



Merchant API Documentation

User-Presented Payment API

Document version: 1.5.1

API version: 2.0

TABLE OF CONTENTS

1、 Introduction	4
1.1Abstract.....	4
1.2Audience.....	4
2、 Program Overview	5
3、 Data Format	8
3.1 Submit Data.....	8
3.2 XML Data Format.....	9
4、 Digital Signature	11
4.1Original string of signature	11
4.2Method of signature	12
5、 User-Presented Payment Interface	15
5.1 Payment Interface	15
5.2 Reversal interface.....	22
5.3 Order Query interface	26
6、 Refund& refund Query Interface	32
6.1Refund interface	32
6.2Retrieve refund result interface.....	39
7 、 Notes	44

8、 Error code	45
1.Swiftpass error code.....	45
2.Wechat and Alipay error code	48
3.UNION PAY error code	50
4.PAYME error code.....	51

Document Changes		
Date	Version	Description
2018-02-24	1.0	First draft
2019-01-24	1.1	Support unionpay
2019-06-12	1.2	Update SHA256signature algorithm.
2022-07-08	1.5	Update mch_id, device_info, time_start, time_expire, trade_state description, delete CLOSED from trade_state added error code
2023-02-20	1.5.1	Update cover and add default MD5 in sign_type Update trade_type

1、 INTRODUCTION

1.1 Abstract

User-Presented Payment is a payment method in which merchants integrate with the Third Party' s API and use a scanner to scan customer's Payment QR Code to complete the payment.



User-Presented Payment can be applied in Off-line payment scene. E.g vending machine, cashier, ticket purchasing, etc.

1.2 Audience

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

2、 PROGRAM OVERVIEW

2.1 Industry Background

Mobile payment is a payment service function provided based on WeChat/Alipay/Union application.etc. It also provides the merchants with supportive functions such as sales, marketing analysis, and management of account and capital. Customers could complete a payment by scanning QR code, presenting a payment QR code to be scanned, or other payment methods.

2.2 Business realization process

User-Presented Payment business (use WeChat as an example)

Usage scenario:

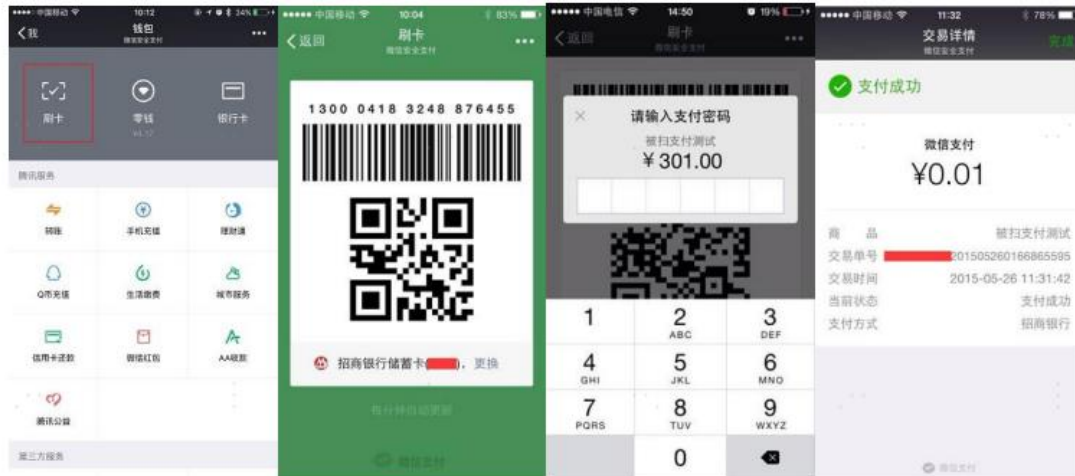
Step 1: Customers select Barcode Payment in WeChat App:

"me" -> "Wallet" -> "Money" show the payment QR code.

Step 2: Cashier creates a payment order in merchant system and customer confirms the order amount.

