



# Pyxpro Payment Integration Guide

## (Business Scan QR)

### 1 Introduction

#### 1.1 Audience

The readers of this document would be technical engineers who will conduct integration work with Pyxpro payment platform, as well as people who want to learn the business process of Pyxpro in-store payment services.

#### 1.2 Change History

Date	Version	Comments
2020-02-10	1.0	First edition

### 2 Message Structure

#### 2.1 Message Sample

All the messages communicated with Pyxpro platform must be sent over HTTPS with POST command and payload of the message is encoded by xml format. Message signature is required to secure the message transmission.

**See example in below:****Request Message**

```
<xml>
<service>unified.trade.micropay</service>
<attach><![CDATA[att]]></attach>
<body><![CDATA[Test]]></body>
<device_info>1000</device_info>
<mch_id>10000100</mch_id>
<nonce_str>adf880d5c8986bd0deb6423c92c9d948</nonce_str>
<out_trade_no>20181112143233</out_trade_no>
<spbill_create_ip>127.0.0.1</spbill_create_ip>
<total_fee>1</total_fee>
<sign><![CDATA[F53145E553092CE52E4CAA4D2B49A91C]]></sign>
</xml>
```

**Response Message, 'status' field indicates the communication result.**

**0 - success, message is received and decoded successfully**

**others - failure**

**'result\_code' field indicates the request processing result.**

**0 - success, request is processed successfully**

**Others - failure**

**Both 'status' and 'result\_code' are 0 if success, see the example in below:**

```
<xml>
<status>0</status>
<message><![CDATA[OK]]></message>
<appid><![CDATA[wX2421b1c4370ec43b]]></appid>
```



```
<mch_id><![CDATA[10000100]]></mch_id>
<device_info><![CDATA[1000]]></device_info>
<nonce_str><![CDATA[FvYsnPuFFPkAr77M]]></nonce_str>
<sign><![CDATA[63238039D6E43634297CF2A6EB5F3B72]]></sign>
<result_code>0</result_code>
<openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>
<is_subscribe><![CDATA[Y]]></is_subscribe>
<trade_type><![CDATA[MICROPAY]]></trade_type>
<bank_type><![CDATA[CCB_CREDIT]]></bank_type>
<total_fee>1</total_fee>
<coupon_fee>0</coupon_fee>
<fee_type><![CDATA[CNY]]></fee_type>
<transaction_id><![CDATA[1008450740201407220000058756]]></transaction_id>
<out_trade_no><![CDATA[1406033828]]></out_trade_no>
<attach><![CDATA[att]]></attach>
<time_end><![CDATA[20140722160655]]></time_end>
</xml>
```

**Common error case: 'status' is not 0**

**Communication error, find details in 'message' field**

```
<xml>
  <version>2.0</version>
  <charset>UTF-8</charset>
  <status>400</status>
  <message>Signature error</message>
</xml>
```

**Request error case: 'status' is 0 but 'result\_code' is not 0**

**Business processing error, find details in 'err\_code\_des' field**



```
<xml> <status>0</status>

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

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

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

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

<nonce_str><![CDATA[sthBJ9QyUG6vkrjJ]]></nonce_str>

<sign><![CDATA[6277A96D7875D4FF23AA7B6A4C3046AB]]></sign>

<result_code>1</result_code>

<err_code><![CDATA[AUTHCODE_EXPIRE]]></err_code>

<err_code_des><![CDATA[QR Code is expired, try new QR]]></err_code_des>

</xml>
```

