.
transaction_id	Yes	String(32)	The unique trade reference of platform system.
out_transaction_id	No	String(32)	The unique trade reference of vendor system. (Return in successful transaction, others will not)
out_trade_no	Yes	String(32)	The unique trade reference of merchant system.
total_fee	Yes	Int	The total amount of the transaction. The unit of the fee is the minimal unit of the currency
coupon_fee	No	Int	Coupon amount. coupon_fee <= total_fee. total_fee - oupon_fee = cash pay amount
fee_type	No	String(8)	Complies with ISO 4217 standards.
attach	No	String(127)	Specifies merchant's data package, which is returned as it is.
bank_type	Yes	String(16)	String states bank_type
bank_billno	No	String(32)	Bank order number. Null for wechat payment.
time_end	Yes	String(14)	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10.(GMT+8 Beijing).
cash_fee	Yes	Int	Cash amount of order
cash_fee_type	No	String(16)	Currency type, Complies with ISO 4217 standards, CNY
rate	Yes	String(16)	Exchange rate between user payment currency and merchants settlement currency

6.6 Refund interface

6.6.1 Business function

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

I. Refund mode

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

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

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

II. Restrictions of refund

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

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

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

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

6.6.2 Interactive mode

Background interaction mode

6.6.3 Request Parameters

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

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Value : unified.trade.refund
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	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 for a transaction. The unit of the fee is the minimal unit of the currency.
refund_fee	Yes	Int	Refund amount . The unit of the fee is the minimal unit of the local Currency. Partial refund can be supported.
op_user_id	Yes	Int	Specifies the Operator ID. This field shows mch_id by default.
refund_channel	No	String(16)	Value : ORIGINAL. The money will refund back to where it came from.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.

6.6.4 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)	Value : UTF-8
sign_type	No	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)	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 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)	The unique trade reference of merchant system.
out_refund_no	Yes	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id	Yes	String(32)	Specifies the internal refund number, which is unique in the platform system.
refund_channel	Yes	String(16)	Value : ORIGINAL. The money will refund back to where it came from.
refund_fee	Yes	Int	Refund amount. The unit of the fee is the minimal unit of the local currency. Partial refund can be supported.
coupon_refund_fee	No	Int	Coupon refund amount. coupon_refund_fee <= refund_fee. refund_fee - coupon_refund_fee = cash refund amount

6.7 Retrieve refund result interface

After the refund application is submitted, the interface could be invoked to inquire the status of refund. The refund has a certain extent of time delay, and please inquire the status of refund once again after 3 working days.

6.7.1 Request Parameters

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

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Value :unified.trade.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 section 4 'Digital Signature'.

6.7.2 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)	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)	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 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)	The unique trade reference of merchant system.
refund_count	Yes	Int	Specifies recorded refund counts.
out_refund_no_\$n	Yes	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id_\$n	Yes	String(32)	Specifies the internal refund number, which is unique in the platform system.
refund_channel_\$n	Yes	String(16)	Value : ORIGINAL. The money will refund back to where it came from.
refund_fee_\$n	Yes	Int	Refund amount. The unit of the fee is the minimal unit of the local

			currency. Partial refund can be supported.
coupon_refund_fee_{\$n}	No	Int	Coupon refund amount. coupon_refund_fee <= refund_fee. refund_fee - coupon_refund_fee = cash refund amount
cash_fee	Yes	Int	Cash amount of order.
cash_fee_type	No	String(16)	Currency type, Complies with ISO 4217 standards, CNY
rate	Yes	String(16)	Exchange rate between user payment currency and merchants settlement currency
refund_time_{\$n}	No	String(14)	Format : yyyyMMddHHmmss
refund_status_{\$n}	Yes	String(16)	Refund Status: SUCCESS: Refunded successfully . FAIL: Refund failed . PROCESSING: Refund is pending . NOTSURE: Require the Vendor to call the Submit Refund API again with the original refund number. CHANGE: Refund can't be processed as the Payer's bank card is either revoked or blocked. As a consequence, the refund will be transferred to the merchant's cash account. In this case, the refund must be processed offline via the help of the merchant's customer.
<p>{\$n} is the record number. It can be 0~(refund_count -1). Example : There are 2 records of refund_count. The first record number should be "0" and the second one should be "1".</p>			

6.8 Close Order Interface

6.8.1 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.

6.8.2 Interactive mode

Request:Background interaction mode

Response:Background interaction mode

6.8.3 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)	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 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'.

6.8.4 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)	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"for SUCCESS. others for FAIL. SUCCESS indicates the order was cancelled for successfully and cannot be paid for again. FAIL refers to exceptions that occur in the interface. The recall function should be used to determine whether the order has been canceled or not.
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'.

7 Notes

The unit of any related amount is the minimal unit of the local currency, and decimals are not allow.

1. notify_url means that platform server directly initiates request from back end to commercial tenant's server, and when disposing, the commercial tenant could not examine the user's cookie or session; the commercial tenant's updating of DB and other goods delivery

procedure needs to be made after notify_url is completed to ensure that to supplement order after order fails will be successful.

2. notify_url means that it might have repeated notification and the commercial tenant needs to do away with the repeated ones to avoid repeated goods delivery.
3. notify_url means the receive notification, and if the commercial tenant disposes successfully or if the examined order has been disposed, then the successful disposal mark, the pure character string success needs to be returned, and the character string success is not case sensitive; if we don't receive the returned success, then our server will keep sending notification to you until three hours later; if it's assumed that all orders don't return success, then the load of notification of our server will be increase, and the worst case is that the notification normally sent to the commercial tenants might delay; besides, we will urge you to perfect, and if you don't improve for a long time period, then the R&D or operation and maintenance technology staff will adopt control measures over the payment interface opened by your company. For the parameters sent in requesting interface in the document, if the one required to fill is given a yes, then it has to be sent (in case of lack, an error will be alarmed), and if the one required to fill is given a no, then it will be optional to be sent.
4. For the parameters sent in requesting interface in the document, if the one required to fill is given a yes, then it has to be sent (in case of lack, an error will be alarmed), and if the one required to fill is given a no, then it will be optional to be sent.
5. For the returned parameters, if the one required to fill is given a yes, then it has to be returned, and if the one required to fill is given a no, then it will be optional to be returned. Because of upgrading or configuration or other cases, the actually returned parameters might not be totally consistent with those in the document, and the actually received parameters shall prevail.
6. As for user opened: after the followers and the official account have information interaction, the official account could obtain the follower's OpenID (the encrypted WeChat account, and each user's OpenID for each official account is exclusive. For different official accounts, the same user's openid will be different), and to obtain openid could refer to the following address: https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140842. To use

the test-version commercial tenant account needs not to send the user's openid; to switch into the official commercial tenant account needs to obtain openid, and it's required to add sub_openid field in the requested parameters and to send the obtained openid to sub_openid. Before switching into the official commercial tenant account and sending sub_openid parameter, it's required to provide official commercial tenant account and official account (the service account) and the appid is to be configured by service providers(provide configuration table from email), and if there is no configuration, the error of sub_appid and su_openid not match will be reported, which will cause the invocation of interface to be abnormal.

7. Other notes

(1) The problem of capital and small letter of parameters

Please pay attention to the problem of capital and small letter required in the document, such as "after signature, the character of the character string needs to be converted to uppercase".

(2) The problem of format of parameters

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

(3)The problem of timestamp

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

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

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

8. Notes: after an order is generated, to immediately invoke the order closing interface will not be allowed, because the shortest invocation interval is 5 minutes.