

1. ABOUT THIS DOCUMENT

This document is the official manual for WeChat's payment APIs. As such, it is the tutorial for technical architects, R&D engineers, testing engineers and service engineers for the integrated WeChat payment system, which includes vendors' systems such as online shopping platforms, cashier systems, or automatic smart POS systems. The following topics are designed to be read in order and there are references to topics "previously seen" and topics "yet to come". These references are linked accordingly and it should generally not be a problem to read ahead on other topics.

2. TERMS USED IN THIS DOCUMENT

1. **Payment Method**

1) Quick Pay

The Payer shows their bar code or QR Code on WeChat's Quick Pay page to the Vendor to scan in order to pay directly. This mode applies to offline payment scenarios.

2) QR Code Payment

The Vendor generates a transaction QR Code according to the WeChat Payment Protocol and the Payer goes to "Scan QR Code" in their WeChat in order to complete payment. This mode is applicable to payments made on websites, physical stores, media advertising, or other scenarios.

3) In-App Web-based Payment

The Payer opens the Vendor's HTML5 pages on their WeChat and calls the WeChat payment module via the JSAPI interface to pay their transaction. This mode is applicable in the following scenarios:

- The Payer enters the Vendor's official account and completes their payment on the transaction page;
- The Payer's friend shares the Vendor's payment URL in a chat or in Moments and the payer clicks the link to complete their payment;
- The Payer scans the payment QR Code displayed within the Vendor's page and opens it in a browser to complete their payment.

4) In-App Payment

In-App payment also refers to a mobile-based payment in which the Vendor calls the WeChat payment module by using the open SDK integrated in their mobile-based app to pay for transactions.

2. **Definitions**

1) WeChat Official Account Admin Platform

The WeChat Official Account Admin Platform serves as the application entry and management platform for official accounts. Using this platform, vendors can submit their basic information, business data and financial information for enabling WeChat payment.

URL: <http://mp.weixin.qq.com>。

2) WeChat Open Platform

The WeChat Open Platform serves as the entry point for a vendors' app to access to the WeChat payment open API. Using this platform, vendors can apply for WeChat in-app payment.

URL: <http://open.weixin.qq.com>。

3) WeChat Vendor Platform

The WeChat Vendor Platform serves as the functional hub for vendor features related to WeChat payment, including parameter settings, payment data query and statistics, online refunds, mobile coupon management, and other features.

URL: <http://pay.weixin.qq.com>。

4) WeChat Payment System

The WeChat Payment System is the generic term for the backend services processing system for APIs, account system, and the callback notification system for the WeChat payment process.

5) Vendor Point of Sale Terminal

The Vendor Point of Sale Terminal refers to the POS system commonly used by a cashier that helps record product data, create orders, assist the Payer's payment and print the transaction bill. When integrating with WeChat payment, this system requires the development and testing of a POS system.

6) Vendor Backend System

The Vendor Backend System is the generic term for the Vendor's backend services processing system, and includes the Vendor's website, checkout system, purchase-sale-stock system, delivery system, and customer service system.

7) Scanner

The Scanner is used to help the Vendor's system to quickly read coding data within an image. Based on the type of image coding, the vendor can use a QR Code scanner or a bar code scanner. In terms of scanner types, there are infrared scanners and laser scanners.

8) Vendor Certificate

The Vendor Certificate is a binary file provided by WeChat, which is used as a certificate to identify the Vendor's identity when the Vendor's system initiates a request session with WeChat's backend payment server.

9) Signature

The Vendor's backend and the WeChat payment system create the same signature based on the same secret key and algorithm and use it to verify each other's identity. The

signature algorithm is created and provided by WeChat. Commonly used signature modes are MD5, SHA1, SHA256, and HMAC.

10) JSAPI-based Web Payment

JSAPI-based Web Payment also refers to the above-mentioned In-App web-based payment. Using this method, the Payer can click a URL in an official account, Moments or a chat, and scan a QR Code using WeChat to open the Vendor's HTML5 page to complete their payment.

11) Native Payment

Native Payment refers to Quick Pay on WeChat. Using this method, the Vendor creates a QR Code using the WeChat payment protocol. The Payer then scans the QR code to confirm the transaction, and enters their payment password to complete the payment.

12) Payment Password

A Payment Password is set independently by the Payer when enabling WeChat payment, and is used to confirm their payment and authorize transactions. This password is different from their WeChat password used to log in to WeChat.

13) OpenID

OpenID is used to share a user's identity to an official account, and is different between official accounts. The Vendor's backend obtains the Payer's OpenID during login authorization, payment notifications, and when calling the Query Order API. With OpenID, the system can check whether the payment-related operations are done by the same payer and send service feeds and templated messages to the Payer.