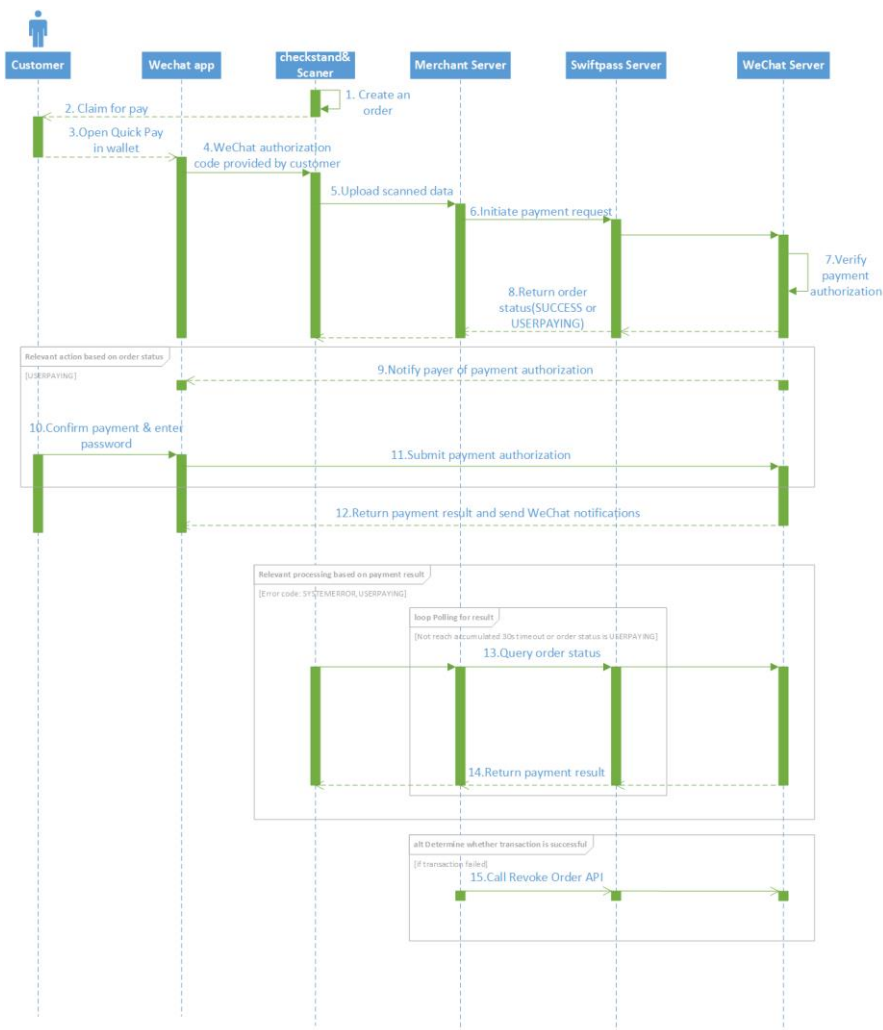
Step 3: Cashier uses scanner device to scan the payment QR code presented by customer. Then merchant system will initiate a request for payment.

Step 4: WeChat will decide if customers need to input the password. A successful payment notification will be prompted after transaction is completed.

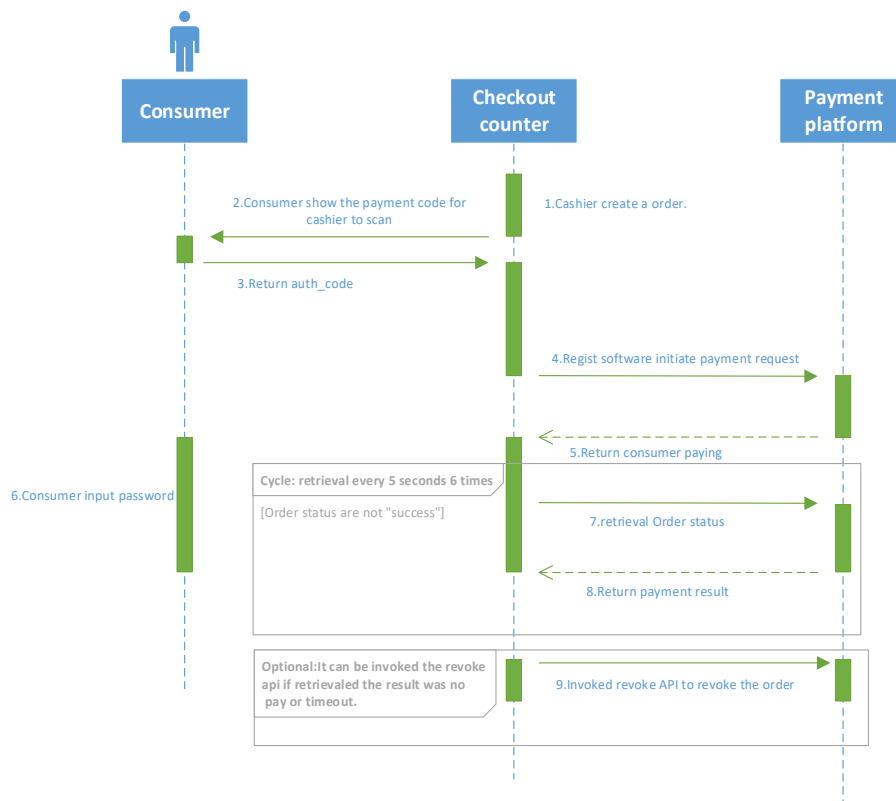


2.3 Process of merchant system for calling API

(1) Process of password verification is not required:



