



# Merchant API Documentation

QR Code Payment API Specification  
(WeChat/Alipay/UnionPay)

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.....                      | 5  |
| 2.1   | Industry Background.....                   | 5  |
| 2.2   | Business realization process.....          | 5  |
| 2.2.1 | Business of QR code pay.....               | 5  |
| 3     | Data format.....                           | 9  |
| 3.1   | Request Data.....                          | 9  |
| 3.2   | .....                                      | 9  |
|       | XML data format.....                       | 9  |
| 4     | Digital Signature.....                     | 11 |
| 4.1   | Original string of signature.....          | 11 |
| 4.2   | Method of signature.....                   | 12 |
| 4.2.1 | MD5 signature.....                         | 13 |
| 4.2.2 | SHA256 signature.....                      | 14 |
| 4.2.3 | RSA signature.....                         | 15 |
| 5.    | Mechanism to Supplement Order.....         | 18 |
| 6.    | Payment Interface.....                     | 19 |
| 6.1   | QR code pay.....                           | 19 |
| 6.1.1 | Business function.....                     | 19 |
| 6.1.2 | Interactive mode.....                      | 19 |
| 6.1.3 | Request Parameters.....                    | 19 |
| 6.1.4 | Response parameters.....                   | 22 |
| 6.2   | Notification interface.....                | 23 |
| 6.2.1 | Notification request parameters.....       | 23 |
| 6.2.2 | Response of notification.....              | 27 |
| 6.3   | Retrieve transaction result interface..... | 28 |
| 6.3.1 | Business function.....                     | 28 |
| 6.3.2 | Request Parameters.....                    | 28 |
| 6.3.3 | Response result.....                       | 29 |
| 6.4   | Refund interface.....                      | 32 |
| 6.4.1 | Business function.....                     | 32 |
| 6.4.2 | Request parameters.....                    | 33 |
| 6.4.3 | Response parameters.....                   | 35 |
| 6.5   | Retrieve refund result interface.....      | 37 |
| 6.5.1 | Request parameters.....                    | 37 |
| 6.5.2 | Response parameters.....                   | 38 |
| 6.6   | Close Order Interface.....                 | 41 |

|                                |    |
|--------------------------------|----|
| 6.6.1 Business function .....  | 41 |
| 6.6.2 Interactive mode.....    | 41 |
| 6.6.3 Request parameters ..... | 41 |
| 6.6.4 Response parameters..... | 42 |
| 7. Notes .....                 | 43 |
| 8. Error Code .....            | 45 |
| 8.1 Swiftpass error code.....  | 45 |
| 8.2 WeChat error code.....     | 48 |
| 8.3 Alipay error code.....     | 49 |
| 8.4 UnionPay error code .....  | 55 |

| Document Changes |         |   |
|------------------|---------|---|
| Date             | Version | Description   |
| 2018-02-24       | 1.0     | First draft   |
| 2018-06-01       | 1.0     | Request URL is changed to <a href="https://gateway.wepayez.com/pay/gateway">https://gateway.wepayez.com/pay/gateway</a>   |
| 2019-03-15       | 1.1     | Integrate Alipay/WeChat Pay/Unionpay  |
| 2019-07-18       | 1.2     | Add retrieve interface  |
| 2022-05-24       | 1.4.3   | Add Error Code  |
| 2022-06-02       | 1.4.4   | Add channel_id; Improve the description of mch_id;  |
| 2022-06-20       | 1.4.5   | Revised interface version   |
| 2022-07-08       | 1.5     | Update description of fields, device_info, time_start, time_expire, trade_state, etc.   |
| 2022-07-08       | 1.5     | Add gateway error code  |
| 2022-07-08       | 1.5     | Added the description of the recommended query times of the order query API   |
| 2022-08-16       | 1.5.1   | Added swiftpass gateway request timeout description   |
| 2022-08-16       | 1.5.1   | Add parameter "sign_agentno"  |
| 2022-08-16       | 1.5.1   | Modify the description of the parameter "Body"  |
| 2022-09-26       | 1.5.2   | Added the Demo of Aggregator Mode and deleted "channel id"  |
| 2022-12-30       | 1.5.3   | Update fields trade_type, order_fee   |
| 2023-01-03       | 1.5.3   | Update the field fee_type, cash_fee   |
| 2023-01-06       | 1.5.3   | Update the field cash_fee, rate   |
| 2023-01-12       | 1.5.3   | Updated the description of the business function for the close order interface  |
| 2023-02-24       | 1.5.3   | Updated the diagram, signature sample message of SHA256.<br>Added MD5 signature method;<br>Updated the following fields: fields charset, sign_type, out_trade_no, device_info, attach, notify_url, err_code, err_msg, out_trade_no, transaction_id, time_end, refund_id, out_trade_refund, time_start, time_expire; |

# 1 Introduction

## 1.1 Abstract

QR Payment is a payment method which means merchants integrate with the Third Party's API and generate a QR Code to show customers for each transaction. Then customers use WeChat/Alipay/Unionpay to scan the QR Code generated by merchants to complete the payment.



QR Pay can be applied in both on-line and off-line payment scenes. E.g. off-line vending machine with screen, off-line cashier, on-line mall on PC 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 Business of QR code pay

##### Usage scenario:

The payer scans a QR Code shown by vendors in the following process.

Step (1): Merchants create a QR Code for their products correspondingly based on payment rules, as shown in Figure 2.1.

Step (2): The payer scans the QR Code with their WeChat to access the Merchants' product data and proceeds with the transaction, as shown in Figure 2.2. The Payer then makes their payment as instructed, as shown in Figure 2.3.



Figure 2.1 Payment QR Code

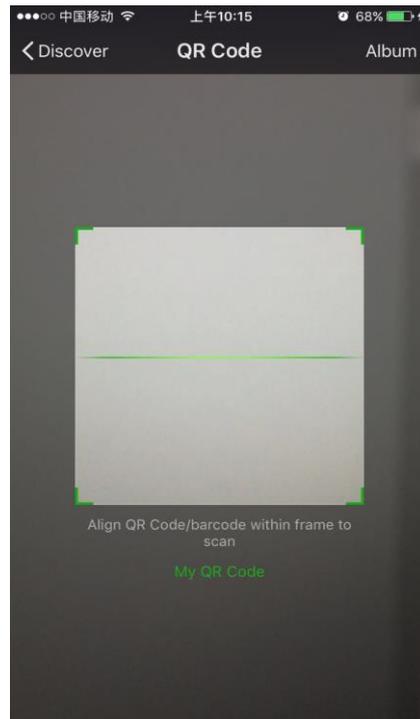


Figure 2.2 Scan QR Code on WeChat

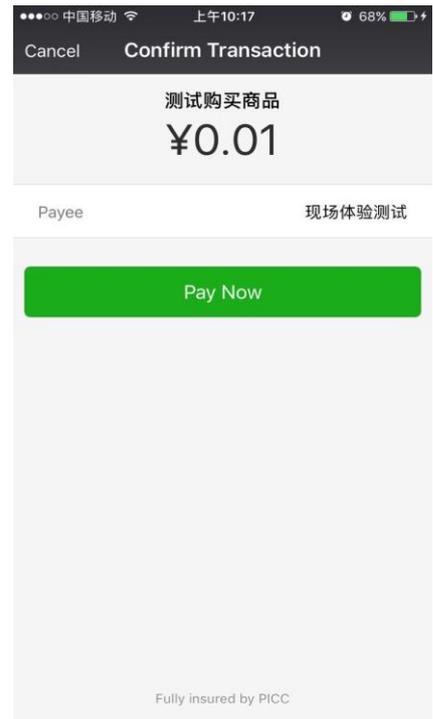


Figure 2.3

Confirm Payment

Step (3): The payer confirms their transaction and enters their payment password, as shown in Figure 2.4.

Step (4): The payer is prompted of a successful payment after completing their payment, as shown in Figure 2.5. The merchant delivers the paid products to the payer after receiving a notification about the successful payment.



