



# Merchant API Documentation

UPOP Payment API

Document version: 1.5

API version: 1.5



# TABLE OF CONTENTS

<b>1、 INTRODUCTION .....</b>	<b>5</b>
1.1 Payment mode .....	5
1.2 Audience .....	5
<b>2、 DESCRIPTION .....</b>	<b>5</b>
2.1 Business realization process (UPOP wap payment) .....	5
2.2 Business realization process (UPOP app payment) .....	7
<b>3、 DATA FORMAT .....</b>	<b>9</b>
3.1 Submit data .....	9
3.2 XML Data Format .....	9
<b>4、 DIGITAL SIGNATURE .....</b>	<b>11</b>
4.1 Original string of signature .....	11
4.2 Methods of signature .....	12
4.2.1 MD5 signature .....	12

4.2.2	SHA256 signature .....	13
4.2.3	RSA signature .....	15
<b>5、</b>	<b>PAYMENT INTERFACE .....</b>	<b>18</b>
5.1	UPOP WAP order API .....	18
5.2	UPOP APP Order API .....	21
5.3	Order Query API .....	24
5.4	Refund API .....	27
5.5	Refund Query API .....	30
5.6	Mobile App SDK Download Address .....	33
5.7	Asynchronous notification API .....	34
5.8	Aggregator Mode .....	37
<b>6、</b>	<b>NOTES .....</b>	<b>38</b>
<b>7、</b>	<b>ERROR CODE .....</b>	<b>40</b>

Document Changes		
Time	Version	Description
2022-10-20	1.0	First draft
2022-10-20	1.1	Add 5.7 Aggregator Mode description
2022-10-25	1.2	Add Request url : <a href="https://gateway.wepayez.com/pay/gateway">https://gateway.wepayez.com/pay/gateway</a>
2022-10-26	1.2	Add sequence diagram for Web2app
2022-10-26	1.2	Update field description for “pay_url” and “tn”
2023-01-03	1.3	Update the fields fee_type, order_fee, local_fee_type, trade_type
2023-01-09	1.3	Update the field trade_type
2023-01-31	1.4	Updated business descriptions and the diagram, signature sample message of SHA256, SDK download address;. Added MD5 signature method; Updated the following fields: fields charset, sign_type, out_trade_no, device_info, attach, notify_url, total_fee, nonce_str, err_code, err_msg, pay_url, tn, out_trade_no, transaction_id, time_end, op_user_id, refund_id, out_trade_refund, out_refund_no_n, refund_id_n, refund_channel_n, refund_fee_n, refund_time_n, refund_status_n; Added field out_transaction_id; Deleted fields out_refund_no, coupon_fee, bank_type, bank_billno, cash_fee, cash_fee_type, rate;
2023-02-21	1.4	Update field description for “pay_url” and “tn”
<a href="#">2023-05-25</a>	<a href="#">1.5</a>	<a href="#">Remove result code description</a> <a href="#">‘ It will be returned only when result_code is 1’</a>

# 1、 Introduction

## 1.1 Payment mode

When consumers use external explorer or APP and choose to pay with UnionPay, the explorer and APP can directly call UnionPay wallet to complete the payment.

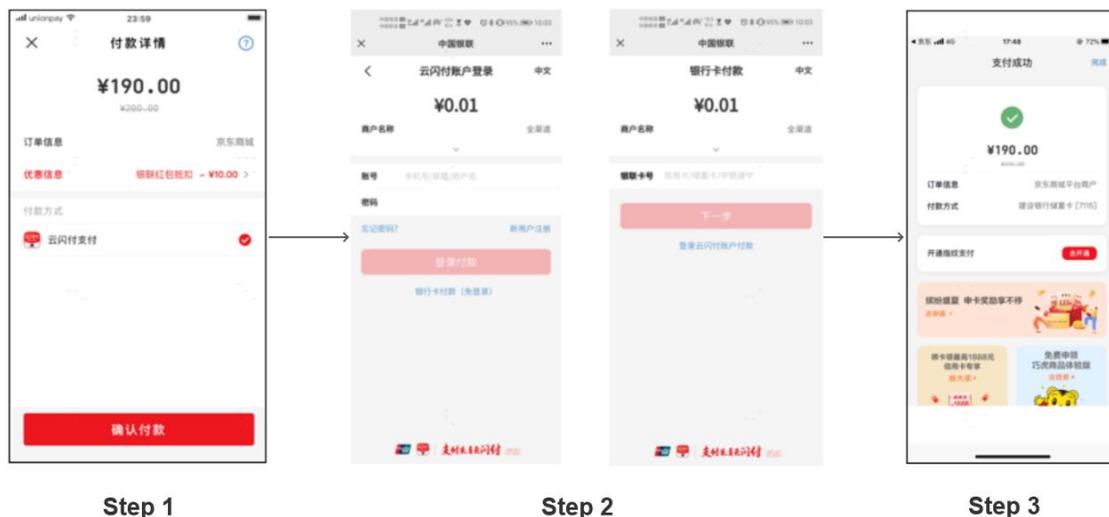
## 1.2 Audience

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

# 2、 Description

## 2.1 Business realization process (UPOP wap payment)

UI:



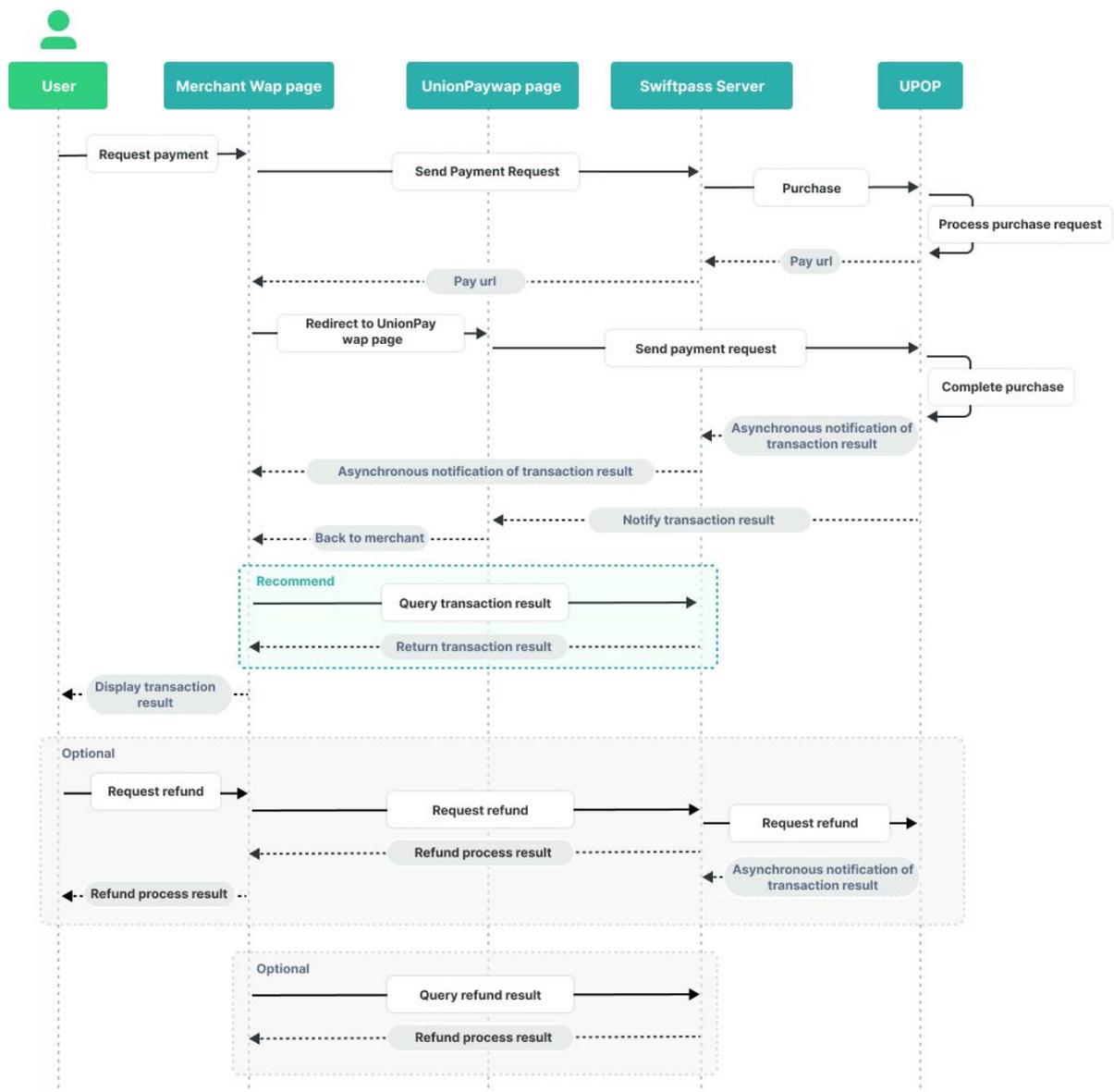
**Step 1:** Select UnionPay in merchant APP or WAP checkout