(2) Process of password verification is required



3、 DATA FORMAT

3.1 Submit Data

Uses HTTPS POST protocol. To ensure the recipient receives the correct data, all data must be signed

```

<xml>
<auth_code><![CDATA[xxxxxxxxxxxxxx]]></auth_code>
<body><![CDATA[SALES]]></body>
<device_info><![CDATA[31786672]]></device_info>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[755100001]]></mch_id>
<nonce_str><![CDATA[1234567890]]></nonce_str>
<op_device_id><![CDATA[31786672]]></op_device_id>
<out_trade_no><![CDATA[75510000178667220230210180221A]]></out_trade_no>
    
```

```
<service><![CDATA[unified.trade.micropay]]></service>
<sign><![CDATA[B05969B99574F74A37C44D01DC86FE6040E2C46C4DCEB1B283C52A0F7941351
D]]></sign>
<sign_type><![CDATA[SHA256]]></sign_type>
<total_fee><![CDATA[12400]]></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>
<alipay_buyer_login_id><![CDATA[888-****8888]]></alipay_buyer_login_id>
<alipay_buyer_user_id><![CDATA[2088632237018888]]></alipay_buyer_user_id>
<appid><![CDATA[-]]></appid>
<bank_type><![CDATA[ALIPAYHK]]></bank_type>
<charset><![CDATA[UTF-8]]></charset>
<coupon_fee><![CDATA[0]]></coupon_fee>
<device_info><![CDATA[31786672]]></device_info>
<fee_type><![CDATA[HKD]]></fee_type>
<local_fee_type><![CDATA[HKD]]></local_fee_type>
```

```
<local_total_fee><![CDATA[12400]]></local_total_fee>
<mch_id><![CDATA[991530003201]]></mch_id>
<nonce_str><![CDATA[1676023347103]]></nonce_str>
<openid><![CDATA[2088632237018404]]></openid>
<order_fee><![CDATA[12400]]></order_fee>
<out_trade_no><![CDATA[300032013178667220230210180221A]]></out_trade_no>
<out_transaction_id><![CDATA[2023021022001418401410080599]]></out_transaction_id>
<pay_result><![CDATA[0]]></pay_result>
<payment_inst><![CDATA[ALIPAYHK]]></payment_inst>
<result_code><![CDATA[0]]></result_code>
<sign><![CDATA[5F6CCAC7A16ACC087664E8ED49576A1A]]></sign>
<sign_type><![CDATA[MD5]]></sign_type>
<status><![CDATA[0]]></status>
<time_end><![CDATA[20230210180227]]></time_end>
<total_fee><![CDATA[12400]]></total_fee>
<trade_type><![CDATA[pay.alipay.micropay]]></trade_type>
<transaction_id><![CDATA[991530003201202302101165615820]]></transaction_id>
<uuid><![CDATA[144985b240585ecfa5b824012448bd041]]></uuid>
<version><![CDATA[2.0]]></version>
</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>
```

4、 DIGITAL SIGNATURE

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

There are two steps in digital signature.

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

Generally, failed results 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, be it 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 Query String (i.e. key1=value1&key2=value2...), and the null value should not be added to this string and will not participate in formation of string of signature.

2. In the original string of signature, both field name and value will adopt original values and will not conduct URL Encode.
3. Responses 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[测试支付]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[7551000001]]></mch_id>
<nonce_str><![CDATA[1409196838]]></nonce_str>
<out_trade_no><![CDATA[141903606228]]></out_trade_no>
<service><![CDATA[unified.trade.micropay]]></service>
<sign><![CDATA[52836FAD27E0813DAA4072A4BDA9F654]]></sign>
<total_fee><![CDATA[1]]></total_fee>
</xml>
```

The signature field sequence:

```
body=测试支付
&mch_create_ip=127.0.0.1&mch_id=7551000001&nonce_str=1409196838&out_trade_no=141903606228&service=unified.trade.micropay&total_fee=1
```

4.2 Method of signature

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.

MD5 signature calculation formula:

$$\text{sign} = \text{Md5}(\text{ "The signature field sequence strings" } \&\text{key} = \text{ "signature key" }). \text{toUpperCase}$$

Example:

Suppose the following are incoming parameters of XML:

```
<xml>
<body><![CDATA[测试支付]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[7551000001]]></mch_id>
<nonce_str><![CDATA[1409196838]]></nonce_str>
<out_trade_no><![CDATA[141903606228]]></out_trade_no>
<service><![CDATA[unified.trade.micropay]]></service>
<sign><![CDATA[52836FAD27E0813DAA4072A4BDA9F654]]></sign>
<total_fee><![CDATA[1]]></total_fee>
</xml>
```

Merchant signature key : 9d101c97133837e13dde2d32a5054abb

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

body=测试支付

&mch_create_ip=127.0.0.1&mch_id=7551000001&nonce_str=1409196838&out_trade_no=141903606228&service=unified.trade.micropay&total_fee=1

ii:the second step of which is to choose MD5 and keys to calculate the result of signature(sign):

sign

```
=md5 (string1&key=9d101c97133837e13dde2d32a5054abb) .toUpperCase  
=md5 (body=测试支付  
&mch_create_ip=127.0.0.1&mch_id=7551000001&nonce_str=1409196838&o  
ut_trade_no=141903606228&service=unified.trade.micropay&total_fee  
=1&key=  
9d101c97133837e13dde2d32a5054abb) .toUpperCase ()  
="52836FAD27E0813DAA4072A4BDA9F654"
```

SHA256 signature calculation formula:

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

Example:

There are XML afferent parameters:

```
<xml>  
<auth_code>135187250012923035</auth_code>  
<body>test</body>  
<charset>UTF-8</charset>  
<mch_create_ip>127.0.0.1</mch_create_ip>  
<mch_id>127530000052</mch_id>  
<nonce_str>1542940680925</nonce_str>  
<out_trade_no>1542940643087</out_trade_no>  
<service>unified.trade.micropay</service>  
<sign>10F2F6DC0D5E008B967CC3C86FC58179686B4EE42F4F68B4A7668501B60  
30C29</sign>  
<sign_type>SHA256</sign_type>  
<total_fee>2</total_fee>  
<version>2.0</version>  
</xml>
```

Merchant signature key: 18e0a2ad5d5571af14b855fcf33091f4

i:the first step of which is to connect the original string(string1) that

needs signature according to certain rules:

```
auth_code=135187250012923035&body=test&charset=UTF-  
8&mch_create_ip=127.0.0.1&mch_id=127530000052&nonce_str=154294068
```

```
0925&out_trade_no=1542940643087&service=
unified.trade.micropay&total_fee=1
```

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(auth_code=135187250012923035&body=test&charset=UTF-
8&mch_create_ip=127.0.0.1&mch_id=127530000052&nonce_str=154294068
0925&out_trade_no=1542940643087&service=
unified.trade.micropay&total_fee=1&key=
18e0a2ad5d5571af14b855fcf33091f4).toUpperCase()
="10F2F6DC0D5E008B967CC3C86FC58179686B4EE42F4F68B4A7668501B6030C2
9"
```

5、 USER-PRESENTED PAYMENT INTERFACE

5.1 Payment Interface

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

Request type: POST

Request format: XML

Request Parameters:

Field Name	Required	Type	Description
Normal Parameters			
service	Yes	String(32)	Interface type :unified.trade.micropay
version	No	String(8)	Version number. default value: 2.0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256

			Default value: MD5
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (ID of ordinary merchant or store).
out_trade_no	Yes	String(32)	The unique trade reference(merchant order id of 5-32 bits) in merchant's system. Letter, number and underline are allowed. Case-sensitive
device_info	No	String(32)	Specifies a Terminal device id (can be used to filter results at SwiftPass merchant platform).
body	Yes	String(127)	Description of merchants' goods
attach	No	String(127)	Merchant additional information.
total_fee	Yes	Int	Integral number is allowed only. The unit of the fee is the minimal unit of the local currency.
mch_create_ip	Yes	String(16)	Specifies the machine IP that calls the API.
auth_code	Yes	String(1024)	Specifies the authorization code for scanning a barcode/QR Code on User-Presented Payment.
time_start	No	String(14)	Order created date. Format : yyyyMMddHHmss. i.e.20091225091010. GMT+8 Beijing Time. Range: 1) WeChat: 1 minute to 2 hours. If not passed, default value will be 10 minutes; 2) Alipay: 1 minute to 15 days. If not passed, default value will be 2 hours; 3) UPI: no limits. If not passed, default value will be 10 minutes.
time_expire	No	String(14)	Order timeout date. Format:yyyyMMddHHmss. i.e.20091225091010. GMT+8 Beijing Time

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

Return 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)	Value : UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	"0" : success. Others value : fail.

			Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.
message	No	String(128)	Return message. Only return when the signature verification invalid.
need_query	No	String(1)	Y:Need query; N:not need to;
The following fields will returned when status is "0"			
result_code	Yes	String(16)	"0" : success. Others value : fail.
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (ID of ordinary merchant or store).
device_info	No	String(32)	Specifies a Terminal device id (can be used to filter results on 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
err_msg	No	String(128)	Error information description

sign	Yes	String(34 4)	Please refer to section 4 'Digital Signature'.
The following fields will returned when status and result_code both are "0"			
openid	No	String(1 28)	Unique user identification under the current appid.
sub_openid	No	String(1 28)	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.
trade_type	Yes	String(32)	pay.weixin.proxy.micropay.intl pay.aps.alipay.micropay pay.alipay.micropay pay.weixin.wallet.micropay.intl pay.upi.micropay.intl
is_subscribe	No	String(1)	Specifies whether the payer is following the associated official account or not. "Y" means 'following', and "N" means 'not following'.
pay_result	Yes	Int	Payment result. "0": success. Others fail.
pay_info	No	String(64)	Payment result information. Payment successful return null.

payment_in st	No	String(64)	User payment wallet type. It will return when using Alipay 'ALIPAYCN': Chinese mainland wallet 'ALIPAYHK': HK wallet
transaction_ id	Yes	String(32)	platform order id
out_transact ion_id	Yes	String(32)	order number provided by the third-party (WeChat /Alipay/Union)
sub_is_subs cribe	No	Int	Specifies whether the payer is following the associated official account or not. "Y" means 'following', and "N" means 'not following'.
sub_appid	No	String	Specifies an Official Account ID assigned by WeChat.
out_trade_n o	Yes	String(32)	Specifies an order number created by merchant's system, which is consistent with request.
total_fee	Yes	Int	Specifies the total amount for a transaction. The unit of the fee is the minimal unit of the currency.
coupon_fee	No	Int	Coupon amount. coupon_fee <= total_fee. total_fee - oupon_fee = cash pay amount
fee_type	No	String(8)	Complies with ISO 4217 standards.

attach	No	String(127)	Merchant's data package to be returned intact
bank_type	No	String(16)	String states bank_type
bank_billno	No	String(32)	Bank order number. Null for WeChat payment.
time_end	Yes	String(14)	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. GMT+8 Beijing Time.
cash_fee	No	Int	Cash amount of order. The unit of the fee is the minimal unit of CNY. (cent)
cash_fee_type	No	String(16)	User Payment currency. It will return when using WeChat 'CNY': Chinese mainland wallet 'HKD': HK wallet
rate	No	String(16)	Exchange rate between customers' payment currency and merchants settlement currency.