Figure 2.4 Enter Payment

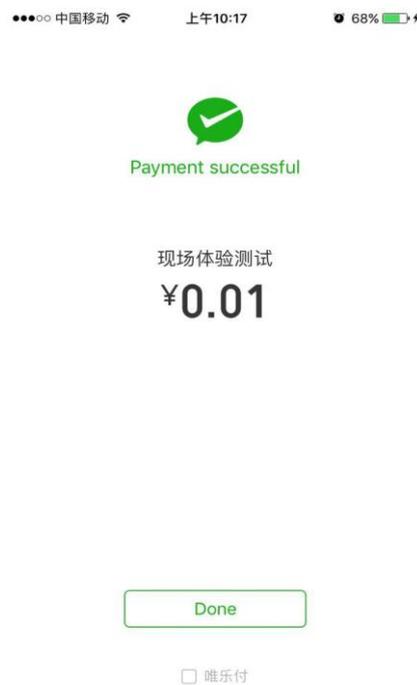
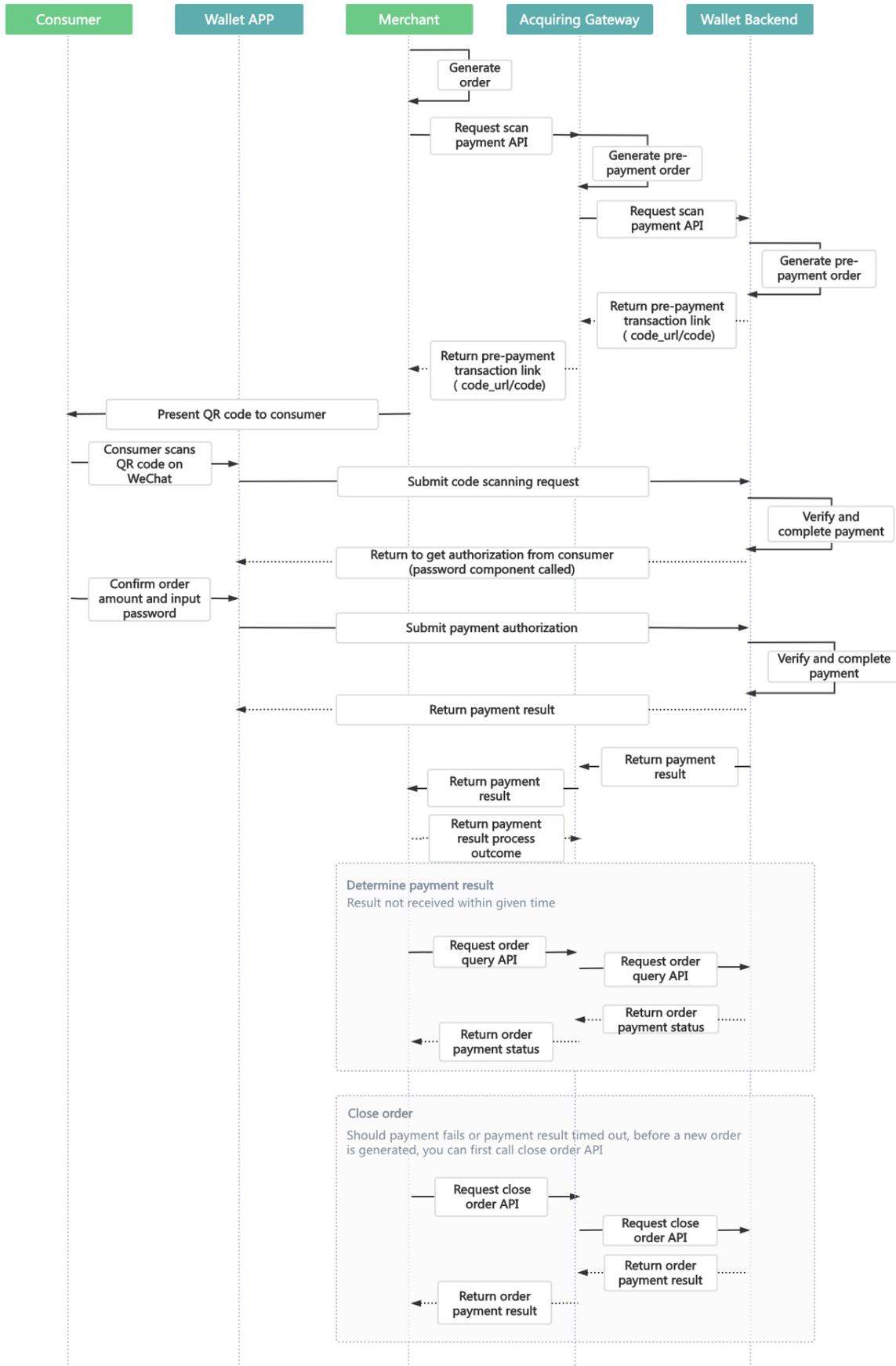


Figure 2.5 Prompt after Successful

**Time sequence of interface calls:**

After the merchant generates an order, the QR code pay interface will be invoked to obtain code\_url, and this URL will generate QR code, and after the user scans the QR code, the payment could be invoked.



## 3 Data format

### 3.1 Request Data

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

```
<xml>

  <mch_id>7551000001</mch_id>

  <out_trade_no>PRDT4CAVMCIY247-T4CAVMCIY247</out_trade_no>

  <nonce_str>167755100000104437</nonce_str>

  <sign>1ABFAE28486FA4BDC698337A2F6EB9A4</sign>

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

  <body>Tickets</body>

  <attach>T4CAVMCIY247</attach>

  <total_fee>496650</total_fee>

  <mch_create_ip>103.43.162.161</mch_create_ip>

  <notify_url>https://www.xxxxx.com/Update</notify_url>

  <time_start>20230210175644</time_start>

  <time_expire>20230210180644</time_expire>

</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[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>
```

## 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>
<body><![CDATA[Tickets ]]></body>
<mch_create_ip><![CDATA[103.43.162.161]]></mch_create_ip>
<mch_id><![CDATA[7551000001]]></mch_id>
<nonce_str><![CDATA[167755100000104437]]></nonce_str>
<notify_url><![CDATA[https://www.xxxxxx.com/Update]]></notify_url>
<out_trade_no><![CDATA[PRDT4CAVMCIY247-T4CAVMCIY247]]></out_trade_no>
<service><![CDATA[pay.weixin.native.intl]]></service>
<sign><![CDATA[EF1D35BE0ABD975915196EC515E90CF3]]></sign>
<total_fee><![CDATA[496650]]></total_fee>
< time_start ><![CDATA[20230210175644]]></ time_start >
< time_expire ><![CDATA[20230210180644]]></ time_expire >
</xml>
```

The signature field sequence:

```
body=
Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=https://www.xxxxxx.com/Update&out_tra
de_no= PRDT4CAVMCIY247-T4CAVMCIY247&service=pay.weixin.native.intl &total_fee=496650& time_start=20230210175644&
time_expire=20230210180644
```

## 4.2 Method of signature

MD5/ SHA256 / RSA signature is 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>

  <mch_id>7551000001</mch_id>

  <out_trade_no>PRDT4CAVMCIY247-T4CAVMCIY247</out_trade_no>

  <nonce_str>167755100000104437</nonce_str>

  <sign>EF1D35BE0ABD975915196EC515E90CF3</sign>

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

  <body>Tickets</body>

  <attach>T4CAVMCIY247</attach>

  <total_fee>496650</total_fee>

  <mch_create_ip>103.43.162.161</mch_create_ip>

  <notify_url>https://www.xxxxxx.com/Update</notify_url>

  <time_start>20230210175644</time_start>

  <time_expire>20230210180644</time_expire>

</xml>
```

Suppose merchant key is: 9f72151b6592fab3e0c63a1ab3c0877b

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

```
body=Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=https://www.xxxxxx.com/Update&out_trade_no=PRDT4CAVMCIY247-T4CAVMCIY247&service=pay.weixin.native.intl&total_fee=496650&time_start=20230210175644&time_expire=20230210180644
```

ii: sign after process b:

```
sign
=md5(string1&key=9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase
=md5(body= Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=
https://www.xxxxx.com/Update &out_trade_no= PRDT4CAVMCIY247-T4CAVMCIY247&service= pay.weixin.native.intl &total_fee=496650&
time_start=20230210175644& time_expire=20230210180644
&key=
9d101c97133837e13dde2d32a5054abb).toUpperCase()
=" EF1D35BE0ABD975915196EC515E90CF3"
```

## 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>
  <mch_id>7551000001</mch_id>
  <out_trade_no>PRDT4CAVMCIY247-T4CAVMCIY247</out_trade_no>
  <nonce_str>167755100000104437</nonce_str>
  <sign>B6C40961FF67604B9F085CD24312B026BF1F60B17B59C5EEA8A42FA55F841BC9 </sign>
  <service>pay.weixin.native.intl</service>
  <body>Tickets</body>
  <attach>T4CAVMCIY247</attach>
  <total_fee>496650</total_fee>
  <mch_create_ip>103.43.162.161</mch_create_ip>
  <notify_url>https://www.xxxxx.com/Update</notify_url>
  <time_start>20230210175644</time_start>
```

```
<time_expire>20230210180644</time_expire>  
  
<sign_type>SHA256</sign_type>  
  
</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=Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=https://www.xxxxxx.com/Update&out_trade_no=PRDT4CAVMCIY247-T4CAVMCIY247&service=pay.weixin.native.intl&total_fee=496650&time_start=20230210175644&time_expire=20230210180644
```

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

```
sign  
  
=SHA256(string1&key= 9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase  
  
=SHA256(body= Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=https://www.xxxxxx.com/Update &out_trade_no= PRDT4CAVMCIY247-T4CAVMCIY247&service= pay.weixin.native.intl &total_fee=496650&time_start=20230210175644& time_expire=20230210180644  
  
&key=  
  
9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase()  
  
="B6C40961FF67604B9F085CD24312B026BF1F60B17B59C5EEA8A42FA55F841BC9 "
```

### 4.2.3 RSA signature

**RSA signature calculation formula:**

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

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

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

Description: The name of the platform signature algorithm: RSA\_1\_256, corresponding to the standard signature algorithm name: SHA256WithRSA, the length of the RSA key is required to be 2048.

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

#### Example:

There are XML afferent parameters:

```
<xml>

  <mch_id>7551000001</mch_id>

  <out_trade_no>PRDT4CAVMCIY247-T4CAVMCIY247</out_trade_no>

  <nonce_str>167755100000104437</nonce_str>

  <sign>

Gu41L6tPanrzejGiDw8XWPliJAJ0rw+tyBys7Picr3a1B6Di29WXe0OuwjghqtBWrFIRWzzb+2DcwH5sbSwl0D6lyzxpLctxxqVNSWeR7X0R5w/Mtw8Gi0IFq

+IXV5D2RH47IHE5j7pliUGfraJVNUXK8VnFuw41Gf6So+5OdrNy4I5clztglp0Kh6NexglNy77CeXSvs2LP3pcuussirjRd6oVqwwqPR61XVbdVrBhtStLbTUjdT

AT3e9w5Wa1YQ4INvENjd71AxMm2dxADfdDSh0rAGz2H9uyZW1PfedcRxp45gJQBBg48Smmla+jpzf0E0Ay1fAatkEKjDZHmHwqg== </sign>

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

  <body>Tickets</body>

  <attach>T4CAVMCIY247</attach>

  <total_fee>496650</total_fee>

  <mch_create_ip>103.43.162.161</mch_create_ip>

  <notify_url>https://www.xxxxxx.com/Update</notify_url>

  <time_start>20230210175644</time_start>

  <time_expire>20230210180644</time_expire>

  <sign_type> RSA_1_256</sign_type>
```

</xml>

## Merchant RSA Private key:

```
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCkcgwgSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEpO8nMyk+XuYSatHwnU4//m47R+4G2YB4Z6PHsJi
4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64Rjra0rZql23dvPW4vVomMRgRcoPOn0YVWp+M6T5PaFgE4M8dh4IMZz
57gVwOdd08F99Z92f3QgZtEj+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9Uu5v3WgLuuv5QnQYzj/RMr8GD+wrdYiNQJxsaTmE/OEJggsumhD4eYY5
YIRy2EIN504cuYVKU1wOSZgq9oJCynGR0aPuQWx58IHxetAgMBAEECggEAHfEFd8qm2PTE2ITAVEC7F+TcgD84UAz0dZnURtx6YIOz5+LH/zVG6j
uYLJU/Oo5RPAC+iMVS68u2JMCp7zm8F7B3JkrbuHLNHGuR6Q7PQuXN8PkDcOxqDmZ2kPjzI4PZvBZRE0abdug+tmMatGzpGAUJzrWcB/N0oVlvrXp9
PnOqfo/Y5nxmpOFClmJppiS3AL1pftNtQZ09G15CPHDYtpUbXPtD2MjjW4OLxKuPRoHSwUgo6LW9XSwNXfcuK+lzbLL0BhWd9IV/+yCEUEbIN87yxxf
hpQFaAhXj5W+B3YsMOZuK93+XMOpYmw8EpUDMOBOnvwb0NSHUrV2RUAQKBgQDTojnNS1e7+tjPzFtOhGPj1uCBPAEieHAcnPgd80bEiujxMLCn
GaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsfr3CAAu7CadQ49TW+SovAvcy9AZuSVVlwynu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNA5pGYKg8+4M
mmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSHnO/nGT2OuxSTTkc8yrx7qd9SmAxXl5kR238DhMOQOnRBomldmVtAJuJg
rdQt0wXfeVQqshqCUAe/xhEbpSCdbPSZbKZZdpIV0y6O5vXlhxw+1qAvXLcxw46s3R92QQKBgQCIQ+ejywkVPDILHMwSsehwThufkCYWYUbbcV
DowpOe5AMozidNju7MNjg2rLHTsCx/kBzOr+7THNw4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvgVtlGBCh6BW5J
ZcSlmT+ykiOoYfvDRuQKBgCgwoHxnBGFfORoLxE3dhpSk8LT05cbueIBVuZW6UC3+8PeK82AjlBLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBzLBLD7G
m4vJTqW0DknYo5sMXS1JmofcKjBv7nbHXZT3EtJSxpVaOdpca/HpsCuCP3AH2e1yk9sZ3wu6IBYSBAoGACYM60j1CVRNSZxUNRgiwfWzS69ql1eez
Pc7xQEganpVBI9SZcTNp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqsvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbF00mdP4CeyuGP
dvDIMXP4dJdLhMUL4pcJLI0B7gBE=
```

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

```
body=Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=https://www.xxxxxx.com/Update&o
ut_trade_no=PRDT4CAVMCIY247-T4CAVMCIY247&service=pay.weixin.native.intl&total_fee=496650&sign_type=RSA_1_256&time_start=202302101
75644&time_expire=20230210180644
```

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

```
sign=RSA(string1, Merchant RSA Private
```

```
key)==RSA(body=Tickets&mch_create_ip=103.43.162.161&mch_id=7551000001&nonce_str=167755100000104437&notify_url=https://www.xxxxxx.co
```

```
m/Update&out_trade_no=PRDT4CAVMCIY247-T4CAVMCIY247&service=pay.weixin.native.intl&total_fee=496650&sign_type=RSA_1_256&time_start=20230210175644&time_expire=20230210180644,MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCkwcggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEpO8nMyk+xuYSatHwnU4//m47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkJq5G271g+PmrzFp7f40/E9m0qjeL64RJRa0rZql23dvPW4vVomMRgRcoPOn0YWVp+M6T5PaFgE4M8dh4IMZz57gVwOdd08F99Z92f3QgZtEjI+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9U5v3WgLuUv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5YIRy2EIN504cujYVKU1wOSZgq9oJCynGR0aPuQWx58IHxEtAgMBAACggEAHfEFd8qm2PTE2ITAvEc7F+TcgD84IUaZ0dZnURtx6YIOz5+LH/zVG6juYLU/Oo5RPAc+iMVS68u2JMCp7zm8Ft7B3JKrbuHLNHGUR6Q7PQUXN8PkDcOxqDmZ2kPjz4PZvBZRE0abdug+tMatGzpGAUJzrWcB/N0oVlvrXp9PnOqfo/Y5nrxmpOfClmJpplS3AL1pftNtQZ09G15CPHDYtpUbXpTD2MjjW4OLxKuPRoHSUwgo6LW9XSwNXfucK+HbzLL0BhIWD9IV/+yCEUEblN87yxxfhpQFaAhXj5W+B3YsMOZuK93+XMOpYmw8EpUDMOBOnvwb0NSHURv2RUAQKBgQDT0jInNS1e7+tjPzFtOhGPj1uCBPAEieHAcnPgd80bEiujxMLCnGaAvmnTrMu4Xo0e5fAP4F7R6UD+IUfr3CAAu7CadQ49TW+SovAvcy9AZuSVVlwynu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNA5pGYKg8+4MmmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSHnO/nGT2OuxSTTkc8yrx7qd9SmAxXl5kR238DhMOQOnRBomldmVtAJUJgrdQyt0wXfeVQqshqCUaE/xhEbpSCdbPSZbKZZdplV0y6O5vXlhxw+1qAvXLcxw46s3R92QQKBgQCIQ+ejywkVPDILHMWSSehwThufkCYWYUbbbcVDownOe5AMoZidTnju7MNjg2rLHTsCx/kBzOr+7THNw4R7kTiEmg09cO+fu5rHXepGgtg+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvgVtIGBCh6BW5JZcSImT+ykIOoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbuelBVuZW6UC3+8PeK82AjlBLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBzLBD7Gm4vJTqW0DknYo5sMXS1JrnofcKjBv7nbHXZT3EtJSxpVaodpcA/HpsCuCP3AH2e1yk9sZ3wu6lBYSBaGACYM60j1CVRNSZxUNRgiwfWzS69q11eezPc7xQEganpVBI9SZcTNp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqvxg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbfFO0mdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLI0B7gBE=")=Gu41L6tPanrzejGidw8XWPijAJ0rw+tyBys7Picr3a1B6Di29Wxe0OuwjghqtBWrFIRWzzb+2DcwH5sbSwl0D6lyzxpLctxxqVNSWeR7X0R5w/Mtw8Gi0IFq+IXV5D2RH47IHE5j7pliUGfraJVNUXK8VnFuw41Gf6So+5OdrNy4I5clztglp0Kh6Nexglny77CeXSvs2LP3pcuussirjRd6oVqwqPR61XVbdVrBhtStLbTUjdTAT3e9w5Wa1YQ4INvENjd71AxMm2dxADfdDSh0rAGz2H9uyZW1PfedcRxp45gJQBBg48Smmla+jpzf0E0Ay1fAatkEKjDZHmHwqg==
```

## 5. MECHANISM TO SUPPLEMENT ORDER

Notes: in respect of the backstage notification interactive mode, if the response of the merchant 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 merchant system many times. The merchant 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 disposed, 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 QR code pay

#### 6.1.1 Business function

The QR code payment request need to be initialized. The request is for generating QR code to make transactions of QR Code Pay.

#### 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   |
|--------------------------|----------|------------|---|
| <b>Normal Parameters</b> |          |            |   |
| service                  | Yes      | String(32) | Wechat: pay.weixin.native.intl<br>Alipay: pay.alipay.native.intl<br>Unionpay: pay.upi.native.intl |
| version                  | No       | String(8)  | Version number. default value: 2.0  |

|               |     |             |  |
|---------------|-----|-------------|--|
| charset       | No  | String(8)   | Encoding method. Default Value: UTF-8  |
| sign_type     | No  | String(16)  | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5   |
| groupno       | No  | String (32) | Master merchant ID   |
| mch_id        | Yes | String(32)  | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.  |
| out_trade_no  | Yes | String(32)  | Internal order number of the merchant. It should be unique in the merchant system and can only contain number, letter, or underscore. Case sensitive.  |
| device_info   | No  | String(32)  | Specifies a Terminal device id. (You can query based on this field on the swiftpass merchant platform)   |
| body          | Yes | String(128) | Description of merchants' goods. Can upload 128 English, 42 Chinese  |
| attach        | No  | String(127) | Additional merchant info. The original value will be returned in the query response and notification after successful payment. This field is mainly used for custom data of the order carried by the merchant.   |
| total_fee     | Yes | Int         | Integral number is allowed only. The unit of the fee is the minimal unit of the local currency. For example, if the currency of payment is Hong Kong dollars, 1000 means HKD10.00  |
| 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. <a href="http://wap.tenpay.com/tenpay.asp">http://wap.tenpay.com/tenpay.asp</a>  |
| time_start    | No  | String(14)  | Order created date. Format: yyyyMMddHHmmss. i.e.20091225091010. GMT+8 Beijing Time<br><br>(WeChat can set the time range from 1 minute to 2 hours, if not uploading, the default is 10 minutes; Alipay can set the time range from 1 minute to 15 days, if not uploading, the default is 2 hours; UPI: If not uploading, the code will not be expired. |
| time_expire   | No  | String(14)  | Order timeout date. Format: yyyyMMddHHmmss. i.e.20091225091010.  |

|                  |     |             |   |
|------------------|-----|-------------|---|
|                  |     |             | <p>GMT+8 Beijing Time</p> <p>(WeChat can set the time range from 1 minute to 2 hours, if not uploading, the default is 10 minutes; Alipay can set the time range from 1 minute to 15 days, if not uploading, the default is 2 hours; UPI: If not uploading, the code will not be expired.</p> |
| op_user_id       | No  | String(32)  | Specifies the Operator ID. This field shows mch_id by default.  |
| goods_tag        | No  | String(32)  | Specifies the label of goods, which is a parameter in the coupon feature for businesses.  |
| product_id       | No  | String(32)  | This ID contains the product ID as set by the merchant.   |
| nonce_str        | Yes | String(32)  | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.   |
| limit_credit_pay | No  | String(32)  | <p>Restrict consumer pay in credit card.</p> <p>"1": forbidden using credit card.</p> <p>"0" or null: permit using credit card.</p>   |
| sign             | Yes | String(344) | Please refer to the section 4 'Digital Signature'.  |
| sign_agentno     | No  | String(32)  | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.  |

## Demo of Aggregator Mode:

```

<xml>

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

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

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

<mch_create_ip><![CDATA[58.33.106.38]]></mch_create_ip>

<mch_id><![CDATA[181520234234]]></mch_id>

<nonce_str><![CDATA[HFfP43tL2i]]></nonce_str>

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

<out_trade_no><![CDATA[202209261130000000]]></out_trade_no>

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

```

```

<service><![CDATA[pay.weixin.native.int]]></service>

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

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

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

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

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

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

</xml>

```

## 6.1.4 Response parameters

Data return in real time with XML format

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

|  |     |              |  |
|--|-----|--------------|--|
|  |     |              | Merchant ID is valid.  |
| device_info  | No  | String(32)   | Specifies a Terminal device id. (You can query based on this field on the swiftpass merchant platform)   |
| nonce_str  | Yes | String(32)   | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.  |
| err_code   | No  | String(32)   | Reference error code. It will be returned only when result_code is different from 0.   |
| err_msg  | No  | String (128) | Error information description. It will be returned only when result_code is different from 0.  |
| sign   | Yes | String(344)  | Please refer to the section 4 'Digital Signature'.   |
| sign_agentno   | No  | String(32)   | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |
| The following fields will be returned when status and result_code both are "0" |     |              |  |
| code_url   | Yes | String(64)   | This parameter should be used to create a QR Code that is displayed to the Payer later.  |
| code_img_url   | Yes | String(256)  | This parameter should be the QR code image source URL created by code_url.   |
| appid  | No  | String(18)   | Wechat Appid , It will return this parameter in third-party.   |
| uuid   | No  | String(33)   | uuid   |
| account_pan  | No  | string(128)  | panToken for UnionPay transaction  |

## 6.2 Notification interface

### 6.2.1 Notification request parameters

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

When interaction of the backstage notification is being made, if the response received by platform from the merchant 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 merchant system many times. The merchant 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 has not 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 merchant has received the parameters of notification. It' s required to verify out\_trade\_no and total\_fee in notification received according to the order number and amount of order of their own business system. And the status of order in the database will only be allowed to update after the verification is consistent. There is no notification for refund transaction.**

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)  | Encoding method. Default Value: UTF-8 |
| sign_type  | Yes      | String(16) | MD5:MD5                               |

|  |     |              |  |
|--|-----|--------------|--|
|  |     |              | SHA256 : SHA256;<br>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 be returned when status is "0"                       |     |              |  |
| result_code  | Yes | String(16)   | "0": success. Others value: fail.  |
| groupno  | No  | String (32)  | Master merchant ID   |
| mch_id   | Yes | String(32)   | Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.   |
| device_info  | No  | String(32)   | Specifies a Terminal device id. (You can query based on this field on the swiftpass merchant platform)   |
| nonce_str  | Yes | String(32)   | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.  |
| err_code   | No  | String(32)   | Reference error code. It will be returned only when result_code is different from 0.   |
| err_msg  | No  | String (128) | Error information description. It will be returned only when result_code is different from 0.  |
| sign   | Yes | String(344)  | Please refer to the section 4 'Digital Signature'.   |
| sign_agentno   | No  | String(32)   | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |
| The following fields will be returned when status and result_code both are "0" |     |              |  |
| openid   | No  | String(128)  | It is the only user identification under the current WeChat appid or Alipay PID.   |
| trade_type   | Yes | String(32)   | Value:<br>Wechat: pay.weixin.native.intl   |

|                    |     |             |   |
|--------------------|-----|-------------|---|
|                    |     |             | Alipay: pay.alipay.native.intl<br>Unionpay: pay.upi.native.intl   |
| 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”.   |
| pay_result         | Yes | Int         | Payment result.<br>“0”: success.<br>Others: fail.   |
| pay_info           | No  | String(64)  | Payment result information. Payment successful return null.   |
| transaction_id     | Yes | String(32)  | Platform transaction ID, it’s unique ID in platform.  |
| out_transaction_id | Yes | String(32)  | Transaction ID returned by the third-party.   |
| 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 Third Party 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. Integral number is allowed only. The unit of the fee is the minimal unit of the local currency.   |
| coupon_fee         | No  | Int         | Coupon amount. coupon_fee <= total_fee.<br>total_fee - coupon_fee = cash pay amount   |
| order_fee          | No  | Int         | Order amount in Cent  |
| fee_type           | No  | String(8)   | Currency type, 3-digit code under ISO 4217 standard.  |
| attach             | No  | String(127) | Additional merchant info. The original value will be returned in the notification after successful payment. This field is mainly used for custom data of the order carried by the merchant. |
| bank_type          | No  | String(16)  | String states bank_type, returned by third-party.   |

|                        |     |             |  |
|------------------------|-----|-------------|--|
| time_end               | Yes | String(14)  | Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10: 10.GMT+8 Beijing Time. Returned when orders tatus is payment successful or refunded.            |
| cash_fee               | No  | Int         | Cash amount of order, the unit of the fee is the minimal unit of CNY.  |
| cash_fee_type          | No  | String(16)  | Currency type, Complies with ISO 4217 standards.   |
| rate                   | No  | String(16)  | Exchange rate between user payment currency and merchant's settlement currency.  |
| local_total_fee        | No  | Int         | Local total order amount in Cent.  |
| local_fee_type         | No  | String(8)   | Local currency type, three-letter code in accordance with ISO 4217   |
| discount_detail        | No  | String(512) | Offer details, different payment types return different results. For example, UnionPay returns:<br>[{"discountAmt":"98.00","discountNote":"Uplan discount"}, {"discountAmt":"5.00","discountNote":"Instant Discount"}] |
| unsettled_discount_fee | No  | int         | The discount amount that does not participate in settlement, the unit is cents   |
| cost_amount            | No  | int         | Consumer actual payment amount (channel currency), in cents  |
| voucher_no             | No  | int         | QR code voucher number   |
| account_pan            | No  | string(128) | panToken for UnionPay transaction  |

## 6.2.2 Response of notification

platform server will send notification, then post send XML data flow, the notify\_url of the merchant 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#, TestPayResultServlet method in java), and after the merchant 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.

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

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

## 6.3 Retrieve transaction result interface

### 6.3.1 Business function

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

### 6.3.2 Request Parameters

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

POST request with content of XML

| Field Name | Required | Type       | Description  |
|------------|----------|------------|--|
| service    | Yes      | String(32) | Value: unified.trade.query   |
| version    | No       | String(8)  | Version number. default value: 2.0                                   |
| charset    | No       | String(8)  | Encoding method. Default Value: UTF-8                                |
| sign_type  | No       | String(16) | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5 |

|                |     |             |  |
|----------------|-----|-------------|--|
| groupno        | No  | String (32) | Master merchant ID   |
| mch_id         | Yes | String(32)  | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.  |
| out_trade_no   | No  | String(32)  | The unique trade reference of merchant system. At least one of the out_trade_no and transaction_id should be exist in query request; when both exist, transaction_id shall prevail.                                  |
| transaction_id | No  | String(32)  | The unique trade reference of platform system. At least one of the out_trade_no and transaction_id should be exist in query request; when both exist, transaction_id shall prevail.                                  |
| 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'.   |
| sign_agentno   | No  | String(32)  | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |

### 6.3.3 Response result

Data return in real time with XML format

| Field Name | Required | Type       | Description  |
|------------|----------|------------|--|
| version    | Yes      | String(8)  | Version number. default value: 2.0   |
| charset    | Yes      | String(8)  | Encoding method. Default Value: UTF-8  |
| sign_type  | Yes      | String(16) | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5   |
| status     | Yes      | String(16) | "0": success. Others value: fail.<br>Specifies communicating label (not transaction label). The status of a transaction is determined by the value of trade_state. |

|  |     |              |   |
|--|-----|--------------|---|
| message  | No  | String(128)  | Return message. Only return when the signature verification invalid.  |
| The following fields will be returned when status is "0"                       |     |              |   |
| result_code  | Yes | String(16)   | "0": success. Others value: fail.   |
| groupno  | No  | String (32)  | Master merchant ID  |
| mch_id   | Yes | String(32)   | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.   |
| device_info  | No  | String(32)   | Specifies a Terminal device id. (You can query based on this field on the swiftpass merchant platform)  |
| nonce_str  | Yes | String(32)   | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.   |
| err_code   | No  | String(32)   | Reference error code. It will be returned only when result_code is different from 0.  |
| err_msg  | No  | String (128) | Error information description. It will be returned only when result_code is different from 0.   |
| sign   | Yes | String(344)  | Please refer to the section 4 'Digital Signature'.  |
| sign_agentno   | No  | String(32)   | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.  |
| The following fields will be returned when status and result_code both are "0" |     |              |   |
| trade_state  | Yes | String(32)   | <p>SUCCESS: Payment successful</p> <p>REFUND: Order to be refunded</p> <p>NOTPAY: Order not paid</p> <p>CLOSED: Order closed</p> <p>REVOKED: order revoked</p> <p>USERPAYING: consumer paying</p> <p>PAYERROR: Payment failed (payment status failed to be returned by bank or other reasons)</p> |
| The following fields will be returned when trade_state is "SUCCESS"            |     |              |   |

|                    |     |             |   |
|--------------------|-----|-------------|---|