**Step 2:** Call UPOP WAP Order API and redirect to browser to open UnionPay Secure Pay page

**Step 3:** Input verification information (by logging in to UnionPay user account or use bank card) and complete payment

Note: Step 2 can also happen when UnionPay app is not installed when calling the UPOP APP Order API

**Diagram:**



## 2.2 Business realization process (UPOP app payment)

UI:

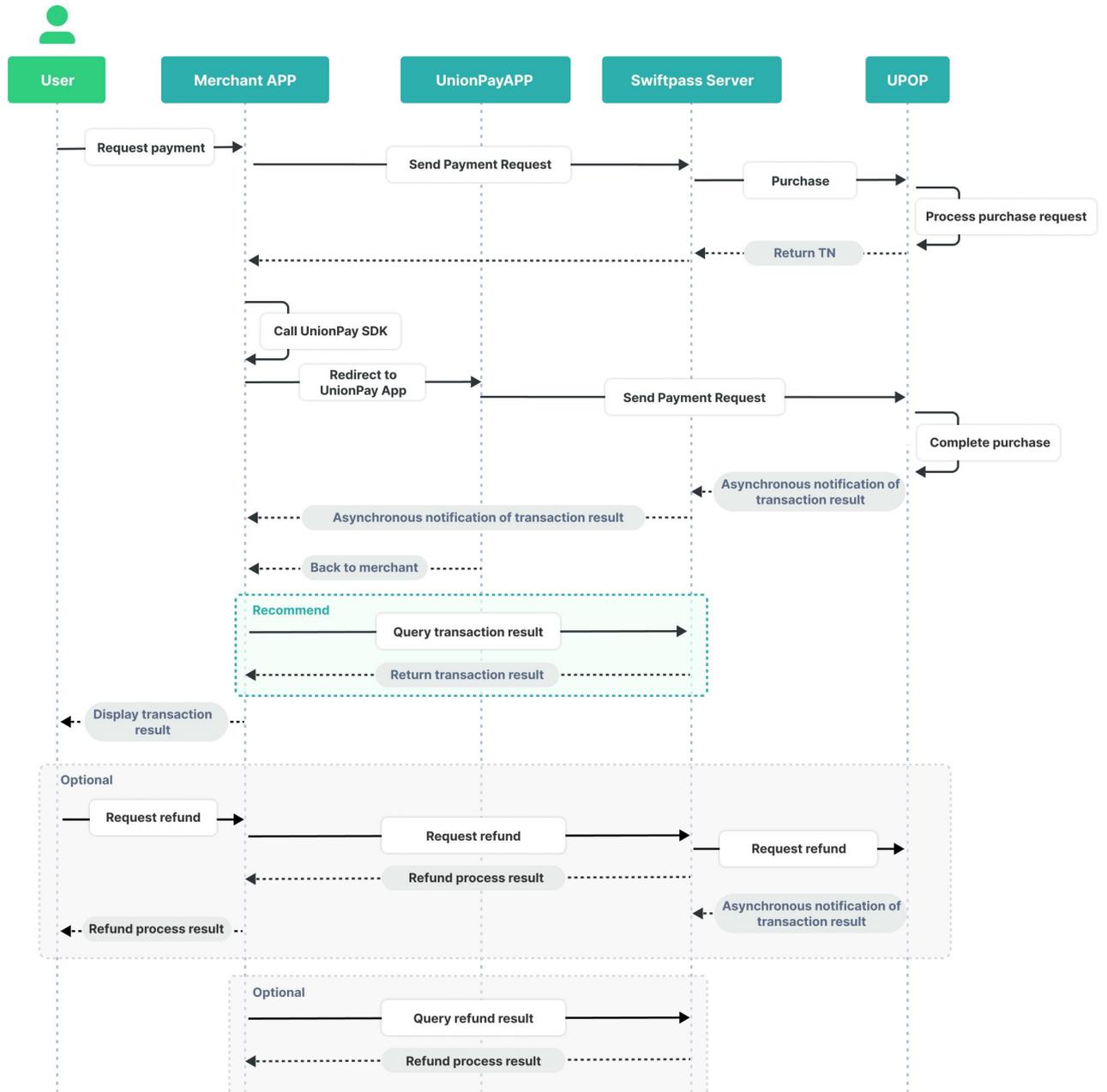


**Step 1:** Select UnionPay on merchant APP checkout page

**Step 2:** Call UPOP WAP Order API and redirect to UnionPay app

**Step 3:** Input verification information in UnionPay app and complete payment

Diagram:



## 3、 Data Format

### 3.1 Submit data

Uses HTTPS POST protocol. To ensure the recipient receives the correct data, all data must be signed (the following is an example for UPOP wap payment. Please fill in the corresponding type for the service field).

```
<xml>
<body><![CDATA[UPOP payment]]></body>
<device_info><![CDATA[000001]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100510000133]]></mch_id>
<nonce_str><![CDATA[1659089197]]></nonce_str>
<out_trade_no><![CDATA[1659089197]]></out_trade_no>
<service><![CDATA[pay.upi.upop.wap]]></service>
<sign><![CDATA[075A292FC14AC0B8EE406EFC2E1DFF0B]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<total_fee><![CDATA[10000]]></total_fee>
</xml>
```

### 3.2 XML Data Format

Uses Standard XML protocol. All parameters exist only in first-level node and needs to be included in CDATA. No nested nodes will be used.

Protocol error return:

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

Successful response:

```
<xml>
<status>0</status>
<message><![CDATA[OK]]></message>
<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>
<total_fee>1</total_fee>
<fee_type><![CDATA[CNY]]></fee_type>
<transaction_id><![CDATA[1008450740201407220000058756]]></transactio
n_id>
<out_trade_no><![CDATA[1406033828]]></out_trade_no>
<attach><![CDATA[att]]></attach>
<time_end><![CDATA[20140722160655]]></time_end>
</xml>
```

Business error return:

```
<xml>
<status>0</status>
<message><![CDATA[OK]]></message>
<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[QR code expired, refresh and
retry]]></err_code_des>
</xml>
```

## 4. Digital Signature

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

### 4.1 Original string of signature

The original string of signature will be assembled into character strings according to the following modes (applicable to both request and response):

- All parameter fields (except for sign field) 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...). Null value will neither transfer nor participate in formation of string of signature.
- In the original string of signature, both field name and field value will adopt original values and will not conduct URL Encode.
- Response or notification information returned by the platform might bring more parameters due to upgrade. This case should be allowed when the response signature is verified.