1. When payment interface is called, the disposal of each case could refer to the following schemes:

2. After payment is requested: if both returned status value and result code are 0, the payment of the order will be deemed successful; if the returned value of status is '0' , result code is not '0' , and the value of need_query is 'N' , there is no need to call query API. In other cases (e.g. need_query is not returned or returned value of need_query is 'Y'), query API must be called to confirm order status.

3. Suggestion for calling query API: The interface shall be called 6 times, the time interval between each call will be 5 seconds (both time can be customized, we recommend inquiry time for WeChat/Alipay transactions shall be more than 30 seconds, and UPI transactions query shall be more than 70 seconds). If the interface does not return the success identification after 6 inquiries for WeChat/Alipay transactions or 12 inquiries for UPI transactions (meaning returned value of trade_state is not 'SUCCESS'), reversal interface should be called and the order will be closed.

5.2 Reversal interface

In case the transaction of payment fails to return a value or the payment system is timed out, then the transaction needs to be canceled and the reversal interface should be called.

Interface logic: orders with failed payment will be closed, and orders with successful payment will be revoked. Note: only orders made within 5 minutes could be reversed, for other successful orders, please call the refund API. It is recommended to call the reversal API at least 15s (70s for UPI payments) after calling payment API.

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

Request type: POST

Request format: XML

Request Parameters:

POST request with content of XML

Field Name	Required	Type	Description
Normal Parameters			
service	Yes	String(32)	Value : unified.micropay.reverse
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5

mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants and stores).
out_trade_no	Yes	String(32)	The unique trade reference(merchant order id of 5-32 bits) in merchant's system. Letter, number and underline are allowed. Case-sensitive
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(44)	Please refer to section 4 'Digital Signature'.

Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5

status	Yes	String(16)	<p>“0” : success. Others value : fail.</p> <p>Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.</p>
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will returned when status is “0”			
result_code	Yes	String(16)	<p>“0”for SUCCESS. others for FAIL.</p> <p>SUCCESS indicates the order was cancelled for successfully and cannot be paid for again. If the payment is completed, a refund is initiated.</p> <p>FAIL refers to exceptions that occur in the interface. The recall function should be used to determine whether the order has been canceled or not.</p>
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants and stores).
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
err_code	No	String(32)	Reference error code.

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

5.3 Order Query interface

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

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

Request type: POST

Request format: XML

Request Parameters:

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Value : unified.trade.query
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	Value : UTF-8

sign_type	No	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants and stores)
out_trade_no	No	String(32)	The unique trade reference of merchant system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.

Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	"0" : success. Others value : fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of trade_state .
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will returned when status is "0"			
result_code	Yes	String(16)	"0" : success. Others value : fail.
mch_id	Yes	String(32)	Merchant ID, Specifies an official account id assigned by platform (for ordinary merchants and stores).

device_info	No	String(32)	Specifies a Terminal device id (used to filter results on 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
err_msg	No	String(128)	Error information description
sign	Yes	String(44)	Please refer to section 4 'Digital Signature'.
The following fields will returned when status and result_code both are "0"			
trade_state	Yes	String(32)	<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: Payment in progress</p> <p>PAYERROR: Payment failed (other reasons, e.g. cannot be returned by bank)</p>
The following fields will returned when trade_state is "SUCCESS"			

trade_type	Yes	String(32)	<p>pay.weixin.proxy.micropay.intl——WeChat</p> <p>pay.alipay.micropay.intl——Alipay</p> <p>pay.upi.micropay.intl——Union pay</p> <p>pay.gcash.micropay——Gcash</p> <p>pay.grab.micropay——Grab</p> <p>pay.payme.micropay——PayMe</p>
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. Unique to each appid.
is_subscribe	No	String(1)	Specifies whether the payer is following the associated official account or not. "Y" means 'following', and "N" means 'not following'.
sub_is_subscribe	No	String(1)	Specifies whether the payer is following the associated official account or not. "Y" means 'following', and "N" means 'not following'.

payment_inst	No	String(64)	User payment wallet type. It will return when using Alipay 'ALIPAYCN': Chinese mainland wallet 'ALIPAYHK': HK wallet
transaction_id	Yes	String(32)	platform order id
out_transaction_id	Yes	String(32)	order number provided by the third-party (WeChat, Alipay or UnionPay)
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. $coupon_fee \leq total_fee$. $total_fee - coupon_fee = cash\ pay\ amount$
fee_type	No	String(8)	Complies with ISO 4217 standards.
attach	No	String(127)	Specifies merchant's data package, which is returned as it is.
bank_type	No	String(16)	String states bank_type

bank_billNo	No	String(32)	Bank order number. Null for WeChat payment.
time_end	Yes	String(14)	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. GMT+8 Beijing.
cash_fee	No	Int	Cash amount of order, The unit of the fee is the minimal unit of CNY.
cash_fee_type	No	String(16)	User Payment currency. It will return when using WeChat 'CNY': Chinese mainland wallet 'HKD': HK wallet
rate	Yes	String(16)	Exchange rate between user payment currency and merchants settlement currency.