3. PAYMENT ACCOUNT

Vendors can apply for a payment method on the WeChat Official Account Admin Platform (for Native Quick Pay and In-App Web-based Payment) or the WeChat Open Platform (for In-App Payment) as instructed. After the WeChat payment staff receives and reviews the application, the corresponding payment permission will be opened for the Vendor. The Vendor will then receive an email containing required payment instructions from the WeChat Payment Assistant, as shown in Figure 3.1.

尊敬的【 **测试商户** 】：

恭喜您已成功通过微信支付商户资料审核！以下为您的账户所需要的重要信息，请注意保密：

微信支付商户号	
商户平台登录帐号	
商户平台登录密码	
申请微信支付公众号	
APPID	

[前往微信支付商户平台](#)

后续操作指引

前往公众平台完成入驻

您还需要登录微信公众平台，继续按操作指引完成微信支付入驻，请点击【公众平台-微信支付-支付申请-②签署承诺函】签署承诺函，并点击【公众平台-微信支付-支付申请-②签署协议】在线签署微信支付服务协议
[点击查看示意图](#)

根据开发文档进行开发

现在您已具备微信支付功能开发的资格，请您下载最新的文档进行开发

下载API证书、设置API密钥

开发所需的接口API证书、密钥，请您登录微信支付商户平台，在【账户设置-密码安全-API安全】中下载以及设置，如果你此前已经申请过微信支付，请您注意，新接口的API证书和密钥不会再随邮件下发

登录微信支付商户平台

商户平台是微信支付提供给您的商户管理软件系统，包括交易管理、结算管理、营销推广等。为了保证订单数据准确性以及退款的唯一性，请您只通过微信支付商户平台进行订单查询和退款操作

Figure 3.1 Email Template for Application Approval

Table 3.1 shows the relation between email parameters and API parameters.

Table 3.1 Account parameters

Parameters in Email	API Parameter Name	Description
APPID	appid	appid is a unique identity key for each app within the WeChat Official Account Admin Platform or WeChat Open Platform, and is assigned by WeChat after developers apply for it on these platforms. The application approval email also contains this field.
Vendor ID for WeChat Payment	mch_id	Specifies vendor's receipt ID assigned by WeChat Payment after they have applied for WeChat Payment
API Key	key	This key is created for transaction signatures and is retained in the Vendor's backend and WeChat payment system, and should not be made available publicly or on the Internet. The Vendor should keep this key secured and avoid disclosing it to others. Vendors may configure the key according to the email instructions.
Appsecret	secret	AppSecret is the API password corresponding to APPID, and is used to obtain a certificate (access_token) for calling API access_token. Using WeChat payment, you should obtain an OpenID via the OAuth2.0 interface and use it within the single interface for In-App Web-based payment. Developers shall be qualified to get AppSecret in development mode.

4. API RULES

1. Protocol Rules

The following specifies the rules for calling the API when a vendor accesses WeChat payment:

Table 4.1 API Rules

Transfer Mode	Use HTTPS for secure transactions
Submit Mode	Use POST method
Data Format	Data submitted and returned is in XML format
Char Encoding	Use UTF-8 character encoding
Signature Algorithm	MD5
Signature Requirement	Signature-checking is required for requesting and receiving data. For more information, see Section 4.3.1 Signature Algorithm .
Certificate Requirement	A vendor certificate is required for calling the Submit Refund API or Revoke Order API.

Logic Judgment	Determine protocol field, service field and transaction status.
----------------	---

2. Parameter Specifications

1) Payment Amount

The currency type for transaction is CNY (Chinese yuan) by default. The unit used in payment amount is 【cent】 and must be an integer. However, the unit 【yuan】 is used in the transaction amount when downloading transaction history.

For foreign transactions, the transaction amount will use the smallest unit of currency, but the reference value must be an integer without decimals. For example, using a currency type of "United States dollars" a reference value of "1750" in the payment amount would be equivalent to US\$17.50.

2) Currency Type

The list of currency types is as below:

GBP: Great Britain pound

HKD: Hong Kong dollar

USD: United States dollar

JPY: Japanese yen

CAD: Canadian dollar

AUD: Australian dollar

EUR: Euro cent

NZD: New Zealand Dollar

KRW: South Korean won

Notes: The currency type for payment and refund must be identical.

3) Time Protocol

China Standard Time (UTC+08) is used in this document. If the Vendor is not in this time zone, they should convert their current time into China Standard Time. For instance, if a vendor is in London at the local time of November 11, 2014 0:00, it will be November 11, 2014 8:00 AM in Beijing.