| trade_type         | Yes | String(32)  | Wechat: pay.weixin.native.intl<br>Alipay: pay.alipay.native.intl<br>Unionpay: pay.upi.native.intl   |
| appid              | No  | String      | Specifies an official account ID of institution assigned by WeChat.   |
| sub_appid          | No  | String      | Specifies an official account ID of merchant assigned by WeChat.  |
| openid             | No  | String(128) | It is the only user identification under the current appid.   |
| 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.   |
| 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_is_subscribe   | No  | String(1)   | Specifies whether the payer follows the associated official account or not, with "Y" meaning 'follows' and "N" meaning "not follows".   |
| transaction_id     | Yes | String(32)  | Platform transaction ID, it's unique ID in platform.  |
| out_transaction_id | Yes | String(32)  | Transaction ID returned 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.   |
| order_fee          | No  | Int         | Order amount in Cent  |
| local_total_fee    | No  | Int         | Local total order amount in Cent.   |
| coupon_fee         | No  | Int         | Coupon amount. coupon_fee <= total_fee.<br>total_fee - oupon_fee = cash pay amount  |
| fee_type           | No  | String(8)   | Currency type, 3-digit code under ISO 4217 standard.  |
| attach             | No  | String(127) | Additional merchant info. The original value will be returned in the query response after successful payment. This field is mainly used for custom data of the order carried by the merchant. |
| bank_type          | No  | String(16)  | String states bank_type, returned by third-party.   |

|                        |     |              |  |
|------------------------|-----|--------------|--|
| bank_billno            | No  | String(32)   | Bank order number. Null for WeChat payment.  |
| time_end               | Yes | String(14)   | Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. GMT+8 Beijing. Returned when orders tatus is payment successful or refunded.             |
| cash_fee               | No  | Int          | Cash amount of order, the unit of the fee is the minimal unit of CNY.  |
| cash_fee_type          | No  | String(16)   | Currency type, Complies with ISO 4217 standards.   |
| local_fee_type         | No  | String(8)    | Local currency type, three-letter code in accordance with ISO 4217   |
| rate                   | No  | String(16)   | Exchange rate between user payment currency and merchant's settlement currency.  |
| discount_detail        | No  | String (512) | Offer details, different payment types return different results. For example, UnionPay returns:[{"discountAmt":"98.00","discountNote":"Uplan discount"}, {"discountAmt":"5.00","discountNote":"Instant Discount"}] |
| unsettled_discount_fee | No  | int          | The discount amount that does not participate in settlement, the unit is cents   |
| cost_amount            | No  | int          | The actual payment amount of the consumer (channel currency), the unit is cents  |
| voucher_no             | No  | int          | QR code certificate number   |
| account_pan            | No  | string(128)  | panToken for UnionPay transaction  |

## 6.4 Refund interface

### 6.4.1 Business function

If the merchant 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, the fund of the refund will follow the same route back to customers 'account.

Notes: Generally, the amount will be refunded to the bank account within 1 to 3 working days after the refund is initiated.

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 merchant 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 merchant, and so the merchant has to guarantee the uniqueness of the refund application form. In the process of refund, the merchant 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 merchant could negotiate with the buyer to refund to the WeChat balance account.

Currently only refund interface without key is provided, and in case the merchant needs refund interface with key, please contact the business person.

### **6.4.2 Request parameters**

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

POST request with content of XML

| Field Name     | Required | Type        | Description   |
|----------------|----------|-------------|---|
| service        | Yes      | String(32)  | Value: unified.trade.refund   |
| version        | No       | String(8)   | Version number. default value: 2.0  |
| charset        | No       | String(8)   | Encoding method. Default Value: UTF-8   |
| sign_type      | No       | String(16)  | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5  |
| groupno        | No       | String (32) | Master merchant ID  |
| mch_id         | Yes      | String(32)  | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.   |
| out_trade_no   | No       | String(32)  | The unique trade reference of merchant system. At least one of the out_trade_no and transaction_id should exist in refund request. transaction_id priority when both be filled.   |
| transaction_id | No       | String(32)  | Platform transaction ID, it's unique ID in platform. At least one of the out_trade_no and transaction_id should exist in refund request. transaction_id priority when both be filled.   |
| out_refund_no  | Yes      | String(32)  | Specifies the internal refund number, which is unique in the system. A single transaction can be processed as multiple partial refunds, with the total sum of the partial refunds being equal to the original one. If the refund is not successful. The recall function should be used with same refund number to avoid duplication of refunds. |
| total_fee      | Yes      | Int         | The total amount 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'.   |
| sign_agentno | No  | String(32)  | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |

### 6.4.3 Response parameters

Data return in real time with XML format

| Field Name   | Required | Type        | Description  |
|--|----------|-------------|--|
| version  | Yes      | String(8)   | Version number. default value: 2.0   |
| charset  | Yes      | String(8)   | Encoding method. Default Value: UTF-8  |
| sign_type  | Yes      | String(16)  | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5   |
| status   | Yes      | String(16)  | "0": success. Others value: fail.<br>Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code. |
| message  | No       | String(128) | Return message. Only return when the signature verification invalid.   |
| The following fields will be returned when status is "0" |          |             |  |
| result_code  | Yes      | String(16)  | "0": success. Others value: fail.  |
| groupno  | No       | String (32) | Master merchant ID   |
| mch_id   | Yes      | String(32)  | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.  |
| device_info  | No       | String(32)  | Specifies a Terminal device id. (You can query based on this field on the swiftpass merchant platform)   |

|  |     |              |  |
|--|-----|--------------|--|
| nonce_str  | Yes | String(32)   | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.  |
| err_code   | No  | String(32)   | Reference error code. It will be returned only when result_code is different from 0.   |
| err_msg  | No  | String (128) | Error information description. It will be returned only when result_code is different from 0.  |
| sign   | Yes | String(344)  | Please refer to the section 4 'Digital Signature'.   |
| sign_agentno   | No  | String(32)   | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |
| The following fields will be returned when status and result_code both are "0" |     |              |  |
| trade_type   | Yes | String(32)   | Wechat: pay.weixin.native.intl<br>Alipay: pay.alipay.native.intl<br>Unionpay: pay.upi.native.intl  |
| transaction_id   | Yes | String(32)   | Platform transaction ID, it's unique ID in platform.   |
| 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.   |
| order_fee  | No  | Int          | Order amount in Cent   |
| local_total_fee  | No  | Int          | Local total order amount in Cent.  |
| fee_type   | No  | String(8)    | Currency type, 3-digit code under ISO 4217 standard.   |

|                   |    |           |  |
|-------------------|----|-----------|--|
| coupon_refund_fee | No | Int       | Coupon refund amount.<br>coupon_refund_fee <= refund_fee.<br>refund_fee - coupon_refund_fee = cash refund amount |
| local_fee_type    | No | String(8) | Local currency type, three-letter code in accordance with ISO 4217   |

## 6.5 Retrieve refund result interface

### 6.5.1 Request parameters

Request: <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)   | Encoding method. Default Value: UTF-8  |
| sign_type      | No       | String(16)  | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5   |
| groupno        | No       | String (32) | Master merchant ID   |
| mch_id         | Yes      | String(32)  | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.  |
| out_trade_no   | No       | String(32)  | The unique trade reference of merchant system. At least one of the out_trade_no and transaction_id should be exist in refund query request. transaction_id priority when both be filled. |
| transaction_id | No       | String(32)  | The unique trade reference of platform system. At least one of the out_trade_no and transaction_id should be exist in refund query request. 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. At least one of the refund_id and out_refund_no should be existed in refund query request. refund_id priority when both be filled.   |
| refund_id    | No  | String(32)  | Specifies the internal refund number, which is unique in the platform system. At least one of the refund_id and out_refund_no should be existed in refund query request. 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'.   |
| sign_agentno | No  | String(32)  | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |

## 6.5.2 Response parameters

Data return in real time with XML format

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

|  |     |              |   |
|--|-----|--------------|---|
| mch_id   | Yes | String(32)   | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.   |
| device_info  | No  | String(32)   | Specifies a Terminal device id. (You can query based on this field on the swiftpass merchant platform)  |
| nonce_str  | Yes | String(32)   | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.   |
| err_code   | No  | String(32)   | Reference error code. It will be returned only when result_code is different from 0.  |
| err_msg  | No  | String (128) | Error information description. It will be returned only when result_code is different from 0.   |
| sign   | Yes | String(344)  | Please refer to the section 4 'Digital Signature'.  |
| sign_agentno   | No  | String(32)   | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.                |
| The following fields will be returned when status and result_code both are "0" |     |              |   |
| transaction_id   | Yes | String(32)   | Platform transaction ID, it's unique ID in platform.  |
| 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)   | Merchant refund number. "\$n" stands for refund order count, calculated from 0. Value taken from 0 to (total refund count – 1). E.g. out_refund_no_0, out_refund_no_1, etc.   |
| refund_id_\$n  | Yes | String(32)   | Specifies the internal refund number, which is unique in the platform system. "\$n" stands for refund order count, calculated from 0. Value taken from 0 to (total refund count – 1). E.g. refund_id_0, refund_id_1, etc.           |
| refund_channel_\$n   | Yes | String(16)   | Value: ORIGINAL. The money will refund back to where it came from. "\$n" stands for refund order count, calculated from 0. Value taken from 0 to (total refund count – 1). E.g. refund_channel_0, refund_channel_1, etc.            |
| refund_fee_\$n   | Yes | Int          | Refund amount. The unit of the fee is the minimal unit of the currency. Partial refund can be supported. "\$n" stands for refund order count, value taken from 0 to (total refund count – 1). E.g. refund_fee_0, refund_fee_1, etc. |

|  |     |            |   |
|--|-----|------------|---|