For example:

Calling an interface with following fields:

```
<xml>
<body><![CDATA[UPOP_payment]]></body>
<device_info><![CDATA[000001]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100510000133]]></mch_id>
<nonce_str><![CDATA[1659089197]]></nonce_str>
<out_trade_no><![CDATA[1659089197]]></out_trade_no>
<service><![CDATA[pay.upi.upop.wap]]></service>
<sign><![CDATA[075A292FC14AC0B8EE406EFC2E1DFF0B]]></sign>
```

```
<sign_type><![CDATA[MD5]]></sign_type>
<total_fee><![CDATA[10000]]></total_fee>
</xml>
```

The correct signature field sequence should be:

```
body=UPOP_payment&device_info=000001&mch_create_ip=127.0.0.1&mch_id=100510000133&nonce_str=1659089197&out_trade_no=1659089197&service=pay.upi.upop.wap&sign_type=MD5&total_fee=10000
```

## 4.2 Methods of signature

### 4.2.1 MD5 signature

MD5 signature calculation formula:

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

Suppose the following are incoming XML parameter:

```
<xml>
<body><![CDATA[TestPay]]></body>
<callback_url><![CDATA[http://js.testpay.com/testpay.asp]]></callback_url>
<device_info><![CDATA[100]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[127520000042]]></mch_id>
<nonce_str><![CDATA[2209196862]]></nonce_str>
<notify_url><![CDATA[https://int.payable.lk:8088/mposcbc/callback]]></notify_url>
<out_trade_no><![CDATA[127590000128]]></out_trade_no>
<service><![CDATA[pay.upi.upop.wap]]></service>
<sign><![CDATA[9D2C356E9356330EA49F660CB5B40722]]></sign>
<total_fee><![CDATA[250]]></total_fee>
</xml>
```

Suppose merchant key is: 9f72151b6592fab3e0c63a1ab3c0877b

i: string1 after URL key sequencing the dictionary order in process a:  
 body=TestPay&callback\_url=http://js.testpay.com/testpay.asp  
 &device\_info=100&mch\_create\_ip=127.0.0.1&mch\_id=127520000042&nonce  
 \_str=2209196862&notify\_url=https://int.payable.lk:8088/mposcbc/callback&ou  
 t\_trade\_no=127590000128&service=pay.upi.upop.wap&total\_fee=250

ii: sign after process b:

sign  
 =md5(string1&key=9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase  
 =md5(body=TestPay&callback\_url=http://js.testpay.com/testpay.asp  
 &device\_info=100&mch\_create\_ip=127.0.0.1&mch\_id=127520000042&nonce  
 \_str=2209196862&notify\_url=https://int.payable.lk:8088/mposcbc/callback&ou  
 t\_trade\_no=127590000128&service=pay.upi.upop.wap&total\_fee=250  
 &key=  
 9d101c97133837e13dde2d32a5054abb).toUpperCase()  
 ="9D2C356E9356330EA49F660CB5B40722"

## 4.2.2 SHA256 signature

SHA256 signature calculation formula:

sign =SHA256("The signature field sequence strings" &key= "signature key").  
 toUpperCase

Example:

There are XML afferent parameters:

```
<xml>
<body><![CDATA[TestPay]]></body>
<callback_url><![CDATA[http://js.testpay.com/testpay.asp]]></callback_url>
<device_info><![CDATA[100]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[127520000042]]></mch_id>
<nonce_str><![CDATA[2209196862]]></nonce_str>
```

```
<notify_url><![CDATA[https://int.payable.lk:8088/mposcbc/callback]]></notify_
url>
<out_trade_no><![CDATA[127590000128]]></out_trade_no>
<service><![CDATA[pay.upi.upop.wap]]></service>
<sign><![CDATA[2D73F49E3F4681BA4AFAD9E73D88D2DAD448E1A077B5
51D137555401330401F3]]></sign>
<total_fee><![CDATA[250]]></total_fee>
</xml>
```

**Suppose the merchant signature key is:**

**18e0a2ad5d5571af14b855fcf33091f4**

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

```
body=TestPay&callback_url=http://js.testpay.com/testpay.asp
&device_info=100&mch_create_ip=127.0.0.1&mch_id=127520000042&nonce
_str=2209196862&notify_url=https://int.payable.lk:8088/mposcbc/callback&ou
t_trade_no=127590000128&service=pay.upi.upop.wap&total_fee=250
```

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

```
sign
=SHA256(string1&key=18e0a2ad5d5571af14b855fcf33091f4).toUpperCase
=SHA256(body=TestPay&callback_url=http://js.testpay.com/testpay.asp
&device_info=100&mch_create_ip=127.0.0.1&mch_id=127520000042&nonce
_str=2209196862&notify_url=https://int.payable.lk:8088/mposcbc/callback&ou
t_trade_no=127590000128&service=pay.upi.upop.wap&total_fee=250
&key=
18e0a2ad5d5571af14b855fcf33091f4).toUpperCase()
="2D73F49E3F4681BA4AFAD9E73D88D2DAD448E1A077B551D137555401
330401F3"
```

### 4.2.3 RSA signature

RSA is the most widely used “asymmetric encryption algorithm”. It adds the RSA private key of the merchant communication to the end of the signed original string, and the result string is the RSA operation result.

Note: When converting a string to a byte stream, the specified set of coded characters 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.

RSA signature calculation formula:

- 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 signature algorithm will be signed using merchant’s private key and conduct Base64 encoding. The created signature will fill in the “sign” parameter and pasted to the request parameter.

#### Example:

There are XML afferent parameters:

```
<xml><body><![CDATA[UPO_payment]]></body>
<device_info><![CDATA[000001]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[100510000133]]></mch_id>
<nonce_str><![CDATA[1659090140]]></nonce_str>
<out_trade_no><![CDATA[1659090140]]></out_trade_no>
<service><![CDATA[pay.upi.upop.wap]]></service>
<sign><![CDATA[Yq/RKPDs7/p7bMJUC4NG9ybdNoqSGTf76e1dg8J4SS5OP
8aDoMQRMjcv50V+rItYW5jwJ2G3uo6SIC8+6TmXjP3EFdluNxvuG7UBVWm
EtGmrSXFvWNMFx6CtRfbf1ZJmagN5rxOKWPkjin8DGlyNMI7sFSM4XQOaZ
7vUEyM5ejKh1rz/CUd5INTzqG4egccHfX94oBpYNg44DZcJagYuTaJ3aCUIJw
3HAKvAASNgNIJ4py3nGPOq5nEHonrgaMJ0lvyNJ3VFMWh5BnFFiA4U6046
```

```
cmQP/cbjZoMXKkJB49jlvk33SCXs8VzfFjgGi6WHCP7qzKp5BOvqCe8/oH+E  
o7g==]]></sign>  
<sign_type><![CDATA[RSA_1_256]]></sign_type>  
<total_fee><![CDATA[10000]]></total_fee>  
</xml>
```