4) Timestamp

Taking China Standard Time (UTC+08) as the standard time zone, a timestamp is calculated as seconds since 1970/01/01 00:00:00 UTC and is the required method to store timestamps in this document. Note: Milliseconds should be rounded to seconds (10-digits).

5) Vendor's Order Number

The order number for a payment is defined by the vendor and must be unique. We suggest adding a random sequence to the end of the current time in order to create a unique

and sequenced number. The original order number is used when initiating a payment again in order to avoid duplicate transactions. However, orders that are paid, closed or revoked (for more information, see Section 9 “Public API”) cannot be paid again.

3. Security Specifications

1 Signature Algorithm

General steps to create a signature:

Step 1: Presume all data sent and received is the set M. Sort non-empty values in M in ascending alphabetical order (i.e. lexicographical sequence), and join them into string A via the corresponding URL key-value format (e.g. key1=value1&key2=value2...).

Notes:

- Sort parameter names in ascending alphabetical order based on their ASCII encoded names (e.g. lexicographical sequence);
- Empty parameter values are excluded in the signature;
- Parameter names are case-sensitive;
- When checking returned data or a WeChat push notification signature, the transferred sign parameter is excluded in this signature as it is compared with the created signature.

Step 2: Add "key= (API key value) to the end of stringA to get stringSignTemp, perform MD5 arithmetic on stringSignTemp, convert all result chars to upper case, thus get sign's value (signValue).

Example:

For the following transferred parameters:

appid: **wxd930ea5d5a258f4f**

mch_id: **10000100**

device_info: **1000**

body: **test**

nonce_str: **ibuaiVcKdpRxkhJA**

(1) Sort ASCII code of parameter names by lexicographical sequence based on the format of "key=value"

*stringA="appid=wxd930ea5d5a258f4f&body=test&device_info=1000&mch_id=10000100
&nonce_str=ibuaiVcKdpRxkhJA";*

(2) Join API keys

stringSignTemp="stringA&key=192006250b4c09247ec02edce69f6a2d"

sign=MD5(stringSignTemp).toUpperCase()="9A0A8659F005D6984697E2CA0A9CF3B7"

Obtain data to be transferred below:

```
<xml>
  <appid>wx930ea5d5a258f4f</appid>
  <mch_id>10000100</mch_id>
  <device_info>1000</device_info>
  <body>test</body>
  <nonce_str>ibuaiVcKdpRxxhJA</nonce_str>
  <sign>9A0A8659F005D6984697E2CA0A9CF3B7</sign>
</xml>
```

WeChat provides online signature tools for this API: URL1.

2 Random String Algorithm

nonce_str is included in WeChat payment API protocols to ensure unpredictability for signatures. We suggest calling the random() function to create a signature and convert its value into a string.

3 Vendor Certificate

1) Obtain Vendor Certificate

APIs related to payment rollbacks (such as refunds or revoked orders) require a vendor's certificate. The certificate is issued to vendors via an email notification after the vendor applies for WeChat payment successfully. There are four certificates that might be required as indicated below:

Table 4.2: Certificate Description

Certificate Attachment	Description	Use Case	Remarks
pkcs12 format (apiclient_cert.p12)	Includes certificate for private key information, in p12(pfx) format and issued by WeChat payment for identity verification	Calling the Revoke Order API and Submit Refund API	Double-click to import into a Windows system and enter certificate password as prompted. By default, the certificate password is the vendor's ID (e.g. 10010000)
pem format for certificate (apiclient_cert.pem)	apiclient_cert.p12 certificate files may be imported to create a certificate in pem format. Do not disclose to others.	pem format should be used for PHP applications as PHP can't use the p12 format	You can also use the "openssl" command to import the p12-format certificate as below: openssl pkcs12 -clcerts -nokeys -in apiclient_cert.p12 -out apiclient_cert.pem
pem format for certificate secret key (apiclient_key.pem)	apiclient_cert.p12 certificate files may be imported to create a certificate in pem format.	pem format should be used for PHP applications as PHP can't use the p12 format	You can also use the "openssl" command to import the p12-format certificate as below: openssl pkcs12 -nocerts -in apiclient_cert.p12 -out apiclient_key.pem
CA certificate (rootca.pem)	WeChat payment API server also deploys server certificates to verify identity for WeChat payment. When vendors	This file is the root certificate issued by authorities that sign WeChat payment certificates, which can	Root certificates are built-in to some tools. For tools without root certificates, the ones provided here may be used.

	call APIs, the authenticity of the server called and domain name shall be verified.	be used to verify the authenticity of WeChat payment server certificates.	
--	---	---	--

2) Use Vendor Certificate

- ◆ apiclient_cert.p12 is vendor's certificate files for all R&D operations except PHP-based development.
- ◆ Vendors using a .NET environment should ensure that their framework version is greater than 2.0. They can double-click to install the certificate "apiclient_cert.p12" before using.
- ◆ The default password for the calling vendor's certificate and installation is vendor's ID (mch_id).
- ◆ "apiclient_cert.pem" and "apiclient_key.pem" are required for PHP-based development, and rootca.pem is CA certificate.