| coupon_refund_fee_{\$n}  | No  | Int        | <p>Coupon refund amount.</p> <p>coupon_refund_fee &lt;= refund_fee.</p> <p>refund_fee - coupon_refund_fee = cash refund amount</p> <p>“\$n” stands for refund order count, value taken from 0 to (total refund count – 1). E.g. coupon_refund_fee_0, coupon_refund_fee_1, etc.</p>  |
| cash_fee   | No  | Int        | Cash amount of order, the unit of the fee is the minimal unit of the currency.  |
| cash_fee_type  | No  | String(16) | Currency type, Complies with ISO 4217 standards   |
| rate   | No  | String(16) | Exchange rate between user payment currency and merchant’s settlement currency.   |
| refund_time_{\$n}  | No  | String(14) | <p>Payment completion time. Format: yyyyMMddhhmmss, e.g. 9:10:10 on December 25, 2009 will be displayed as 20091225091010, the timezone is GMT+8 Beijing. The time is taken from merchant’s server. Note: order creation time and timeout time must both be uploaded to take effect. “\$n” stands for refund order count, value taken from 0 to (total refund count – 1). E.g. refund_time_0, refund_time_1, etc.</p>   |
| refund_status_{\$n}  | Yes | String(16) | <p>Refund Status:</p> <p>SUCCESS: Refunded successfully.</p> <p>FAIL: Refund failed.</p> <p>PROCESSING: Refund is pending.</p> <p>NOTSURE: Require the Vendor to call the Submit Refund API again with the original refund number.</p> <p>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> <p>“\$n” stands for refund order count, value taken from 0 to (total refund count – 1). E.g. refund_status_0, refund_status_1, etc,</p> |
| <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> |     |            |   |

## 6.6 Close Order Interface

### 6.6.1 Business function

If the payment of order of the merchant 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. Close order only applicable for Wechat Pay and Alipay, but not for UPI QR payment. UPI can handle repeat QR payment transaction, so merchant don't need to initiate close order when payment of order fails.

### 6.6.2 Interactive mode

Request: Background interaction mode

Response: Background interaction mode

### 6.6.3 Request parameters

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

POST request with content of XML

| Field Name               | Required | Type       | Description  |
|--------------------------|----------|------------|--|
| <b>Normal Parameters</b> |          |            |  |
| service                  | Yes      | String(32) | Value: unified.trade.close   |
| version                  | No       | String(8)  | Version number. default value: 2.0                                   |
| charset                  | No       | String(8)  | Encoding method. Default Value: UTF-8                                |
| sign_type                | No       | String(16) | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5 |

|              |     |             |  |
|--------------|-----|-------------|--|
| groupno      | No  | String (32) | Master merchant ID   |
| mch_id       | Yes | String(32)  | Specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.  |
| out_trade_no | Yes | String(32)  | The unique trade reference of merchant system.   |
| nonce_str    | Yes | String(32)  | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.  |
| sign         | Yes | String(344) | Please refer to the section 4 'Digital Signature'.   |
| sign_agentno | No  | String(32)  | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |

## 6.6.4 Response parameters

Data return in real time with XML format

| Field Name   | Required | Type        | Description  |
|--|----------|-------------|--|
| version  | Yes      | String(8)   | Version number. default value: 2.0   |
| charset  | Yes      | String(8)   | Encoding method. Default Value: UTF-8  |
| sign_type  | Yes      | String(16)  | MD5: MD5<br>SHA256 : SHA256;<br>RSA: RSA_1_256<br>Default value: MD5   |
| status   | Yes      | String(16)  | "0": success. Others value: fail.<br>Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code. |
| message  | No       | String(128) | Return message. Only return when the signature verification invalid.   |
| The following fields will be returned when status is "0" |          |             |  |
| result_code  | Yes      | String(16)  | "0" for SUCCESS. others for FAIL.<br>SUCCESS indicates the order was cancelled for successfully and cannot be  |

|              |     |              |  |
|--------------|-----|--------------|--|
|              |     |              | paid for again.<br><br>FAIL refers to exceptions that occur in the interface. The recall function should be used to determine whether the order has been canceled or not.  |
| groupno      | No  | String (32)  | Master merchant ID   |
| mch_id       | Yes | String(32)   | Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.   |
| nonce_str    | Yes | String(32)   | Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.  |
| err_code     | No  | String(32)   | Reference error code. It will be returned only when result_code is different from 0.   |
| err_msg      | No  | String (128) | Error information description. It will be returned only when result_code is different from 0.  |
| sign         | Yes | String(344)  | Please refer to the section 4 'Digital Signature'.   |
| sign_agentno | No  | String(32)   | When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key. |

## 7. NOTES

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

5. For the 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. Other notes
  - The problem of capital and small letter of parameters  
Please pay attention to the problem of capital and small letter required in the document.
  - 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.
  - The problem of time stamp  
Please use Linux time stamp, and note that its format is character string.
  - The problem of order number payment made by the same merchant
7. If the payment of order of the merchant fails, then it is 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 the close order interface.

8. Notes: after an order is generated, to immediately invoke the close order interface will not be allowed, because the shortest invocation interval is 5 minutes.
9. Request swiftpass gateway If there is no clear result of synchronization within 10 seconds, it can be considered that the transaction request has timed out

## 8. ERROR CODE

### 8.1 Swiftpass error code

| Error Code         | Error Message  |
|--------------------|--|
| Auth code invalid  | Invalid payment code   |
| 400                | Payment code cannot be blank                                     |
| 400                | Missing pre-order info   |
| 400                | Require xml content  |
| 400                | Require POST method  |
| 400                | Parse xml error  |
| 400                | unsupported sign method  |
| 400                | Signature error  |
| 400                | Parse params error   |
| 400                | Parse xml error, please use UTF-8 encoded                        |
| 400                | XXX: This field is required                                      |
| 400                | total_fee: Invalid value   |
| 400                | Verification failed  |
| INVALID_FEE        | Amount error   |
| INVALID_REFUND_FEE | Refund amount error  |
| REFUND_FEE_LIMIT   | Refund request intercepted, please check if settings are correct |

|   |   |
|---|---|
| REFUND_FEE_INVALID  | Invalid refund amount                   |
| Validation parameter failed                                     | Parse params error                      |
| ORDER_FEE_INVALID   | Order amount invalid                    |
| ORDER_DATE_INVALID  | Order date invalid                      |
| transaction_id out_trade_no is required                         | Missing parameters                      |
| buyer_logon_id buyer_id is required                             | Missing parameters                      |
| transaction_id out_trade_no out_refund_no refund_id is required | Missing parameters                      |
| out_refund_no refund_id is required                             | Missing parameters                      |
| Refundid and refundpwd is required                              | Missing parameters                      |
| thi_mch_id is required  | Missing parameters                      |
| 400   | Sign error                              |
| Refund exists   | Refund already existed                  |
| Order exists  | Order already existed                   |
| Refund not exists   | Refund do not exist                     |
| 400   | Pre-paid order number do not exist      |
| Order not exists  | Order do not exist                      |
| 400   | Missing pre-paid order number           |
| 400   | Order do not exist                      |
| 500   | Internal error                          |
| 500   | SYSTEMERROR                             |
| SYSTEMERROR   | System error                            |
| Internal error  | WeChat request error, URL error         |
| Internal error  | WeChat request error, connection failed |

|                     |  |
|---------------------|--|
| Internal error      | WeChat request error, timeout                      |
| Internal error      | WeChat request error, protocol error               |
| Internal error      | WeChat request error, unknown error                |
| Internal error      | Third-party request error, URL error               |
| Internal error      | Third-party request error, connection failed       |
| Internal error      | Third-party request error, timeout                 |
| Internal error      | Third-party request error, protocol error          |
| Internal error      | Third-party request error, unknown error           |
| Order reverse       | Order is reversed                                  |
| 400                 | Unsupported Api                                    |
| 400                 | The merchant has not opened the [XXX] payment type |
| 400                 | Merchant is suspended                              |
| 400                 | Merchant cannot initiate order alone               |
| 400                 | This order is paid                                 |
| Order paid          | Order already paid                                 |
| 400                 | Order amount exceeds merchant limit                |
| 400                 | Order amount is below merchant limit               |
| Amount limit        | PRODUCT_AMOUNT_LIMIT_ERROR                         |
| BALANCE_NOT_ENOUGH  | Insufficient balance                               |
| USERPAYING          | WeChat requires users to enter a password          |
| Refund status error | Refund status error                                |
| REFUND_ERROR        | REFUND ERROR                                       |
| Order date limit    | Order expired                                      |
| Order status error  | Order status error                                 |

|                        |  |
|------------------------|--|
| Reverse fail           | Reverse failed                         |
| Order close fail       | Failed to close order                  |
| 400                    | This order cannot be reversed          |
| Refund limit           | Do not support this function           |
| 400                    | Order status error, cannot close order |
| ORDER ERROR            | Order failed                           |
| refund frequency high  | Refund too frequent                    |
| 400                    | Request too frequent                   |
| REQUEST CHANGE ERROR   | Do not match with original order       |
| Order trade type error | Cannot match payment type              |

## 8.2 WeChat error code

Please check below link for details.

[https://pay.weixin.qq.com/wiki/doc/api/wxpay/en/fusion\\_wallet/NativePay/chapter4\\_3.shtml](https://pay.weixin.qq.com/wiki/doc/api/wxpay/en/fusion_wallet/NativePay/chapter4_3.shtml)

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

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

### 8.3 Alipay error code

Please check below link for details.

[https://global.alipay.com/docs/ac/gr/error\\_code](https://global.alipay.com/docs/ac/gr/error_code)

| Error code          | Description   |
|---------------------|---|