**Suppose the Merchant RSA Private key is:**

MIIEvglBADANBkgqhkIG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCu2Zs+L  
HVmiURxnnMT8B9kdquVRq5naBOUjIWqU/E9o8U2H1GmCG9J/ejmYSVhh8  
M27YQMvvyff0PmQFNlmtouo5nsThqKDSipqaoFvPY6y6XgQb13NEcF1os  
1+fErlAJmdn/SGDioNNsgKlcxd0gcCnsgw5Dv49j23+gjtWhHc31WVUrNNxFD  
AjZuqA/4bk4EwKkgoD5ojQLWTAHj/xFotl2GNXTXYRzkOYka/B+cYu1e4XO3  
KhW4cZkqcqsz5K3SibZnIWdsKoQW6+YSD4nwu0QX0D85dhE71acYaHY3u  
6eo1l614F3EAbpxrvkF1/rKQ33lSoAHKoN4hVN9RAgMBAAECggEABfBaN4h  
nDG6JJprNKdY0GsN1HILkFZsRXzBOsfRb804GusnhihQrQ4YeTQTrsh5cYR  
c8LvqltqTpp48e8pZgAWfRMNfvBx4Ke6PRcVAmRC3ozTiStfMkHylj6/YBZ0r  
MaScSonsg31G7HKpaFk6aLinhlqlo6KAvDKM9+IQheW3gTa77Op7DyvlmUm  
enATgYuRDNAI9AtzM+kcUbmvnQO3AScD2zQ+AtU0Qp0GBAv6T3zrfHCzcL  
1aBMsnTCsrPj3DZ/2WEm2Dfdy9OaIX6tkaypQeVzRn/QsINJVYRL5doSh4yM  
UKyJnR+vCcVWnhErR6oZ65ocYwsLAhmRNasAQKBgQDnH2jEoynWz65Bj/f  
ojUtfpr2G/N2O1Q8xyXZxqoMS0CEuip0AqIHTEpgsNHfWIVutDE004QVKimL  
m0Z+GbodCiCMO6ePDItXw88Zc4seZbb84a1UPeYLSH3efMgxy6gm4ZemAH  
4XOfMfsk7gN27dC54l+8TYDb9sCfQ+v1vDqIQKBgQDBq5pEuuU0Gkk1EMN  
uyavTi83isYkfSGTQWNI9NGONLUNs5pHOGBrJUOP7wMbJPU0JDnASy32l  
4cKDJrX3CFJaubWQIFrrde48MUIvQtRFCI3+jtHhnci4go0to4pHLQm7y2znnb  
KudE3r88Oz4B0sZg2Fsqi6UFR+0cGtKrwwMQKBgQCkZHPpwcCluUzR92w3Li  
EZpQkT9dJFoQhQumKUENtwSzf+2bs416KYFpbEgoFmfo2ttZ1l6nEQBwS39  
wxH22A8E4cUOYq2ppWT8GD8epXMbQJhR/7VpfFYsD0ur7OpHQzS6zi93q  
4XJCwz+xlkUU61uOjUsQ9hPHOD/p5kzTbI4QKBgGyoO3AuL2QRD1ufYRZq  
PWX9vPBfzN/pPO6LtsCyANx3NaoGnm0zpuul35MicCT9qJfZKtXQp1kHFZuB  
f3vuhk3yfUR1h6EtVBEfUeUDaJann16vgF+Fp9SIlllu/H8+cYL9rjoCBTKmeE0

```
rWrg5V1AkHufZfzKEpN+2ZvphH0xAoGBAJc7Q9PASirQcrgGz7sF3JbVJQva
eeulUUXSuYiLpGSQ5fKElpTpJQg+m40muA58TLgTe3KDqw6de25osIYEBKo
4XuC5c2LkC9Xhgv9nJHEXJqseER82f3JROP6M2IY7JI6OPp7tmNj4pcH5irZK
QSRfW/0EqAC70xqMRANioLP6
```

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

```
body=UPOP_payment &device_info=000001&mch_create_ip=127.0.0.1&mch_id=100510000133&nonce_str=1659090140&out_trade_no=1659090140&service=pay.upi.upop.wap&sign_type=RSA_1_256&total_fee=10000
```

**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=UPOP_payment&device_info=000001&mch_create_ip=127.0.0.1&mch_id=100510000133&nonce_str=1659090140&out_trade_no=1659090140&service=pay.upi.upop.wap&sign_type=RSA_1_256&total_fee=10000)
```

sign=