## 2.2 Test Information

**Server URL:**

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

**Test merchant account (GBP):**

124570000012

**Signature key:**

853071d096e17d30212bff5d92de23e9

## 3 Payment Interface Description

### 3.1 Merchant Scan QR

#### 3.1.1 Scenario

**Step 1:** Consumer opens 'QR Code' on Wechat/Alipay wallet, see figure 1& 2 below:



Figure 1



Figure 2

**Step 2:** Cashier creates a payment order on till and confirm the payment amount with consumer.

**Step 3:** Cashier scans consumer's payment QR code to initiate the payment request.

**Step 4:** Consumer receives authorization request on Wechat/Alipay to confirm the transaction, see figures below:



Figure 3

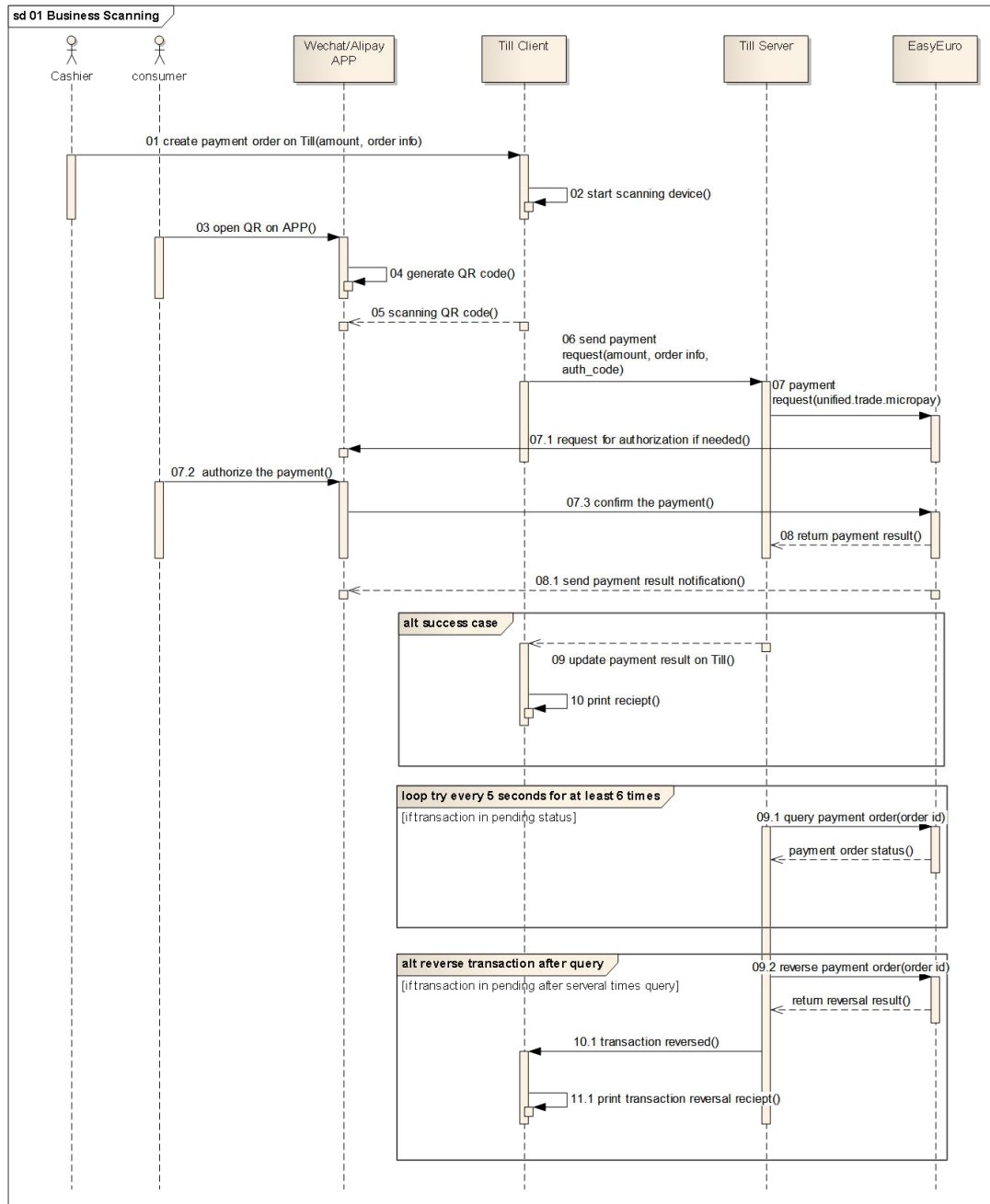


Figure 4

**Step 5:** Cashier reviews payment result on till and gives goods to consumer.



### 3.1.2 System interaction flow



**Notes:**

- ✓ Till client & Till Server are exiting payment collecting system
- ✓ In step 07, Till server starts invoking Pyxpro payment interface after read payment QR code (auth\_code) by scanning device
- ✓ In step 07.1-07.3, additional authorization from consumer may be required:
  - Case 1: no authorization for small amount payment  
Pyxpro returns payment result to Till system directly.
  - Case 2: consumer authorization required  
Transaction is in waiting authorization status
- ✓ In step 09.1, Till server needs to query order status if Pyxpro returns uncertain status of the payment order.
- ✓ In step 09.2, Till server should reverse payment order if fails on several times queries.