| ACCESS_FORBIDDEN    | You have no permission to use the product. Check your agreement with Alipay.  |
| AGREEMENT_NOT_EXIST | The related agreement does not exist.   |
| ALIN10023           | An order is being paid by a second payer.   |
| ALIN10070           | The signature is incorrect.   |
| ALIN10129           | In APP payment, the PID does not match the environment. For example, the PID is from production environment but is tested in sandbox environment. |
| ALIN42282           | The same external transaction number, out_trade_no, is used to repeatedly initiate requests, and the request parameters are                       |

|  |  |
|--|--|
|  | inconsistent.  |
| ALIN43877                              | The merchant called Alipay domestic payment solution interface, but the buyer is trying to make payment in a foreign country (outside of China). |
| ALIN43911                              | Transaction risks are detected at the user side.   |
| BEYOND_PAY_RESTRICTION                 | The balance payment amount exceeds the limit. Change to other payment method such as cash or credit card.  |
| BUYER_BALANCE_NOT_ENOUGH               | Insufficient balance for current operation. Top-up the balance or link a bank card.  |
| BUYER_BANKCARD_BALANCE_NOT_ENOUGH      | Insufficient bank balance for current operation. Top-up the balance or link a bank card.   |
| BUYER_ENABLE_STATUS_FORBID             | The buyer account is disabled because of identity or security reasons.   |
| BUYER_NOT_EXIST                        | The buyer_identity_code value is incorrect.  |
| BUYER_NOT_MAINLAND_CERT                | The user is not an eligible customers of the Alipay payment products that you use.   |
| BUYER_PAYMENT_AMOUNT_DAY_LIMIT_ERROR   | The total payment amount for the buyer exceeded user daily max-amount limit. Change to other payment method such as cash or credit card.         |
| BUYER_PAYMENT_AMOUNT_MONTH_LIMIT_ERROR | The total payment amount for the buyer exceeded user monthly max-amount limit. Change to other payment method such as cash or credit card.       |
| CONTACT_NO_MANDATORY                   | contact_no is not passed in the request parameters.  |
| CONTEXT_INCONSISTENT                   | The transaction information is inconsistent by the same out_trade_no. Check the request data.  |
| CURRENCY_NOT_SAME                      | The refund currency is inconsistent.   |
| CURRENCY_NOT_SUPPORT                   | The currency is not supported. Check the agreement with Alipay.  |
| DATA_NOT_EXIST                         | The secondary merchant is not successfully registered.   |
| DISCORDANT_REPEAT_REQUEST              | The same out_return_no is used to initiate multiple refund requests, and the request parameters are inconsistent.                                |

|                                   |   |
|-----------------------------------|---|
| DUPLICATE_PAY_CURRENCY_NOT_EQUAL  | More than one payment requests share a same out_trade_no, and the parameter currency is not the same.   |
| DUPLICATE_REQUEST                 | The registration request for the same PM / PMS is submitted already and is currently under review.  |
| EXCHANGE_AMOUNT_OR_CURRENCY_ERROR | The exchange amount or currency is incorrect. Check the amount and the currency parameter.  |
| EXIST_FORBIDDEN_WORD              | Prohibited words are included in the transaction request.   |
| HAS_NO_PRIVILEGE                  | The contract of the PID expires or is terminated.   |
| ILLEGAL_ACCESS_SWITCH_SYSTEM      | You are not allowed to access the system of this type. Contact Alipay Technical Support if needed.  |
| ILLEGAL_ARGUMENT                  | The parameter is incorrect. Check each request parameter according to the API specification.  |
| ILLEGAL_CLIENT_IP                 | The client IP address is illegal. Check the Client IP address.  |
| ILLEGAL_EXTERFACE                 | The Alipay API you call is incorrect.   |
| ILLEGAL_PARTNER                   | The partner ID is incorrect. Ensure the value of partner parameter matches the partner value provided by Alipay.                                |
| ILLEGAL_PARTNER_EXTERFACE         | The PID used in the request did not sign the corresponding contract.  |
| ILLEGAL_SECURITY_PROFILE          | The matching private key configuration is not found.  |
|                                   | For the solution, see ILLEGAL_SECURITY_PROFILE.   |
| ILLEGAL_SIGN                      | Errors exist in the signature.  |
| ILLEGAL_SIGN_TYPE                 | Alipay's request only supports three signing methods: MD5, RSA, and RSA2. If other values are passed to sign_type, this error will be returned. |
|                                   | Please check if sign_type is one of MD5, RSA, RSA2, and check if the value is passed incorrectly.   |
| INVALID_PARAMETER                 | The parameter name is wrong, the parameter value does not meet the requirements, and required parameters are not filled, and so on.             |

|                                    |   |
|------------------------------------|---|
| INVALID_RECEIVE_ACCOUNT            | Error exists in the receipt account information.  |
| INVALID_ROUNDED_AMOUNT             | The refund with this amount may violate the rule that the calculated amount of both CNY and foreign currency should be fully or not fully refunded at the same time. Take a transaction with 0.07 CNY (0.01 USD) as an example. A refund with 0.06 CNY would not be accepted because this comes to a situation where there is 0.01 CNY (0 USD) left for this transaction. |
| LBS_GEOGRAPHIC_INFORMATION_INVALID | The actual country of the address defined by store_address does not match the country defined by store_country, or the address defined by store_address cannot be found on Google Map.  |
| MCC_TYPE_ILLEGAL                   | MCC (parameter secondary_merchant_industry) is invalid.   |
| MOBILE_PAYMENT_SWITCH_OFF          | The Payment Code feature is disabled in the customer's Alipay app.  |
| MORE_THAN_ALLOW_REFUND_FOREX_FEE   | The principle of Alipay's split-refund is to return the original way. If a transaction involves refunding the split, and the refund amount exceeds the actual receipt amount of the account, this error code will be returned.  |
| NOT_CERTIFIED_USER                 | Buyer qualifications do not match.  |
| NOT_PRIVATE_ACCOUNT_USER           | The transaction can only be paid with the buyer's personal account, not the business account.   |
| NOT_SUPPORT_PAYMENT_INST           | The Alipay Wallet version is not supported.   |
| PARAM_ILLEGAL                      | The parameter is illegal. Modify the parameter according to the API document.   |
| PAYER_ENABLE_STATUS_FORBID         | The buyer's Alipay account is deactivated and the real-name verification information cannot be sent to the customs.   |
| PAYMENT_FAIL                       | The transaction failed.   |
| PAYMENT_REQUEST_HAS_RISK           | The merchant status is abnormal, or the buyer's payment has risks, such as offsite payment or large payment.  |
| PRODUCT_AMOUNT_LIMIT_ERROR         | The transaction amount exceeded the limit.  |
| PULL_MOBILE_CASHIER_FAIL           | Failed to call the mobile cashier.  |

|                                 |   |
|---------------------------------|---|
| PURCHASE_TRADE_NOT_EXIST        | The order for the refund request does not exist (status is TRADE_NOT_EXIST) or has not been paid (status is WAIT_BUYER_PAY).                      |
| QRCODE_HAS_BEEN_EXIST           | The QR code already exists. Try again or Contact Alipay Technical Support if needed.  |
| REASON_TRADE_REFUND_FEE_ERR     | Invalid refund amount.  |
| REFUND_CHARGE_ERROR             | The refund is initiated before Alipay receive the payment from the user.  |
| REFUND_REQUEST_HAS_RISK         | The merchant status is abnormal and transactions are not refundable.  |
| REFUNDMENT_VALID_DATE_EXCEED    | The transaction exceeded the valid refund period.   |
| REGISTRATION_NO_MANDATORY       | registration_no is not passed in the request parameters.  |
| REPEATED_REFUNDMENT_REQUEST     | Duplicated refund request. Please do not refund repeatedly.   |
| REQUEST_AMOUNT_EXCEED           | The value of refund_amount or the total refund amount is more than the transaction amount.  |
| RESTRICTED_MERCHANT_INDUSTRY    | The amount exceeded the merchant industry single order amount limit.  |
| RETURN_AMOUNT_ERROR             | The refund amount in CNY, which is the value of return_rmb_amount, is less than 0.01 when converted to the settlement amount in foreign currency. |
| RETURN_AMOUNT_EXCEED            | The total refund amount exceeds the amount actually paid for the transaction.   |
| SECONDARY_MERCHANT_ID_BLANK     | No secondary merchant information was passed in the payment request.  |
| SECONDARY_MERCHANT_ID_INVALID   | The parameter secondary_merchant_id passed in the request is incorrect.   |
| SECONDARY_MERCHANT_STATUS_ERROR | The secondary merchant status is abnormal.  |
| SELLER_BALANCE_NOT_ENOUGH       | When refunding, the seller's account balance was insufficient and the refund failed.  |
| SELLER_NOT_EXIST                | The parameter seller_id in the request is not consistent with   |

|                       |  |
|-----------------------|--|
|                       | partner. Or seller_email does not match partner.   |
| SOUNDWAVE_PARSER_FAIL | The payment code (buyer_identity_code) is invalid.   |
| SYSTEM_ERROR          | Alipay system failed to process the request because of temporary internal glitch.  |
| TARGET_SYSTEM_ERROR   | For interface alipay.acquire.overseas.spot.pay, parameter extend_info is not in JSON format.   |
|                       | Make sure the format of extend_info is JSON.   |
| TOTAL_FEE_EXCEED      | The transaction amount exceeds the limit.  |
| TRADE_BUYER_NOT_MATCH | The buyer does not match.  |
| TRADE_CANCEL_TIME_OUT | The cancellation request is beyond the opening hours.  |
| TRADE_HAS_CLOSE       | Payment or refund is not allowed for orders with a status of TRADE_CLOSED.   |
| TRADE_HAS_FINISHED    | The transaction is successfully paid and the refund period has expired. No payment or cancel request is allowed.   |
| TRADE_HAS_SUCCESS     | Already successful transactions are not allowed to initiate a payment again.   |
| TRADE_NOT_EXIST       | The transaction does not exist.  |
| TRADE_SETTLE_ERROR    | The error might be caused by the following reasons:  |
|                       | Errors exist in split_fund_info. For example, the value of transOut is incorrect.  |
|                       | Multiple partial refunds are requested within a same second.   |
|                       | Network jitter occurs during the settlement.   |
| TRADE_STATUS_ERROR    | The corresponding transaction status is not allowed for current operation.   |
| UNKNOWN               | In the in-store payment, if the amount is small and confidential, payment is returned to SUCCESS. If the amount is large, the confirmation payment page will be invoked, and the return code is UNKNOWN. |

|  |  |
|--|--|
| UNKNOWN_ERROR  | The service is temporarily unavailable.  |
|  | Try again later.   |
| The amount or currency of the foreign exchange purchased is wrong            | The transaction amount in transaction currency is rounded to 0 in the settlement currency. |
| The system is a little busy, try again later, or you can pay on the computer | The customer cannot use Alipay Global Payment Products.                                    |

## 8.4 UnionPay error code

Please check below link for details.

<https://developer.unionpayintl.com/cjweb/support/file/online/0/2/50>

| Response code | Response Message                                       | Error Detail   |
|---------------|--|--|
| 00            | Transaction approved / acknowledged                    |  |
| 01            | Transaction rejected, Refer to card Issuers for reason | The transaction is rejected by the Issuer, contact the Issuer for detailed reason  |
| 03            | Invalid merchant                                       | 1.Exceptional MCC;   |
|               |  | 2.MCC not supported for the transaction with this card;  |
|               |  | 3.This merchant is in the blacklist.   |