```
Yq/RKPDs7/p7bMJUC4NG9ybdNoqSGTf76e1dg8J4SS5OP8aDoMQRmjcv50
V+rltYW5jwJ2G3uo6SIC8+6TmXjP3EFdluNxvuG7UBVWmEtGmrSXFvWNM
Fx6CtRfbf1ZJmagN5rxOKWPkjin8DGlyNMI7sFSM4XQOaZ7vUEyM5ejKh1rz/
CUd5INTzqG4egccHfX94oBpYNg44DZcJagYuTaJ3aCUIJw3HAKvAASNgNIJ
4py3nGPOq5nEHonrgaMJ0lvyNJ3VFMWh5BnFFiA4U6046cmQP/cbjZoMXKk
JB49jlvk33SCXs8VzfFjgGi6WHCP7qzKp5BOvqCe8/oH+Eo7g==
```

## 5、 Payment Interface

### 5.1 UPOP WAP order API

Merchant call an UPOP wap request, an pre-payment order will be generated in the swiftpass and UPOP background. Merchant will receive the field pay\_url in the response, it will call up the UPOP payment page to open the UnionPay App or H5 browser's and complete payment.

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

**Request type:** POST

**Request format:** XML

#### Fields:

Field Name	Required	Type	Description
<b>Normal Parameters</b>			
service	Yes	String(32)	Interface type: pay.upi.upop.wap
version	No	String(8)	Version number. Default value: 2.0
charset	No	String(8)	Encoding method. Default value: UTF-8
sign_type	No	String(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
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)	Terminal serial number customized by merchant (only recorded in database)
body	Yes	String(128)	Description of merchants' goods. Can upload 128 English, 42 Chinese

attach	No	String(128)	Additional merchant info, JSON format. 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	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1)
mch_create_ip	Yes	String(64)	Machine IP generated by order
notify_url	No	String(255)	URL to receive asynchronous notification, absolute path is needed, format within 255 characters, such as: <a href="http://js.testpay.com/testpay.asp">http://js.testpay.com/testpay.asp</a> ; should make sure the platform can access the address from internet . If it's absent, merchant will not receive notification defined in Section 5.7
callback_url	Yes	String(255)	Redirection URL after payment completes. Absolute path needed, format within 255 characters, such as: <a href="http://js.testpay.com/testpay.asp">http://js.testpay.com/testpay.asp</a> ; should make sure the platform can access the address from internet
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
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.

**Return field:**

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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256

status	Yes	String(16)	"0": success. Other values: 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 is invalid.
The following fields will be returned when status is "0":			
result_code	Yes	String(16)	"0": successful; "1": failed
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform.
device_info	No	String(32)	Terminal serial number customized by merchant (only record in database)
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
err_code	No	String(32)	Please refer to the error code list.
err_msg	No	String (128)	Description of result information.
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 both status and result_code are "0".			
pay_url	Yes	String(256)	pay_url is the middle page of the payment cashier. You can access the url to open the UnionPay App and complete the payment.  Please note that if the device does not install the UnionPay APP, the the url will jump to the browser's H5 payment page to complete the payment.  The validity period for order is depend on UnionPay, and it's 30 minutes.

## 5.2 UPOP APP Order API

Merchant call an UPOP app request, an pre-payment order will be generated in the swiftpass and UPOP background. Merchant will receive an TN number in the response, merchant APP using this TN number to call UnionPay payment SDK to open the UnionPay App or H5 browser's and complete payment.

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

**Request type:** POST

**Request format:** XML

Fields:

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type: pay.upi.upop.app
version	No	String(8)	Version number. Default value: 2.0
charset	No	String(8)	Encoding method. Default value: UTF-8
sign_type	No	String(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
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)	Terminal serial number customized by merchant (only record in database)
body	Yes	String(128)	Description of merchants' goods. Can upload 128 English, 42 Chinese
attach	No	String(128)	Additional merchant info, JSON format. 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	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1)
mch_create_ip	Yes	String(64)	IP Machine IP generated by order
notify_url	No	String(255)	URL to receive asynchronous notification, absolute path needed, format within 255 characters, such as: <a href="http://js.testpay.com/testpay.asp">http://js.testpay.com/testpay.asp</a> ; should make sure the platform can access the address from internet. If it's absent, merchant will not receive notification which define in Section 5.7
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated sign value unpredictable.
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.

#### Return fields:

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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256
status	Yes	String(16)	"0": success. Other values: 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 is invalid.
The following fields will be returned when status is "0":			
result_code	Yes	String(16)	"0": successful; "1": failed

mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
device_info	No	String(32)	Terminal serial number customized by merchant (only record in database)
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated sign value unpredictable.
err_code	No	String(32)	Please refer to the error code list.
err_msg	No	String (128)	Description of result information.
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 both status and result_code are "0".			
tn	Yes	String(256)	<p>After calling the pay.upi.upop.app API, the SwiftPass will return a TN number generated by UnionPay. If the merchant APP has integrated the UnionPay payment SDK, add this TN number to the parameters of the calling. After the UnionPay payment SDK calls the UnionPay APP, it can identify the transaction request initiated by the merchant according to this TN number and continue to complete the payment in the UnionPay APP. Please note that if the device does not install the UnionPay APP, the SDK will jump to the browser's H5 payment page to complete the payment.</p> <p>The validity period for order is depend on UnionPay, and it's 1 hours.</p>

## 5.3 Order Query API

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

**Request type:** POST

**Request format:** XML

Request parameters are as follow:

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type: 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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
out_trade_no	No	String(32)	Order number of the payment transaction (excluding refund order). 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)	Platform transaction ID of the payment transaction (excluding refund order), it is returned in notification. At least one of the out_trade_no and transaction_id should be existed in query request; when both exist, transaction_id shall prevail
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
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.
--------------	----	------------	--

### Return parameters:

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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256
status	Yes	String(16)	"0": success. Other values: 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 is invalid.

The following fields will be returned when both status and result\_code are "0".

result_code	Yes	String(16)	"0": successful; "1": failed
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
device_info	No	String(32)	Terminal serial number customized by merchant (only recorded in database)
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
err_code	No	String(32)	Please refer to the error code list.
err_msg	No	String (128)	Result information description.

trade_state	Yes	String(32)	SUCCESS—Payment successful REFUND—Refund received NOTPAY—Not paid CLOSED—Closed REVOKED—Revoked
sign	Yes	String(344)	Please refer to “Chapter 4 Digital Signature”
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.
The following fields will return when trade_state is SUCCESS and REFUND			
trade_type	Yes	String (32)	Interface type: pay.upi.upop.wap pay.upi.upop.app
trade_state_desc	No	String(32)	Transaction status description
pay_result	Yes	Int	“0”: success Other: need to call query interface again.
transaction_id	Yes	String(32)	Platform transaction number
out_transaction_id	Yes	String(32)	Transaction ID in UnionPay
out_trade_no	Yes	String(32)	Unique order number in the merchant system with 5-32 digits. Can include only letters, numbers, and underscore. Upper and lower case sensitive.
total_fee	Yes	Int	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1)
fee_type	No	String(8)	Currency, 3-digit code under ISO 4217 standard.
attach	No	String(127)	Additional merchant info, JSON format. The original value will be returned after successful payment.

time_end	No	String(14)	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. Returned when order status is successful or refunded
local_fee_type	No	String(8)	Currency labeled merchant, e.g HKD, which is the pricing currency of merchandises
local_total_fee	No	Int	Amount labeled by merchant in pricing currency
order_fee	No	Int	Order amount (in pricing currency)

## 5.4 Refund API

The merchant initiates a refund for an order that has been successfully paid, and the results will be returned synchronously in the same session.

### 1) Refund routes

Currently, only refunds by the same route are supported.

Note: For refunds to bank cards, due to the different processing time of each bank, the refund will be received within 7 weekdays after the initiation of the refund request.

Partial refunds for the same order require the same order number and a different out\_refund\_no. If a refund fails and is resubmitted, the original out\_refund\_no shall be adopted. The total refund amount cannot exceed the amount actually paid by the user.

### 2) Restrictions

In the platform, as long as the cumulative amount of refunds does not exceed the total amount paid for a transaction order, a transaction order can be refunded several times. The refund requisition number (this parameter is in the refund API) will determine a refund once, instead of the transaction order number. The refund requisition number is generated by the merchant, so the merchant must ensure the uniqueness of the refund application form. Merchants should pay special attention in the refund process: only when a refund actually fails can another refund be initiated.

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

**Request type:** POST

**Request format:** XML

Request parameters are as follow:

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type: 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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
out_trade_no	No	String(32)	Order number of the payment transaction (excluding refund order). At least one of the out_trade_no and transaction_id should exist in refund request; when both exist, transaction_id shall prevail.
transaction_id	No	String(32)	Platform transaction ID of the payment transaction (excluding refund order). At least one of the out_trade_no and transaction_id should exist in refund request; when both exist, transaction_id shall prevail