**3.1.3 Create Payment Order Interface**

Parameter	Is Mandatory	Data Type	Description
service	Y	String(32)	Service type: unified.trade.micropay
version	N	String(8)	Interface version, default is 2.0
charset	N	String(8)	Message encoding charset, fixed value: UTF-8
sign_type	N	String(8)	Signature algorithm fixed value: MD5
mch_id	Y	String(32)	Unique merchant id allocated by Pyxpro
out_trade_no	Y	String(32)	Unique payment order id generated by merchant system. Maximum is 32 characters, only alphabet, digit and underscore and mixture.
device_info	N	String(32)	Terminal number
body	Y	String(127)	Product/Service purchased
attach	N	String(127)	Extension message for merchant system
total_fee	Y	Int	Value of payment amount Value unit is 1% of the currency. For example, If payment amount is HKD 10.00, the value of total_fee should be 1000



mch_create_ip	Y	String(16)	IP address of the merchant system (client)
auth_code	Y	String(128)	authorization data captured by scanner from consumer's payment QR code/bar code
time_start	N	String(14)	Time to initial payment order, format: yyyyMMddHHmmss, e.g. 25 <sup>th</sup> Sept, 2009, 9:10:10 value: 20091225091010 Time Zone: GMT+8 beijing
time_expire	N	String(14)	Expiry of payment order Same format as time_start
op_shop_id	N	String(32)	Store number
op_user_id	N	String(32)	Cashier id
op_device_id	N	String(32)	Device number
goods_tag	N	String(32)	Promotion flag supported by Wechat
nonce_str	Y	String(32)	Random string less than 32 characters
sign	Y	String(32)	Message signature with MD5

### Response Message

Parameter	Is Mandatory	Data Type	Description
version	Y	String(8)	Default 2.0.
charset	Y	String(8)	Fixed :UTF-8
sign_type	Y	String(8)	Fixed:MD5
status	Y	String(16)	0 - success other - failure this field indicates communication status of the request. Only 'result_code' 0 indicates request is successful.
message	N	String(128)	Error description of communication failure
Only return when 'status' is 0			





result_code	Y	String(16)	Request status 0 - success other - failure
mch_id	Y	String(32)	Unique merchant id allocated by Pyxpro
device_info	N	String(32)	Terminal number
nonce_str	Y	String(32)	Random string, less than 32 characters
err_code	N	String(32)	Error code
err_msg	N	String (128)	Error description
sign	Y	String(32)	Message signature
need_query	N	String(1)	Query tag Y - need query N - no further query
Only return when both 'status' and 'result_code' are 0			
openid	N	String(128)	Unique id of Wechat user
sub_openid	N	String(128)	Unique id of Wechat user associated with sub merchant
trade_type	Y	String(32)	Wechat: pay.weixin.proxy.micropay.intl Alipay: pay.alipay.micropay.intl
is_subscribe	N	String(1)	Y - subscribed N - not subscribed
pay_result	Y	Int	0 - success Other - failure
pay_info	N	String(64)	Error message in failure case
transaction_id	Y	String(32)	Unique order id generated by Pyxpro
out_transaction_id	Y	String(32)	Unique order id generated by Wechat/Alipay
sub_is_subscribe	N	String(1)	Y - subscribed N - not subscribed



sub_appid	N	String	Appid of the merchant
out_trade_no	Y	String(32)	Unique order id generated by merchant system
total_fee	Y	Int	Amount of payment order
coupon_fee	N	Int	Consumer paid amount = total_fee - coupon fee, coupon fee<= total_fee
fee_type	Y	String(8)	Currency code of merchant collecting payment, complaint to ISO4217
attach	N	String(127)	Echo of the extension message added by merchant
bank_type	Y	String(16)	Bank name
bank_billno	N	String(32)	Bank order number
time_end	Y	String(14)	Transaction completed time, format: yyyyMMddHHmmss, time zone: GMT+8 beijing.
cash_fee	Y	Int	Amount paid by cash wallet, unit 1% of CNY
cash_fee_type	N	String(16)	Currency code of cash wallet, fixed CNY
rate	Y	String(16)	FX rate between CNY and merchant collecting currency