6、 REFUND& REFUND QUERY INTERFACE

6.1 Refund interface

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:

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Value : unified.trade.refund
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants or stores).
out_trade_no	No	String(32)	The unique trade reference of merchant system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.

transaction_id	No	String(32)	The unique trade reference of platform system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
out_refund_no	Yes	String(32)	Specifies the internal refund number, which is unique in the system. A single transaction can be processed as multiple partial refunds, with the total sum of the partial refunds being equal to the original one. If the refund is not successful. The recall function should be used with same refund number to avoid duplication of refunds.
total_fee	Yes	Int	The total amount for a transaction. Integral number is allowed only. 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(34)	Please refer to section 4 'Digital Signature'.

Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5

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 invalid.
The following fields will returned when status is “0”			
result_code	Yes	String(16)	“0”: success. Others value : fail.
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants and stores).
device_info	No	String(32)	Specifies a Terminal device id (can be used to filter results on 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
err_msg	No	String(128)	Error information description

sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
The following fields will returned when status and result_code both are "0"			
transaction_id	Yes	String(32)	The unique trade reference of platform system.
out_trade_no	Yes	String(32)	The unique transaction order id of merchant system.
out_refund_no	Yes	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id	Yes	String(32)	Specifies the internal refund number, which is unique in the platform system.
refund_channel	Yes	String(16)	Value: ORIGINAL. The money will refund back to where it came from.
refund_fee	Yes	Int	Refund amount. The unit of the fee is the minimal unit of the local currency. Partial refund can be supported.
coupon_refund_fee	No	Int	Coupon refund amount. $coupon_refund_fee \leq refund_fee$. $refund_fee - coupon_refund_fee = \text{cash refund amount}$

6.2 Retrieve refund result interface

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

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

Request type: POST

Request format: XML

Request Parameters:

Field Name	Required	Type	Description
service	Yes	String(32)	Value: unified.trade.refundquery
version	No	String(8)	Version number. default value : 2.0
charset	No	String(8)	Value : UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants and stores).

out_trade_no	No	String(32)	The unique trade reference of merchant system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. out_trade_no and transaction_id at least one required. transaction_id priority when both be filled.
out_refund_no	No	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id	No	String(32)	Specifies the internal refund number, which is unique in the platform system. out_refund_no and refund_id at least one required. refund_id priority when both be filled.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.

Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value : 2.0
charset	Yes	String(8)	Value : UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256: SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	"0" : success. Others value : fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will returned when status is "0"			
result_code	Yes	String(16)	"0" : success. Others value : fail.
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform (for ordinary merchants and stores).

device_info	No	String(32)	Specifies a Terminal device id (can be used to filter results on 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
err_msg	No	String(128)	Error information description
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
The following fields will returned when status and result_code both are "0"			
transaction_id	Yes	String(32)	The unique trade reference of platform system.
out_trade_no	Yes	String(32)	The unique trade reference of merchant system.
refund_count	Yes	Int	Specifies recorded refund counts.
out_refund_no_\$n	Yes	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id_\$n	Yes	String(32)	Specifies the internal refund number, which is unique in the platform system.

refund_channel_\$n	Yes	String(16)	Value : ORIGINAL. The money will refund back to where it came from.
refund_fee_\$n	Yes	Int	Refund amount in cents. Partial refund can be supported.
coupon_refund_fee_\$n	No	Int	Coupon refund amount. coupon_refund_fee <= refund_fee. refund_fee - coupon_refund_fee = cash refund amount
cash_fee	No	Int	Cash amount of order. The unit of the fee is the minimal unit of CNY.
cash_fee_type	No	String(16)	User Payment currency. It will return when using WeChat 'CNY': Chinese mainland wallet 'HKD': HK wallet
rate	Yes	String(16)	Exchange rate between user payment currency and merchants settlement currency
refund_time_\$n	No	String(14)	Format : yyyyMMddHHmmss

refund_status _ \$n	Yes	String(1 6)	<p>Refund Status:</p> <p>SUCCESS: Refund successful</p> <p>FAIL: Refund failed.</p> <p>PROCESSING: pending refund.</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 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>			

7、 NOTES

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

Other notes

(1) Case issue for parameters

Please pay attention to the case requirements in the document, such as

“after md5 operation, the character of the character string needs to be converted to upper case letter” .

(2) Parameter format issues

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

(3) Time stamp issues

Please use Linux time stamp in character string format.

(4) Order number payment made under the same merchant

Merchants' out_trade_no must be unique.