out_refund_no	Yes	String(32)	Merchant refund ID, within 32 digits and characters can be included, unique in the merchant system. Platform will treat multiple requests from the same ID as one request and refund will be made only once. If the refund fails, please request again with the same ID to avoid duplicate refund.
total_fee	Yes	Int	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1). Value shall be total amount of payment transaction.
refund_fee	Yes	Int	Total refund amount (in cents). Refund can be made partially.
op_user_id	Yes	String(32)	Operator user ID (if there is no user id, it is recommended to fill merchant ID)

nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
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.

**Return result:**

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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256
status	Yes	String(16)	"0": success. Other values: 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 is invalid.
The following fields will be returned when status returns "0"			
result_code	Yes	String(16)	"0": success Others: fail
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
device_info	No	String(32)	Terminal serial number customized by merchant (only record in database)
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
err_code	No	String(32)	Please refer to the error code list. .
err_msg	No	String (128)	Result information description.
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 both status and result_code return "0"			
transaction_id	Yes	String(32)	Platform transaction ID
out_trade_no	Yes	String(32)	Order number of payment transaction (excluding refund order).
out_refund_no	Yes	String(32)	Merchant refund number.
refund_id	Yes	String(32)	Refund transaction ID in platform, 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	Total refund amount in cents. Partial refund is available.
total_fee	No	Int	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1)
fee_type	No	String(8)	Order channel currency
order_fee	No	Int	Order amount
local_total_fee	No	Int	Amount labeled by merchant in pricing currency.
local_fee_type	No	String(8)	Pricing currency labeled by merchant, e.g HKD.
out_transaction_id	Yes	String(32)	Transaction ID in UnionPay
trade_type	Yes	String(32)	Interface type: pay.upi.upop.wap pay.upi.upop.app

## 5.5 Refund Query API

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

**Request type:** POST

**Request format:** XML

Request parameters are as follow:

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type: 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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform
out_trade_no	No	String(32)	Order number of the payment transaction (excluding refund order). At least one of the out_trade_no and transaction_id should be exist in refund query request; when both existed, transaction_id shall prevail.
transaction_id	No	String(32)	Platform transaction ID of the payment transaction (excluding refund order), it is returned in notification. At least one of the out_trade_no and transaction_id should be existed in refund query request; when both exist, transaction_id shall prevail
refund_id	No	String(32)	Platform refund ID. At least one of the refund_id and out_trade_refund should be existed in refund query request;
out_trade_refund	No	String(32)	Merchant refund number. At least one of the refund_id and out_trade_refund should be existed in refund query request;
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
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.

**Return result:**

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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256

status	Yes	String(16)	“0”: success. Other values: 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 is invalid.
The following fields will be returned when status returns “0”			
result_code	Yes	String(16)	“0”: successful; other values: failed
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform
device_info	No	String(32)	Terminal serial number customized by merchant (only record in database)
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated signature value is unpredictable.
err_code	No	String(32)	Please refer to the error code list.
err_msg	No	String (128)	Result information description.
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 both status and result_code return “0”			
transaction_id	Yes	String(32)	Platform transaction ID
out_trade_no	Yes	String(32)	Order number within merchant system
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)	Platform refund ID. “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 return 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	Total refund amount in cents. Refunds can be made partially. Value taken from 0 to (total refund count – 1). E.g. refund_fee_0, refund_fee_1, etc.

refund_time_n	Yes	String(14)	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. Value taken from 0 to (total refund count – 1). E.g. refund_time_0, refund_time_1, etc.
refund_status_n	Yes	String(10)	SUCCESS or PROCESSING. Value taken from 0 to (total refund count – 1). E.g. refund_status_0, refund_status_1, etc,
total_fee	No	Int	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1)
fee_type	No	String(8)	Order channel currency
order_fee	No	Int	Order amount
local_total_fee	No	Int	Amount labeled by merchant in pricing currency
local_fee_type	No	String(8)	Pricing currency labeled by merchant, e.g HKD.
out_transaction_id	Yes	String(32)	Transaction ID in UnionPay
trade_type	Yes	String(32)	pay.upi.upop.wap pay.upi.upop.app
refund_count	No	Int	Current query numbers, fixed as 1

## 5.6 Mobile App SDK Download Address

The SDK download address for UnionPay APP as below:

IOS: <https://clouddisk.unionpayintl.com/link/2iqCAqoOtJ2uwWyl/>

Password:CaH5Lf

Andriod: <https://clouddisk.unionpayintl.com/link/6i1rcylbQWLGXEw/>

Password:HHyHWP

## 5.7 Asynchronous notification API

The notification URL is the parameter `notify_url` submitted in the single interface. After the payment is completed, the platform will send the relevant payment and user information to the URL, and the merchant needs to receive the processing information. When interacting with the background notification, if the platform receives a response from the merchant that is not a pure string success or returns after more than 5 seconds, the platform considers the notification failed, and the platform will indirectly re-initiate the notification through a certain strategy (**notification frequency: 0/15/30/180/1800/1800/1800/1800/3600, unit: seconds**), so as to improve the success rate of the notification as much as possible, but it does not guarantee that the notification will ultimately succeed.

The same notification may be sent to the merchant system multiple times due to the re-sending of background notifications. **The merchant system must be able to correctly handle duplicate notifications.** It is recommended that when receiving a notification for processing, the status of the corresponding business data is first checked to determine whether the notification has been processed. If not, the notification shall be processed again. If processed, the result is returned to success directly. Before checking and processing the status of business data, concurrency control should be implemented by using data locks to avoid data confusion caused by function reentry.

**Special note:** After the merchant back end receives the notification parameter, it should check the order number `out_trade_no` and the order amount `total_fee` in the received notification parameter with the order and amount of its own business system, and update the order status of

**the database only after making sure they are consistent. There is no notification for refund transaction.**

Suggestion: after the order is created and the payment is completed, if payment successful notification request is not received within 5 minutes, it is recommended to initiate the order query API. Each query will last 70 seconds and be re-initiated every 5 seconds to determine the order status.

The asynchronous notification is made through the notify\_url in the request, and the post method is given to the merchant system (the notification parameter is an xml string)

Return format: XML

Swiftpass notification parameters are as follow:

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(12)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256
status	Yes	String(16)	"0": success. Other values: 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 is invalid.
The following fields will be returned when status returns "0"			
result_code	Yes	String(16)	"0": successful Others: failure
mch_id	Yes	String(32)	Specifies a unique ID assigned by platform (for common merchants)
device_info	No	String(32)	Terminal serial number customized by merchant (only record in database)
nonce_str	Yes	String(32)	Random character string no longer than 32 characters. It's used to guarantee the generated sign value unpredictable.
err_code	No	String(32)	Please refer to the error code list.

err_msg	No	String (128)	Result information description.
sign	Yes	String(32)	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 both status and result_code return "0"			
trade_type	Yes	String(32)	pay.upi.upop.wap pay.upi.upop.app
pay_result	Yes	Int	"0": successful Other: failed
transaction_id	Yes	String(32)	Platform transaction ID, it's unique ID in platform.
out_trade_no	Yes	String(32)	Order number of payment transaction.
out_transaction_id	Yes	String(32)	Transaction ID in UnionPay
total_fee	Yes	Int	Amount, determined by payment currency with the minimum unit as cent. E.g. If pay by HKD, 1000 stands for HK\$10.00; maximum limit per transaction for testing merchant is 100 (which is HK\$1)
fee_type	No	String(8)	Currency type, 3-digit code under ISO 4217 standard.
local_fee_type	No	String(8)	Amount labeled by merchant in pricing currency
local_total_fee	No	Int	Pricing currency labeled by merchant, e.g HKD.
order_fee	No	Int	Order amount (in pricing currency)
attach	No	String(127)	Additional merchant info, JSON format. The original value will be returned after successful payment.
time_end	No	String(14)	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. Returned when order status is payment successful or refunded.

### Asynchronous notification feedback

The platform server sends a notification, the post sends an XML data stream, and the merchant notify\_Url address receives the notification result. After the merchant performs business processing, the merchant needs to feed back the processing result in the form of a pure string, with the following contents:

Return results	Description
----------------	-------------

success	Process successful. No more notifications will be sent.
Fail or other fields	Process unsuccessful. The result is either received or unreceived by the platform. The system shall re-notify with the replenishment mechanism (see section 5).

## 5.8 Aggregator Mode

When using the aggregator mode, the merchants need to create the channel ID in SwiftPass Portal. And the Channel ID is necessary to apply in the below Field. Sales and Refund transactions from different merchants will deliver by one single Channel ID.

Field Name	Required	Type	Description
sign_agentno	Yes	String(32)	When the merchant is in the Aggregator mode, 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 the Aggregator Mode:

```

<xml>
<body><![CDATA[changyoyo]]></body>
<callback_url><![CDATA[http://js.testpay.com/testpay.asp]]></callback_url>
<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[2022092611300000000]]></out_trade_no>
<sign_agentno><![CDATA[1231231]]></sign_agentno>
<service><![CDATA[pay.upi.upop.wap]]></service>
<sign><![CDATA[79B0E084D6DB7AFD25224A5AD01774CDFBA1EB5F30521E079
2553E98E9D91BB2]]></sign>
<sign_type><![CDATA[SHA256]]></sign_type>
<total_fee><![CDATA[1]]></total_fee>

```

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

## 6、 Notes

- 1) All amount units are in cents, and the smallest unit is 1 cent. Decimals are not allowed.
- 2) notify\_url means that the platform server directly initiates a request to the merchant server from the background, and the merchant cannot check the cookie or session of the user when processing; Shipping processes such as merchant updating DB need to be completed after notify\_url to ensure that platform replenishment can be successfully completed when the order is dropped
- 3) notify\_url may be notify repeatedly. The merchant needs to reprocess to avoid multiple shipments.
- 4) For the notification received by notify\_url, if the merchant has successfully processed or checked the order, it is necessary to return the pure string success to indicate successful processing. The string success is not case-sensitive. If we do not receive the returned success, our server will continue to send you the notice and will not give you any further notice after three hours. Assuming that all orders have not returned to success, it will increase the notification load on our server. At worst, it may cause a delay in notifying the merchants normally; In addition, we will urge you to improve the situation. If no improvement is made, the research and development or operation and maintenance technology will take control measures over the payment interface.
- 5) If yes is required among the returned parameters, it will be returned; if no is required, it will not always be returned. The actual returned parameters may not be exactly the same as the document due to upgrade or configuration, and must be based on the actual received parameters.

6) Other matters to note

**(1) Case issue**

Please pay attention to the case of the required characters in the document, such as "after the sign-up operation, the characters in the string should be converted to uppercase".

**(2) Parameter format**

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

**(3) Timestamp**

Use the Linux timestamp in the string format.

**(4) Payment for Order Number of the Same Merchant**

## 7 、 Error Code

The merchant's out\_trade\_no must be globally unique, and a unique order number is required for both the commissioning and production environment.

Promise code	meaning	Terminal operation	Terminal display (recommended)	Applicable conditions
00	Acceptance or successful transaction	success	Transaction successful	
01	Check the issuer	fail	Please contact the issuing bank with the cardholder. tie	The reason why the card issuer refused the transaction was that it had to ask for contact.  This response code is only used in the case of the issuing bank.
03	Invalid merchant	fail	Invalid merchant	MCC anomaly; The merchant (MCC) with this card does not allow this transaction; This merchant is in the blacklist.
04	Confiscation card	Card-cutting and confiscation	This card should be swallowed (ATM) This card is an invalid card (POS)	The card issuer is convinced that the card should be cancelled.

05	Authentication failed.	fail	Cardholder authentication failed.	<ol style="list-style-type: none"> <li>1. The transaction information of online transactions is overdue.</li> <li>2. The cardholder's identity authentication fails (e.g. the entrustment relationship or online Class transactions)</li> <li>3. The certificate information (type, number, etc.) is inconsistent.</li> <li>4. Transaction center judges the security information and transaction information. More than 24 hours difference</li> <li>5. The verification of the cardholder's date of birth is inconsistent.</li> <li>6. In the withdrawal business of agricultural assistance, the receiver did not send card information.</li> <li>7. CVN2 inspection failed.</li> <li>8. Failed to verify transaction matching.</li> </ol>
10	Partial amount approval	Success, need prompt.	Show part of the approved amount, and Show operator	Allow it to be used in partial amount transactions.
11	VIP Approval (VIP)	success	Is this VIP customer.	The issuing bank indicates that this is a VIP customer.

12	Invalid related party transaction of	fail	Invalid transaction	<p>1. The original transaction has not been accepted, and the related customs clearance has been received. Joint transactions, such as reversal transactions and cancellation transactions;</p> <p>2. Transactions that should occur every other day do not occur every other day.</p> <p>3. Cancel and reverse the original transaction every other day.</p> <p>4. The transaction was not executed, but the information of related party transactions was received (example For example, the pre-authorization transaction is not accepted, and the pre-authorization is completed. Or pre-authorization to cancel the transaction)</p>
13	Invalid amount	fail	Invalid amount	<p>In the transaction where there should be an effective amount, the amount field should be filled with 0 or its Its illegal value;</p> <p>Accumulated overpayment amount/accumulated overpayment amount;</p> <p>Exceeding the proportion of transaction consumption;</p> <p>Tip amount exceeds limit.</p> <p>This institution cannot/cannot trade in this currency;</p>

14	Invalid card number (without this account number)	fail	Invalid card number	<ol style="list-style-type: none"> <li>1. The card issuer does not have this primary account number.</li> <li>2. When the original transaction is found, the main account of the related party transaction. No. does not match the original transaction primary account number.</li> <li>3. The check digit of card number is incorrect.</li> <li>4. The account has been cancelled or cancelled.</li> <li>5. The primary account number of the responding transaction does not match the primary account number of the requesting transaction.</li> </ol>
15	There is no card issuer.	fail	This card has no corresponding issuer.	The corresponding card issuer cannot be found according to the main account number of the transaction request.
16	Update the third track.	success	Update the third track	reserve
21	Card is not initialized.	fail	The card is not initialized or sleep card.	<ol style="list-style-type: none"> <li>1. The card has not been started or opened;</li> <li>2. The initial password of the card has not changed;</li> <li>3. Transactions restricted by initial password</li> <li>4. Cards that have not been used for a long time and are frozen or in a "sleep" state.</li> </ol>

22	Doubt, related party transaction error	fail	Error in operation, or transaction exceeded. Allowed days	<p>Abnormal related party transactions, such as the following:</p> <ol style="list-style-type: none"> <li>1. After the reversal transaction is executed, the cancellation request is received. trade</li> <li>2. The current transaction has been cancelled and its related transactions have been received. Such as reversal, cancellation, etc.</li> <li>3. After executing the pre-authorization to cancel the transaction, the pre-authorization is received again. Right to complete the transaction</li> <li>4. After the pre-authorization transaction is reversed, the pre-authorization is received. Right to complete the transaction</li> <li>5. When the pre-authorization is completed, the same information is received again. Pre-authorization completion request of pre-authorization transaction</li> <li>6. Pre-authorization transactions (including pre-authorization completion and pre-authorization Cancellation) occurred more than the allowed pre-authorization trading days. count</li> <li>7. Exceeding the normal payment time</li> </ol>
----	--	------	--	---

25	Can't find the original cross. easy	fail	No original transaction, please contact Issuer	<p>It can be expressed as follows:</p> <ol style="list-style-type: none"> <li>1. If the original transaction cannot be found, the original request will be matched. wrong</li> <li>2. Failed to match the original pre-authorization and authorization transaction.</li> <li>3. The reversal transaction request fails to match the original transaction.</li> <li>4. It is used when deducting fees, revoking and changing entrustment, and the entrustment relationship. non-existent</li> </ol>