For more invocation examples, see [Demo outbound links provided by WeChat payment.](#)

3) Vendor Certificate Security

Certificate files should not be stored in a virtual directory on the web server. Instead, they should be placed in a directory with strict access control in order avoid the certificate being downloaded by others. The Vendor's server should also be free from viruses and trojan horses to avoid potential certificate theft.

4 Vendor's Callback API Security

In many network environments, HTTP requests bear the risk of DNS spoofing, unwanted pop-ups, and data theft and modifications. The Vendor's callback API should use HTTPS to ensure data transfer security. For this reason, we suggest all vendors use HTTPS for all WeChat payment callbacks. For more information, see the [HTTPS Building Guide.](#)

5. QUICK PAY PROGRAMMING GUIDE

1. Use Case

Step 1: After logging in to WeChat, the Payer enters "Quick Pay" in "Me"->"Wallet", as shown in Figure 5.1;

Step 2: Cashier creates a transaction order and the Payer confirms the payment amount displayed on the point of sale terminal;

Step 3: Cashier scans barcode or QR code shown by the Payer into the point of sale terminal and the transaction order is submitted to the transaction system on a WeChat payment server;

Step 4: After the payment request is received by the transaction system, the transaction system determines whether the Payers' payment password must be verified. If the payment password is not required, the payment is made directly. Otherwise, the Payer is prompted to enter their password, as shown in Figure 5.2. If the payment is successful, the Payers will see

a 'successful payment' message in WeChat, as shown in Figure 5.3; and if the payment fails, a payment error page will be displayed instead.



Figure 5.1 Quick Pay



Figure 5.2 Confirm Payment



Figure 5.3 Successful Payment

2. Payment Verification Code Rules

- The Payer's payment password must be verified for any transactions totaling more than 1000CNY;
- For transactions less than 1000 CNY, allow up to 5 password-exempt transactions per day for each WeChat account and require password verification after reaching this limit;
- Require password verification for any untrusted or suspicious transactions;

3. Participating Vendors

Users can experience this payment method for themselves at stores and shops that support WeChat Payment.

Convenience stores: 7-Eleven, Guoda36524, Hi-24, etc.

Chain drugstores: LBX Pharmacy, GuoDa Drugstore, Nepstar Drugstore, etc.

Department stores: Rainbow, etc.

4. Process for Vendors

Based on the Vendor's environment, a vendor may process a payment via backend access or via physical store access. Payment scenarios include payments requiring password-verification and payments that are exempt from password-verification.

1 Access Mode - Vendors' Backend Access

This mode is applied to vendors who are equipped with a unified backend. In this mode, the Cashier communicates with the vendor's backend first and the backend will subsequently send transaction requests to and receive results from the WeChat payment system.

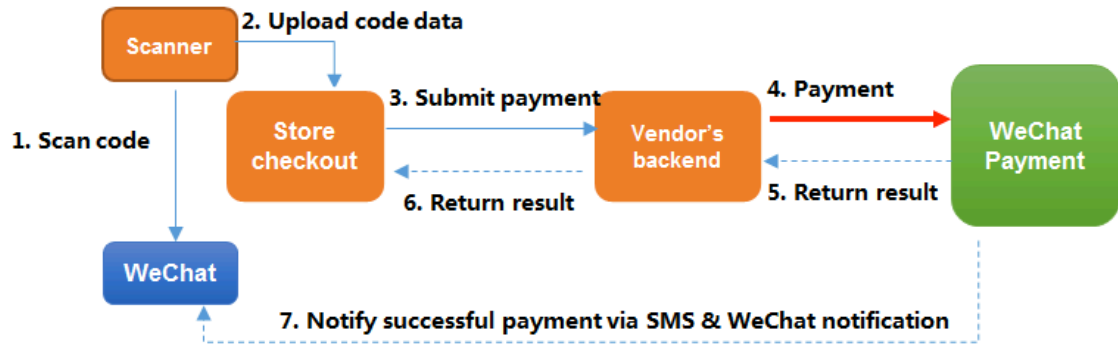


Figure 5.4 Vendors' Backend Access Process

2 Access Mode -- Physical Store Access

This mode is applied to vendors who communicate with the WeChat payment system via a public network. In this mode, the Cashier initiates a transaction request and handles returned results directly with the WeChat payment system. However, the Vendor can process other transactions between their physical stores and backend based on their requirements.