Notes:

1. If response message returns and 'status' and 'result\_code' are 0, query order interface should be invoked in case 'need\_query' is empty or 'Y' in response message.
2. Query order interface will be invoked in every 5 seconds and retry times will not be less than 6 if 'trade\_state' is not 'SUCCESS'.
3. Reverse order interface will be invoked if query order request failed on retry policy.

### 3.1.5 Query Payment Order

#### Introduction

Merchant system can query order details by out\_trade\_no or Pyxpro transaction id.

Request message is sent over HTTPS POST command.



## Request Message

Parameter	Is Mandatory	Data Type	Description
service	Y	String(32)	Service Type unified.trade.query
version	N	String(8)	Fixed: 2.0
charset	N	String(8)	Fixed:UTF-8
sign_type	N	String(8)	Fixed: MD5
mch_id	Y	String(32)	Unique merchant id allocated by Pyxpro
out_trade_no	N	String(32)	Unique order id generated by merchant system
transaction_id	N	String(32)	Unique order id generated by Pyxpro. Either 'out_trade_no' or 'transaction_id' should be filled
nonce_str	Y	String(32)	Random string, less than 32 characters
sign	Y	String(32)	Message Signature in MD5

## Response Message

Parameter	Is Mandatory	Data Type	Description
version	Y	String(8)	Interface version Fixed:2.0.
charset	Y	String(8)	Fixed:UTF-8.
sign_type	Y	String(8)	Fixed:MD5
status	Y	String(16)	Request message communication status. 0 - success other - failure
message	N	String(128)	Error message
Only return when 'status' is 0			
result_code	Y	String(16)	Request message processing status 0 - success other - failure



mch_id	Y	String(32)	Unique merchant id allocated by Pyxpro
device_info	N	String(32)	Terminal Number
nonce_str	Y	String(32)	Random string, less than 32 characters
err_code	N	String(32)	Error code
err_msg	N	String (128)	Error description
sign	Y	String(32)	Message signature in MD5
Only return when both 'status' and 'result_code' are 0			
trade_state	Y	String(32)	Payment order status: <b>SUCCESS</b> - success <b>REFUND</b> - payment is refunded <b>NOTPAY</b> - order is created but not paid <b>CLOSED</b> - payment order is closed <b>PAYERROR</b> - other errors, e.g. issuer bank reject
Only return when 'trade_state' is SUCCESS			
openid	N	String(128)	Appid of Wechat user
appid	N	String(32)	Payment SP Wechat official id
sub_openid	N	String(128)	Unique appid of Wechat user associated with Merchant
sub_appid	N	String(32)	Merchant Wechat official id
trade_type	Y	String(32)	pay.weixin.native.intl
is_subscribe	N	String(1)	If subscribe Wechat official account Y - subscribed N - no subscribed
sub_is_subscribe	N	String(1)	Y - subscribed N - no subscribed
transaction_id	Y	String(32)	Unique payment order id generated by Pyxpro
out_transaction_id	Y	String(32)	Unique transaction id generated by Wechat/Alipay



out_trade_no	Y	String(32)	Unique payment order id generated by merchant system, only including alphabet, digit, underscore and mixture
total_fee	Y	Int	Total value of the payment Unit is 1% of the currency For example in HKD: 1000 indicates HKD 10.0,
coupon_fee	N	Int	Coupon amount Coupon amount <= total_fee Cash paid = total_fee - coupon amount
fee_type	Y	String(8)	Merchant payment collecting currency. Currency code compliant with ISO 4217 standard
attach	N	String(127)	Echo data from request message filled by merchant system
bank_type	Y	String(16)	Bank name
bank_billno	N	String(32)	bank bill number if available
time_end	Y	String(14)	Completed time of payment transaction Format: yyyyMMddHHmmss, Time Zone: GMT+8 beijing
cash_fee	Y	Int	Amount is paid by cash wallet, unit is 1% of the currency
cash_fee_type	N	String(16)	Currency code, Fixed: CNY
rate	Y	String(16)	FX rate between CNY and merchant payment collecting currency

### 3.1.6 Reverse Order

#### Introduction

Merchant system can reverse the payment order in pending or expiry state.