30	Message format error	fail	Please try again.	<p>It can be expressed as follows:</p> <ol style="list-style-type: none"> <li>1. The required message domain does not appear in the message.</li> <li>2. The value of transaction pipeline is not in the standard definition.</li> <li>3. Error in domain resolution.</li> <li>4. Error in subdomain resolution.</li> <li>5. Domain check failed.</li> <li>6. Illegal characters appear in the domain</li> <li>7. bitmap in the received message does not meet the definition of the specification.</li> <li>8. Error in track information.</li> <li>9. There is no transaction amount in the transaction where the transaction amount should appear.</li> <li>10. The VR letter of 61.6 domain should be provided in the message but not provided.</li> </ol> <p>have a rest</p>

34	Suspected of cheating	Card-cutting and confiscation	Cheating card	<p>The card is suspected of cheating (including ARQC error), ATM</p> <p>The card is confiscated by the operator, which is suitable for the following situations:</p> <ol style="list-style-type: none"> <li>1. The number of CVN errors exceeds the limit of card swallowing times;</li> <li>2. The card has been counterfeited (debit)</li> </ol>
38	Exceeded PIN trial input	fail	The number of password errors exceeds the limit, please contact the card issuer	<p>The number of wrong passwords exceeds the limit, and the account has been locked. Please hold it. Card to the issuer for unlocking.</p>

40	The requested feature is not yet supported.	fail	Transactions not supported by the card issuer	<p>For functions not supported by the organization, it can be expressed as follows:</p> <ol style="list-style-type: none"> <li>1. The card issuer has not opened this transaction.</li> <li>2. Although the transaction can be determined from the message of the networking organization Type, but the transaction is not open at present</li> <li>3. Although the networking organization can be determined from the received message The type of transaction, but in the receiving party's permission list or special permission. The transaction is not included in the table.</li> <li>4. Although the transaction can be determined from the message of the networking organization Type, but the message version of the receiver does not support it.</li> <li>5. For an IC card transaction, if the receiver is Early State, but the receiver does not require to check ARQC.</li> <li>6. The card issuer cannot check some verification elements.</li> </ol>
41	Loss reporting card	Card-cutting and confiscation	The loss of this card has been reported, and the loss of card (POS) has been reported by ATM.	Lost card, engulfed

43	Stolen card	Card-cutting and confiscation	This card has been confiscated, please contact the card issuer. Party (ATM) Stolen card (POS)	The card issuer confirms that this card is stolen and engulfs it.
45	Downgrading transactions are not allowed.	fail	Please use the chip.	1. The card issuer does not support the downgrade transaction of composite card. 2. The card issuer does not support the acceptance of initiated composite cards in this region. Degraded transaction
51	Fund shortage	fail	Insufficient available balance	Insufficient available account balance, insufficient credit limit, cash withdrawal limit exceed the limit
54	Expired card	fail	The card has expired.	1. Expired card with incorrect expiration date. 2. Transactions that should be sent with validity period are not sent with validity period.
55	Incorrect PIN	fail	Password error	PIN verification failed.
57	Cardholders are not allowed Make a deal.	fail	This card transaction is not allowed.	The credit and risk status of the card issuer to the cardholder and other reasons, no Circumstances in which transactions are allowed, including but not limited to: 1. This card type cannot make this kind of transaction. 2. Beyond the scope of service 3. The card will not be accepted. 4. No deposit can be made with the company card. 5. This account does not have

				<p>this currency.</p> <p>6. This card is suspected of cashing out.</p> <p>7. Card number or ID number is in the blacklist.</p>
58	Terminal access is not allowed Transaction of the line	fail	The card issuer does not allow the card to be in this account. Terminal makes this transaction.	<p>1, the card issuer in restricting such terminals for related transactions (can be targeted at some cards BIN)</p> <p>2. The terminal number in the related transaction is different from that in the original transaction. matching</p>
59	Suspected of cheating	fail	Card check error	CVN verification failed
61	Amount limit exceeded.	fail	Transaction amount exceeds limit	<p>Transaction amount exceeds the limit, including but not limited to:</p> <p>1. Exceeding the single consumption limit/exceeding the ATM single cash withdrawal limit</p> <p>2. ATM daily cash withdrawal /POS daily consumption amount exceeds the limit.</p> <p>3. Super cardholder can define single withdrawal/consumption.</p> <p>4. Exceeding the transfer limit</p>

62	Restricted card	fail	Restricted card	Restricted cards (accepting service area restrictions and other reasons), no swallow up
64	Original amount error	fail	The transaction amount is not equal to the original transaction. match	1. the transaction amount in the request message and the transaction amount in the response message. Easy amount mismatch (except partial deduction) 2. The transaction amount in the related party transaction message is the same as the original transaction report. The transaction amount in the text does not match (except for some deductions)
65	Exceeded the withdrawal/consumption limit.	Exceeded the withdrawal limit.	1. Exceeding the limit of withdrawal/consumption times of the day. 2. Accumulated times of over-transfer/over-payment;	
sixty-eight	The issuing bank responded to the excess. time	fail	Transaction timed out, please try again.	The receiving institution timed out and did not receive the reply from the issuer.
75	The allowed input PIN Frequency overrun	fail	Password error times exceeded.	The number of incorrect password entries exceeds the limit.
90	Processing at the end of the day.	fail	System daily cut, please try again later.	Date switching is in progress.

91	Card issuer cannot operate. work	fail	The issuer's status is abnormal, please Try again later.	<p>Used to indicate that due to the fault of the card issuer (or transfer-in/transfer-out party) The transaction is rejected by mistake, such as the following:</p> <ol style="list-style-type: none"> <li>1. The card issuer (or transfer-in/transfer-out party) is not operating normally.</li> <li>2. The card issuer (or transfer-in/transfer-out party) is abnormal, but it is not reconciled.</li> </ol> <p>UnionPay processing center signs authorization agreement.</p> <ol style="list-style-type: none"> <li>3. The card issuer (or the transfer-in/transfer-out party) signs out and fails to sign in.</li> <li>4. The running status of the card issuer (or transfer-in/transfer-out party) is invalid</li> <li>5. The card issuer (or transfer-in/transfer-out party) is processed by UnionPay. close</li> <li>6. The line of the card issuer (or transfer-in/transfer-out party) is abnormal.</li> <li>7. The internal system of the issuing bank (or the transfer-in/transfer-out party) times out.</li> </ol>
92	Financial institutions or Looking for internet facilities Unable to reach or reach. arrive	fail	The card issuer's line is abnormal, please try again later.	<ol style="list-style-type: none"> <li>1. There is no available line.</li> <li>2. IP address format of UnionPay processing center or network access organization and Wrong port number</li> </ol>

94	Repeated transactions	fail	Reject, repeat transaction, please try again later.	<ol style="list-style-type: none"> <li>1. Used to detect that the original transaction is a duplicate transaction;</li> <li>2. When establishing the entrustment, it is found that the entrustment relationship already exists.</li> <li>3. Duplicate transaction serial number.</li> <li>4. Transaction VCODE value is duplicate.</li> </ol>
96	UnionPay processing center System abnormality and failure	fail	Rejected, the exchange center is abnormal, Please try again later	<p>It is used to indicate that due to the error of UnionPay processing center, payment is made.</p> <p>Easy to be rejected, given by UnionPay. In the following cases:</p> <ol style="list-style-type: none"> <li>1. The UnionPay processing center failed to perform normal processing, and it happened.</li> </ol> <p>Such as abnormal database operation, abnormal shared memory operation, function</p> <p>The number of internal processing failure such as abnormal operation.</p> <ol style="list-style-type: none"> <li>2. During the maintenance of UnionPay Processing Center, all requests are rejected.</li> </ol>
97	ATM/POS terminal number cannot find	fail	Terminal number is not registered.	Terminal number is not registered.
98	UnionPay processing center Unable to receive the card issuer. reply	fail	Issuer timeout	<ol style="list-style-type: none"> <li>1. The issuer timed out.</li> <li>2. The transferor timed out.</li> <li>3. Timeout of receiving response</li> </ol>

99	Wrong PIN format	fail	Wrong PIN format, please re-sign arrive	Wrong PIN format
----	------------------	------	---	------------------