| 04            | Card should be captured                                | The Issuer believes that the card should be captured.  |
| 05            | Cardholder verification failure                        | 1. The transaction information of the e-commerce transaction is sent and received beyond the time limit.                               |
|               |  | 2. The cardholder identity verification fails (for transactions like commission relationship establishment or e-commerce transaction). |
|               |  | 3. Certificate information (e.g. type, number and so on) does not comply with the requirement.   |

|    |                             |  |
|----|-----------------------------|--|
|    |                             | <p>4. The cardholder authentication is invalid in case of expiry.</p> <p>5. The cardholder birth date verification fails.</p> <p>6. For cross-border remittance, the receiver information provided by the sending institution does not match what is registered at the receiving bank.</p> <p>7. The CVN2 verification fails.</p> <p>8. The cardholder authentication is invalid in case of mismatching.</p>   |
| 12 | Invalid related transaction | <p>1. The original transaction has not been accepted; however, a transaction such as reversal and cancellation, related to the original transaction, is received.</p> <p>2. The transaction that should occur every other day does not occur every other day.</p> <p>3. The cancellation or reversal is conducted for original transaction every other day.</p> <p>4. The transaction has not been conducted, but the information of related transaction is received (for example, the pre-authorization transaction has not been accepted, but the pre-authorization completion or pre-authorization cancellation transaction is received).</p> |
| 13 | Invalid transaction amount  | <p>Example:</p> <p>1.0 or other unallowable values;</p> <p>2.The transferred payment amount limit or the cash payment amount limit is exceeded;</p> <p>3.The transaction exceeds the upper limit required;</p> <p>4.The tip amount is beyond the limit</p> <p>5.Currency not supported;</p> <p>6.In cross-border remittance, the receiver or the sender does not have sufficient yearly foreign exchange quota to send or receive.</p>   |
| 14 | Invalid card number         | <p>1. The Issuer has no such primary account number.</p>   |

|    |  |   |
|----|--|---|
|    |  | <p>2. Though the original transaction is located, the primary account number of the related transaction does not match that of the original transaction.</p> <p>3. The check digit of the card number is incorrect.</p> <p>4. The account is invalid or closed.</p> <p>5. The primary account number of the response transaction does not match that of the request transaction.</p>  |
| 15 | No such Issuer                           | There is no Issuer matching the primary account number requested in the transaction.  |
| 21 | Card not initialized                     | <p>1. This card has not been activated or opened;</p> <p>2. The initial password of this card has not been changed;</p> <p>3. It is restricted to process transactions by the initial password;</p> <p>4. The card is frozen because of not being used for a long period of time or the card is in the "dormant" status.</p>  |
| 22 | Transaction procedure error              | <p>1. A cancellation request follows a completed reversal.</p> <p>2. A related transaction (such as reversal or cancellation) follows a cancelled transaction.</p> <p>3. A pre-authorization completion transaction follows a completed pre-authorization cancellation transaction.</p> <p>4. A pre-authorization completion transaction follows a completed pre-authorization reversal transaction.</p> <p>5. A repeated pre-authorization completion request follows a completed pre-authorization completion transaction.</p> <p>6. The pre-authorization transaction (including pre-authorization completion and pre-authorization cancellation) is initiated later than the last day of the period allowed for pre-authorization.</p> <p>7. The transaction is initiated later than the normal payment time.</p> |
| 25 | Unable to match the original transaction | The original transaction cannot be matched  |

|    |                           |  |
|----|---------------------------|--|
| 30 | Format error              | 1. XXXX missing/XXX format missing   |
|    |                           | 2. Invalid transaction channel value   |
|    |                           | 3. Error in field reading  |
|    |                           | 4. Error in sub-field reading  |
|    |                           | 5. Failed field examination  |
|    |                           | 6. Invalid character   |
|    |                           | 7. Bitmap format error   |
|    |                           | 8. Magnetic track information error  |
|    |                           | 9. Transaction amount missing.   |
| 34 | Suspected fraud           | 1. ARQC authentication error   |
|    |                           | 2. The number of CVN attempts is beyond the limit;   |
|    |                           | 3. Counterfeit card.   |
| 40 | Transaction not supported | 1. The Issuer has not enabled such transaction.  |
|    |                           | 2. The transaction type can be determined according to the message of the Member; however, such transaction has not been enabled.  |
|    |                           | 3. The Member can determine the transaction type according to the received message; however, such transaction is not included in the authority list or special authority list of the receiver. |
|    |                           | 4. The transaction type can be determined according to the message of the Member; however, the message version of the receiver does not support it.  |
|    |                           | 5. For one IC card transaction, the receiver is in the “Early” state and the receiver does not require ARQC authentication.  |
| 41 | Lost card                 | The lost card is captured.   |
| 43 | Stolen card               | The Issuer confirms that this is a stolen card. Please capture it.   |
| 45 | Fallback transaction      | 1. The Issuer does not support fallback transactions of hybrid-card.   |

|    |  |   |
|----|--|---|
|    |  | 2.The Issuer does not support fallback transactions of hybrid-card accepted in this region.   |
| 51 | Insufficient balance                             | The available balance in the account and the credit limit are insufficient, and the amount is beyond the limit for cash withdrawal. |
| 54 | Expired card                                     | 1.It is an expired card. The expiration date is incorrect.  |
|    |  | 2.The expiration date is not present, which should be included in the transaction message.  |
| 57 | Restricted card                                  | 1. The type of card cannot be used for such transaction.  |
|    |  | 2. It is out of the service scope.  |
|    |  | 3. That type of card cannot be accepted.  |
|    |  | 4. Deposit cannot be conducted with corporate card.   |
|    |  | 5. This account does not include this kind of currency.   |
|    |  | 6. This card is subject to suspected illegal cash withdrawal  |
|    |  | 7. The card number or certificate number is in the blacklist.   |
| 58 | Restricted terminal                              | 1. The Issuer prohibits the related transaction(s) from being conducted at such kind of terminals (specific to some card BINs).     |
|    |  | 2. The terminal number in the related transaction does not match that in the original transaction.                                  |
| 59 | Suspected fraud                                  | CVN authentication fails.   |
| 61 | Beyond the transaction amount limit              | 1. The single transaction amount is beyond the limit  |
|    |  | 2. The daily transaction amount is beyond the limit.  |
|    |  | 3. The transaction amount is beyond the limit defined by the Cardholder.  |
| 62 | Restricted card acceptance                       | Card cannot be accepted in the area;  |
| 64 | Unable to match with original transaction amount | 1.The transaction amount in the request message does not match that in the response message;  |
|    |  | 2. The transaction amount in the related transaction message does not   |

|    |                                       |  |
|----|---------------------------------------|--|
|    |                                       | match that in the original transaction message   |
| 65 | Beyond the transaction velocity limit | 1. Exceed the maximum number of transactions   |
| 68 | Response time-out                     | The Issuer cannot return the response within the time limit.   |
| 90 | Cutoff in progress                    | The daily switch is being conducted.   |
| 91 | Issuer system malfunction             | 1. The operation of the Issuer (or the transfer-in side/transfer-outside) is abnormal.   |
|    |                                       | 2. The operation of the Issuer (or the transfer-in side/transfer-outside) is abnormal; however, no stand-in authorization agreement is signed with UNIONPAY. |
|    |                                       | 3. The Issuer (or the transfer-in side/transfer-outside) signs out but does not sign in.   |
|    |                                       | 4. The operation status of the Issuer (or transfer-in side/transfer-outside) is invalid.   |
|    |                                       | 5. The Issuer (or the transfer-in side/transfer-outside) is closed by UNIONPAY.  |
| 92 | Routing failure                       | 1. No available connectivity line.   |
|    |                                       | 2. Error in IP address format and/or port number   |
|    |                                       | 3. State Administration of Foreign Exchange system is closed, or failed in inquiry therein.  |
| 94 | Duplicated transaction                |  |
| 96 | UnionPay system malfunction           | 1. The transaction is rejected due to the malfunction of UNIONPAY.   |
|    |                                       | 2. UNIONPAY internal processing failed   |
|    |                                       | 3. UNIONPAY is under maintenance and rejects all requests.   |
| 97 | Terminal not found                    | The ATM/POS terminal number has not been registered.   |
| 98 | Response timeout                      | The Issuer response is time out  |
| A2 | Successful transaction with defect    | UNIONPAY has forwarded the original requests, but not received the response from the Issuer.   |

|    |                                     |  |
|----|-------------------------------------|--|
| A3 | Transfer-in account not found       | The bankcard account is not found in the fund transfer-in bank.  |
| A4 | Successful transaction with defect  | The requests have not been received, the acceptance of the confirmed related transaction is a successful transaction with defect.    |
| A5 | Successful transaction with defect  | The original transactions are rejected, the acceptance of the confirmed related transaction is a successful transaction with defect. |
| A6 | Successful transaction with defect  | UNIONPAY has forwarded the original requests, but the response from the Issuer has not been received.                                |
| A7 | Security processing failure         | 1. Failure of key generation   |
|    |                                     | 2. Failure of key activation   |
|    |                                     | 3. Failure of key reset  |
|    |                                     | 4. Failure of ARPC generation  |
| E0 | Unauthorized access                 |  |
| E1 | Invalid message                     | 1. The UPI-JWS is missing  |
|    |                                     | 2. The UPI-REQPATH is invalid  |
|    |                                     | 3. Another required element in JWS header is missing   |
|    |                                     | 4. Format error of the UPI-TIMESTAMP   |
|    |                                     | 5. The UPI-TIMESTAMP expires   |
|    |                                     | 6. Duplicate message   |
| E2 | Signature verification failure      |  |
| E3 | System busy, please try again later |  |
| PG | Unknown message result              | PENDING  |