8、 ERROR CODE

1.Swfitpass error code

Error Code	Error Message
Auth code invalid	Invalid payment code
400	Payment code can not 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 a
REFUND_FEE_INVALID	Invalid refund amount
验证参数失败	Parse params error
ORDER_FEE_INVALID	Order amount invalid
ORDER_DATE_INVALID	Order date invalid
transaction_id out_trade_no is required	缺少参数
buyer_logon_id buyer_id is required	缺少参数
transaction_id out_trade_no out_refund_no refund_id is required	缺少参数
out_refund_no refund_id is required	缺少参数
Refundid and refundpwd is required	缺少参数
thi_mch_id is required	缺少参数
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, protocal error
Internal error	Wechat request error, unkown 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, protocal error
Internal error	Third-party request error, unkown error
Order reverse	Order is reversed
400	Unsupport Api
400	商户未开通[XXX]支付类型
400	Merchant is suspended
400	Merchant can not 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
金额限制	PRODUCT_AMOUNT_LIMIT_ERROR
BALANCE_NOT_ENOUGH	Insufficient balance
USERPAYING	微信需要用户输入密码
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 can not be reversed
Refund limit	Do not support this function
400	Order status error, can not 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
APPID_MCHID_NOT_MATCH	appid 和 mch_id 不匹配
APPID_NOT_EXIST	参数中缺少 APPID
Auth code invalid	付款码无效
CONTEXT_INCONSISTENT	交易信息不对等
ILLEGAL_ARGUMENT	参数错误
ILLEGAL_EXTERFACE	接口配置错误
INVALID_FEE	金额参数有误
INVALID_REQUEST	参数格式有误或者未按规则上传
INVALID_TRANSACTIONID	请求参数未按指引进行填写
LACK_PARAMS	缺少必要的请求参数
MCHID_NOT_EXIST	参数中缺少 MCHID
NOT_UTF8	未使用指定编码格式
ORDER ERROR	订单错误
PARAM_ERROR	请求参数未按指引进行填写
POST_DATA_EMPTY	post 数据不能为空
REQUIRE_POST_METHOD	未使用 post 传递参数
Signature error	参数签名结果不正确
SIGNERERROR	商户号填写错误
XML_FORMAT_ERROR XML	XML 格式错误
OUT_TRADE_NO_USED	同一笔交易不能多次提交
0100	CFT 订单不存在
ORDERNOTEXIST	此交易订单号不存在
BANKERROR	银行端超时
Internal error	系统处理超时

SYSTEMERROR	系统超时
ORDERCLOSED	当前订单已关闭，无法支付
ORDERREVERSED	当前订单已经被撤销
TRADE_HAS_CLOSE	相应的交易已结束，不允许进行当前操作
USER_FACE_PAYMENT_SWITCH_OFF	支付方式已关闭
0013	风控
BIZERR_NEED_RETRY	并发情况下，业务被拒绝，商户重试即可解决
ERROR	申请退款业务发生错误
HAS_NO_PRIVILEGE	没有权限
ILLEGAL_PARTNER_EXTERFACE	partner ID 没有访问权限
NOAUTH	商户未开通此接口权限
TRADE_BUYER_NOT_MATCH	用户不匹配
TRADE_ERROR	订单错误
ORDERPAID	订单号已支付
BUYER_BALANCE_NOT_ENOUGH	买方的支付宝账户没有足够的余额用于当前操作。
NOTENOUGH	用户账号余额不足
AUTH_CODE_ERROR	请求参数未按指引进行填写
AUTH_CODE_INVALID	收银员扫描的不是微信支付的条形码
AUTHCODEEXPIRE	用户的条形码已经过期
SOUNDWAVE_PARSER_FAIL	条形码不正确。
ACCOUNTERROR	无效的用户账户
BUYER_ENABLE_STATUS_FORBID	由于安全问题，输入买方账户已被禁用
BUYER_MISMATCH	暂不支持同一笔订单更换支付方
BUYER_NOT_EXIST	输入"buyer_identity_code"不正确，我们根据它找不到账户
BUYER_SELLER_EQUAL	输入买方账户等于卖方账户
CLIENT_VERSION_NOT_MATCH	客户端版本不匹配
EXIST_FORBIDDEN_WORD	根据中国法律，输入上下文内容为禁词
NOTSUPPORTCARD	用户使用卡种不支持当前支付形式
PAYERROR	支付错误
TRADE_STATUS_ERROR	当前操作不允许相应的交易状态
0001	输入错误的密码
UNKNOW	消费者输入密码
USERPAYING	该笔交易因为业务规则要求，需要用户输入支付密码。
USERPAYING	用户支付中
REVERSE_EXPIRE	订单无法撤销
TRADE_OVERDUE	订单已经超过可退款的最大期限(支付后一年内可退款)
FREQUENCY_LIMITED	2个月之前的订单申请退款有频率限制
INVALID_REQ_TOO_MUCH	连续错误请求数过多被系统短暂屏蔽