The result after invoke this interface:



The payment order will be closed if the transaction is not completed in Pyxpro server.

The payment order will be reversed if the transaction is completed in Pyxpro server.

**Notes:**

1. Reverse interface can only be used to handle payment order in unknown status and should be invoked after 15 seconds after the payment order is created.
2. This interface cannot be used to refund the completed transaction in normal case.

**Request Message**

Parameter	Is Mandatory	Data Type	Description
service	Y	String(32)	Service Name: unified.micropay.reverse
version	N	String(8)	Fixed: 2.0
charset	N	String(8)	Fixed: UTF-8
sign_type	N	String(8)	Fixed: MD5
mch_id	Y	String(32)	Unique merchant id allocated by Pyxpro
out_trade_no	Y	String(32)	Unique payment order id generated by merchant system
nonce_str	Y	String(32)	Random string, less than 32 characters
sign	Y	String(32)	Message signature in MD5

**Response Message**

Parameter	Is Mandatory	Data Type	Description
version	Y	String(8)	Fixed: 2.0
charset	Y	String(8)	Fixed: UTF-8
sign_type	Y	String(8)	Fixed: MD5



status	Y	String(16)	Message communication status: 0 - success other - failure
message	N	String(128)	Error message
Only return when 'status' is 0			
result_code	Y	String(16)	Request processing status: 0 - success, payment order in closed status cannot be paid anymore. Other - failure, close payment order failed, you need try again
mch_id	Y	String(32)	Unique merchant id allocated by Pyxpro
nonce_str	Y	String(32)	Random string, less than 32 characters
err_code	N	String(32)	Error code
err_msg	N	String (128)	Error description
sign	Y	String(32)	Message signature in MD5

## 4 Signature Algorithm

### Step 1:

place all data sent or received into dataset M, then sort all non-null parameters in ascending order according to their ASCII codes, then assembling in key-value pair format (i.e. key1=value1&key2=value2...) and set as stringA. Keep key and value in raw value without URL encoding.

### Notes:

1. 'Sign' and null-value parameters should be excluded from signature string
2. Additional parameters may be introduced in later upgrading release, please consider this case in your design

### For example:

stringA="



```
body=iPhone&charset=UTF-8&device_info=SN12345678&mch_create_ip=103.218.216.123&mch_id=157590000059&nonce_str=ibuaiVcKdpRxxhJA&notify_url=http://www.baidu.com&op_user_id=10001&out_trade_no=Q3NL0000442503084402&product_id=123456789&service=pay.weixin.native.intl&sign_type=MD5&time_expire=20180527202310&time_start=20180526202310&total_fee=10&version=2.0 ";
```

**Step 2:**

Append “key=” + <merchant signature key> to stringA and then sign the new string in MD5 algorithm and covert the signed string in uppercase Hex format.

For example:

Merchant signature Key: c43a467098af250171cd787e56d03978,

stringB=stringA + “&key= c43a467098af250171cd787e56d03978”, then you get:

```
“body=iPhone&charset=UTF-8&device_info=SN12345678&mch_create_ip=103.218.216.123&mch_id=157590000059&nonce_str=ibuaiVcKdpRxxhJA&notify_url=http://www.baidu.com&op_user_id=10001&out_trade_no=Q3NL0000442503084402&product_id=123456789&service=pay.weixin.native.intl&sign_type=MD5&time_expire=20180527202310&time_start=20180526202310&total_fee=10&version=2.0&key=c43a467098af250171cd787e56d03978”;
```

Sign in MD5:

```
Final Signature=MD5(stringSignTemp).toUpperCase()="2444D06685977A585F4C98D7C796BD92"
```