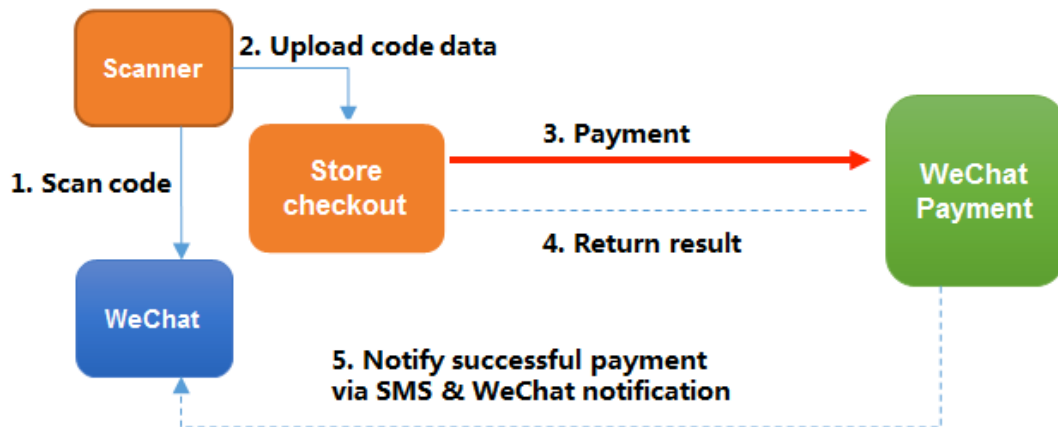


Figure 5.5 Physical Store Access Process

3 Password-Exempt Payment Process

This section uses the Vendor's backend access mode to illustrate the payment process, as shown in the sequence chart below.