2. Wechat and Alipay error code

Error Code	Error Message
APPID_MCHID_NOT_MATCH	appid 和 mch_id 不匹配

APPID_NOT_EXIST	参数中缺少 APPID
Auth code invalid	付款码无效
CONTEXT_INCONSISTENT	交易信息不对等
ILLEGAL_ARGUMENT	参数错误
ILLEGAL_EXTERFACE	接口配置错误
INVALID_FEE	金额参数有误
INVALID_REQUEST	参数格式有误或者未按规则上传
INVALID_TRANSACTIONID	请求参数未按指引进行填写
LACK_PARAMS	缺少必要的请求参数
MCHID_NOT_EXIST	参数中缺少 MCHID
NOT_UTF8	未使用指定编码格式
ORDER_ERROR	订单错误
PARAM_ERROR	请求参数未按指引进行填写
POST_DATA_EMPTY	post 数据不能为空
REQUIRE_POST_METHOD	未使用 post 传递参数
Signature error	参数签名结果不正确
SIGNERROR	商户号填写错误
XML_FORMAT_ERROR XML	XML 格式错误
OUT_TRADE_NO_USED	同一笔交易不能多次提交
0100	CFT 订单不存在
ORDERNOTEXIST	此交易订单号不存在
BANKERROR	银行端超时
Internal error	系统处理超时
SYSTEMERROR	系统超时
ORDERCLOSED	当前订单已关闭，无法支付
ORDERREVERSED	当前订单已经被撤销
TRADE_HAS_CLOSE	相应的交易已结束，不允许进行当前操作
USER_FACE_PAYMENT_SWITCH_OFF	支付方式已关闭
0013	风控
BIZERR_NEED_RETRY	并发情况下，业务被拒绝，商户重试即可解决
ERROR	申请退款业务发生错误
HAS_NO_PRIVILEGE	没有权限
ILLEGAL_PARTNER_EXTERFACE	partner ID 没有访问权限
NOAUTH	商户未开通此接口权限
TRADE_BUYER_NOT_MATCH	用户不匹配
TRADE_ERROR	订单错误
ORDERPAID	订单号已支付
BUYER_BALANCE_NOT_ENOUGH	买方的支付宝账户没有足够的余额用于当前操作。
NOTENOUGH	用户账号余额不足
AUTH_CODE_ERROR	请求参数未按指引进行填写
AUTH_CODE_INVALID	收银员扫描的不是微信支付的条形码
AUTHCODEEXPIRE	用户的条形码已经过期
SOUNDWAVE_PARSER_FAIL	条形码不正确。

ACCOUNTERROR	无效的用户账户
BUYER_ENABLE_STATUS_FORBID	由于安全问题，输入买方账户已被禁用
BUYER_MISMATCH	暂不支持同一笔订单更换支付方
BUYER_NOT_EXIST	输入“buyer_identity_code”不正确，我们根据它找不到账号，请检查
BUYER_SELLER_EQUAL	输入买方账户等于卖方账户
CLIENT_VERSION_NOT_MATCH	客户端版本不匹配
EXIST_FORBIDDEN_WORD	根据中国法律，输入上下文内容为禁词
NOTSUPPORTCARD	用户使用卡种不支持当前支付形式
PAYERROR	支付错误
TRADE_STATUS_ERROR	当前操作不允许相应的交易状态
0001	输入错误的密码
UNKNOW	消费者输入密码
USERPAYING	该笔交易因为业务规则要求，需要用户输入支付密码。
USERPAYING	用户支付中
REVERSE_EXPIRE	订单无法撤销
TRADE_OVERDUE	订单已经超过可退款的最大期限(支付后一年内可退款)
FREQUENCY_LIMITED	2个月之前的订单申请退款有频率限制
INVALID_REQ_TOO_MUCH	连续错误请求数过多被系统短暂屏蔽

3. UNION PAY error code

Error Code	Error Message
00	承兑或交易成功
01	查发卡方
03	无效商户
04	没收卡
05	身份认证失败
10	部分金额批准
11	重要人物批准 (VIP)
12	无效的关联交易
13	无效金额
14	无效卡号 (无此账号)
15	无此发卡方
16	批准更新第三磁道
21	卡未初始化
22	故障怀疑，关联交易错误
25	找不到原始交易
30	报文格式错误
34	有作弊嫌疑
38	超过允许的 PIN 试输入

40	请求的功能尚不支持
41	挂失卡
43	被窃卡
45	不允许降级交易
51	资金不足
54	过期的卡
55	不正确的 PIN
57	不允许持卡人 进行的交易
58	不允许终端进 行的交易
59	有作弊嫌疑
61	超出金额限制
62	受限制的卡
64	原始金额错误
65	超出取款/消费次数限制
68	发卡行回应超 时
75	允许的输入 PIN 次数超限
90	正在日终处理
91	发卡方不能操 作
92	金融机构或中 间网络设施找 不到或无法达 到
94	重复交易
96	银联处理中心 系统异常、失效
97	ATM/POS 终端号 找不到
98	银联处理中心 收不到发卡方 应答
99	PIN 格式错

4. PAYME error code

Error Code

EW001 - EW008, EW012

Error Message

PayMe unable to process payment

EW2001	PayCode is expired.
EW2005	Business Annual Transaction limit is reached.
EW2017	Payment failed.
EW2035	Unable to process payment.
EW2036	PayMe user Annual Pay limit is reached.
EW2037	Merchant's wallet balance limit exceeded.
EW2038	PayMe user Daily Pay limit is reached.
EW2039	PayMe user Monthly Pay limit is reached.
EA000	Unexpected API Gateway Exception
EA001	Server request authentication failed
EA001	API gateway authentication failed
EA002	Server request validation failed
EA002	API gateway validation failed
EA003	Service request execution failed
NA	Service request time out
EA008	Request header not acceptable
EA009	Too many service requests made
EA014	Service Request Validation Failed
EA015	Invalid effective duration
EA017	Service Request Validation Failed
EA018	Content-Type unsupported
EA019	Service Request Validation Failed
EA020	Service Request Validation Failed
EB003	Service Request Validation Failed
EB004, EB005, EB006	Service Request Validation Failed
EB007	Expired pay code
EB008	Service Request Validation Failed
EB009	Transaction not found
EB010	Service Request Validation Failed
EB011	Service Request Validation Failed
EB012	Service Request Validation Failed
EB013	Service Request Validation Failed
EB020	QR code is in Aborted state
EB021	QR code is in Completed state
EB022	QR code payment is currently being processed