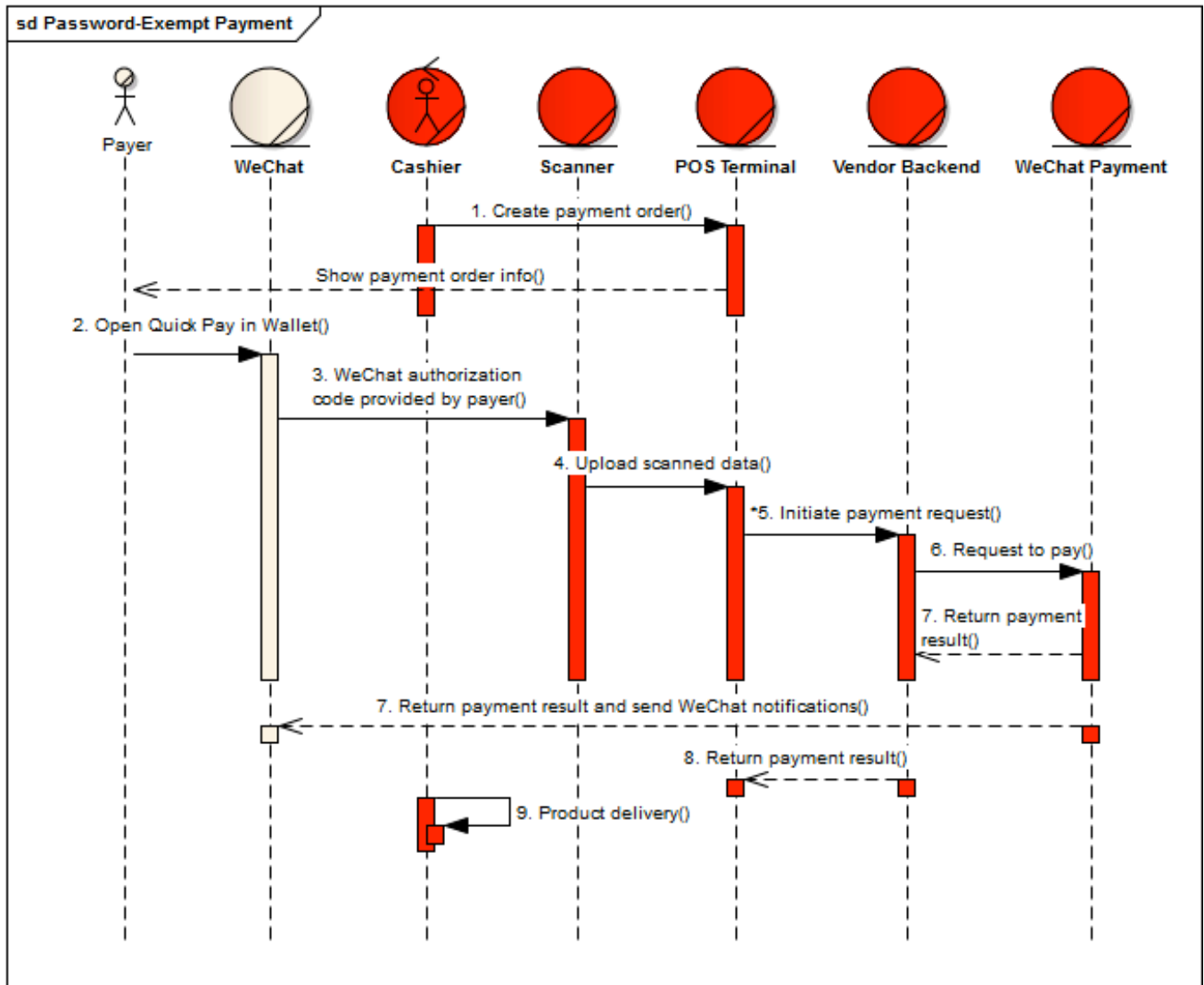


Figure 5.6 Sequence Chart of Password-Exempt Payment

Password-exempt payment steps:

- 1) The Cashier creates a payment order on their point of sale terminal and shows the payment amount to the Payer;
- 2) Payer opens WeChat and enters “Quick Pay” in “Me”->“Wallet”;
- 3) Cashier scans the bar code on the “Quick Pay” screen via their scanner;
- 4) Scanner reads and transfers the code data to the Cashier’s point of sale terminal;
- 5) The Cashier initiates payment request to Vendor’s backend after receiving payment information.
- 6) The Vendor’s backend handles the payment request sent by the physical store point of sale terminal and creates a signature for it, and then calls the **【Submit Quick Pay API】** to start a payment request to the WeChat payment system;

- 7) The WeChat payment system receives the Vendor's payment request, processes the data after validation, and returns a payment result to the Vendor's backend. If the transaction is successfully paid, the WeChat payment system sends the payment result to the Vendor and payer at the same time via SMS or via a WeChat message;
- 8) The Vendor's backend validates the signature to process relevant data, and sends the payment result to the Cashier's point of sale terminal.
- 9) The Cashier delivers goods to the Payer after receiving a successful payment result.

4 Password-Verification Payment Process

The password-verification payment process is quite similar to that of password-exempt payments. The first 5 steps are identical. During the password-verification payment process, the WeChat backend prompts the Payer for their payment password after the Vendor's backend calls the 【Submit Quick Pay API】 to initiate a payment request. After the Payer has successfully verified their payment password, the API returns a USERPAYING status immediately to the Vendor's backend, and the Vendor's backend communicates to the WeChat payment system Query Order API in order to confirm whether the order was successfully paid for.

The password-verification payment process is shown in the following sequence chart:

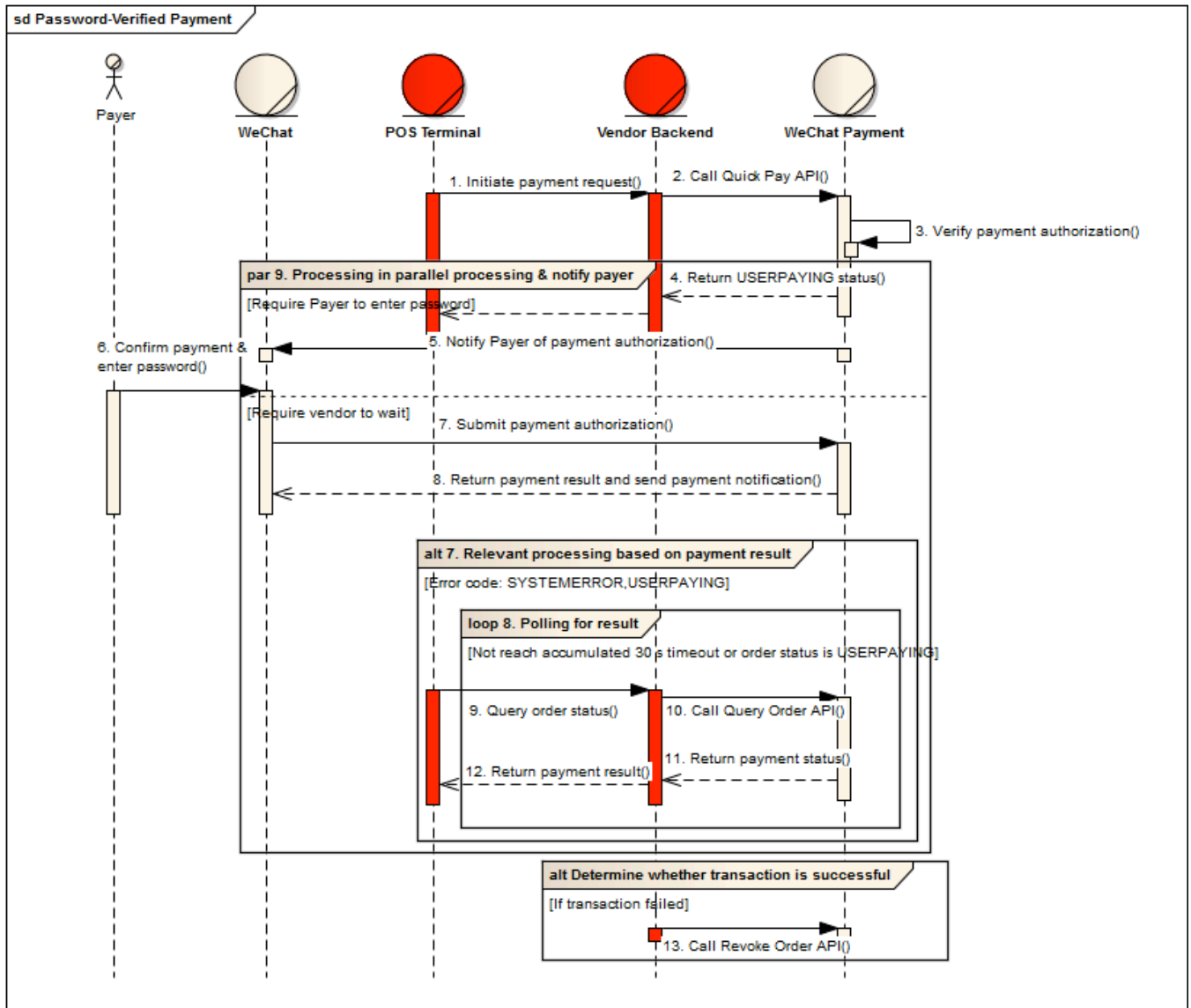


Figure 5.7 Process Diagram for Password-Verified Payment

Here we show only the steps that are different from those previously described.

- 1) The Cashier's point of sale terminal initiates a payment request to the Vendor's backend after creating the order;
- 2) The Vendor's backend calls the **【Submit Quick Pay API】** to create a payment transaction;
- 3) The WeChat payment system verifies the Vendors' request and determines whether password verification is required;
- 4) The WeChat payment system returns a USERPAYING status and the Vendor's backend sends a response containing a result to the Cashier's point of sale terminal;
- 5) WeChat payment system prompts the Payer to enter their payment password on their WeChat;

- 6) The Payer confirms the payment and enters their payment password;
- 7) The Payer submits verification after entering their password;
- 8) The WeChat payment backend returns a payment result after the Payer completes the transaction on WeChat and notifies the Payer of the result via SMS and a WeChat message;
- 9) The Vendor's backend receives USERPAYING status from the WeChat payment system and queries the actual payment result via the [【Query Order API】](#) (For more information, see Public API);
- 10) If the WeChat payment system replies with a payment status of USERPAYING, the Vendor's backend calls the [【Query Order API】](#) every 5 seconds to determine the final actual payment status. If the Payer cancels their payment or delays their payment for more than 30 seconds, the Vendors' backend halts the polling process and calls the [【Revoke Order API】](#) to revoke the transaction.

5 Exception Handling

Follow the instructions below to troubleshoot any payment exceptions:

- 1) If the Payer cannot find their order in their transaction history when they're prompted of a payment error on WeChat, they will require the Vendor to initiate the order again; if the order is successfully paid for, the Vendor's backend calls the [【Query Order API】](#) to query the actual payment status again;
- 2) The Payer will need to repeat the payment process if they're prompted of a payment error caused by insufficient balance, an invalid card or for other reasons;
- 3) The Vendor's backend should call the [【Revoke Order API】](#) (For more information, see Public API) to cancel a transaction whenever the transaction times out or the payment fails;
- 4) The Vendor's backend should send a payment error message to the Cashier's point of sale terminal when it is caused by an abnormal bank system error, insufficient balance, unsupported bank, or for any other reason;
- 5) Depending on the type of error code returned, the transaction may be canceled. For details, see API Return Error Code List.

6. PUBLIC API

The Public API serves as the common API interface to be called to access a variety of supplemental functions. Vendors can choose to integrate these functions based on their needs. The Public API includes the Unified Order, General Notification, Query Order, Close Order, Submit Refund, Refund Query, Download Transaction History and Short URL Conversion functions, which are elaborated in the following subsections.

1. Submit Quick Pay API

1 Use Case

After the Cashier scans a bar or QR code on the Quick Pay page shown by the Payer, the payment parameters are transferred to the Vendor's backend. The Vendor's backend calls the Submit Quick Pay API to initiate a payment.

2 URL

<https://api.mch.weixin.qq.com/pay/micropay>

3 Certificate Requirements

No certificate is required.

4 Parameter Settings

Name	ID	Type	Required	Example	Description
Official Account ID	appid	String(32)	Yes	wx8888888888888888	Specifies an Official Account ID assigned by WeChat
Vendor ID	mch_id	String(32)	Yes	1900000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	String(32)	No	013467007045764	Specifies a Terminal device ID (such as store number as defined by the vendor)
Random string	nonce_str	String(32)	Yes	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm .
Signature	sign	String(32)	Yes	C380BEC2BFD727A4B6845133519F3AD6	For more information, see Section 4.3.1 Signature Algorithm .
Item Description	body	String(127)	Yes	Pay for QQ Coins	Short description of item(s) to be purchased for the Payer
Item Details	detail	String(8192)	No	iPad Mini in white with 16G memory	Displays detailed item list
Additional Data	attach	String(127)	No	User-Defined Data	Allow vendors an additional field to be returned in the payment notification after submitting a payment to the Query Order API
Vendor Order Number	out_trade_no	String(32)	Yes	1217752501201407033233368018	32 alphanumeric characters or less. For more information, see Section 4.2 Vendor's Order Number .
Bid price	total_fee	Int	Yes	888	Specifies the total order amount. The units are expressed in cents as integers. For more details, see Section 4.2 Payment Amount .

Currency Type	fee_type	String(16)	No	CNY	ISO-4217 standard compliant and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type .
Terminal IP	spbill_create_ip	String(16)	Yes	8.8.8.8	Specifies the machine IP that calls the WeChat Payment API
Transaction Start Time	time_start	String(14)	No	20091225091010	Specifies transaction creation time in the format yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10 (UTC+08). For more information, see Section 4.2 Time Protocol .
Transaction End Time	time_expire	String(14)	No	20091227091010	Specifies the transaction end time in the format yyyyMMddHHmmss, such as 20091227091010 for Dec 27, 2009 09:10:10 (UTC+08). For more information, see Section 4.2 Time Protocol .
Item Label	goods_tag	String(32)	No		Specifies the label of goods, which is a parameter in the coupon feature for businesses. For more information, see Section 10 Mobile coupons .
Authorization Code	auth_code	String(128)	Yes	120061098828009406	Specifies the authorization code for scanning a barcode/QR Code on Quick Pay

Example:

```

<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <attach>Additional Order Description</attach>
  <auth_code>120269300684844649</auth_code>
  <body>Quick Pay Testing</body>
  <device_info>1000</device_info>
  <goods_tag></goods_tag>
  <mch_id>10000100</mch_id>
  <sub_mch_id>10000101</sub_mch_id>
  <nonce_str>8aaee146b1dee7cec9100add9b96cbe2</nonce_str>
  <out_trade_no>1415757673</out_trade_no>
  <spbill_create_ip>14.17.22.52</spbill_create_ip>
  <time_expire></time_expire>
  <total_fee>1</total_fee>
  <sign>C29DB7DB1FD4136B84AE35604756362C</sign>
</xml>

```

Notes: Parameters are escaped in XML files and CDATA tags are used to illustrate that data is not parsed by XML parser.

5 Return Data

Field Name	ID	Required	Type	Example	Description
------------	----	----------	------	---------	-------------

Return Status Code	return_code	Yes	String(16)	SUCCESS	Set to SUCCESS or FAIL Specifies communicating label instead of transaction label. The status of the transaction is determined by the value of the result_code field.
Return Data	return_msg	No	String(128)	Signature Failure	If not empty, this is the error description. If not empty, this is the error description Signature Failure Parameter format checking error

When return_code is SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies an Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	Specifies the ID of the terminal device with from which the Vendor submitted their order
Random String	nonce_str	Yes	String(32)	Fsdfds1235df231asdfg32145gfdse	32 characters or fewer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm
Service Result	result_code	Yes	String(16)	SUCCESS	Set to SUCCESS or FAIL

When return_code and result_code are SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Follows Official Account or not	is_subscribe	Yes	String(1)	Y	For users who pay for transactions related to an official account, the value in this field states whether the user is current following the official account Y: Follows N: Doesn't follow
Transaction Type	trade_type	Yes	String(16)	MICROPAY	Transaction type is MICROPAY
Payment Bank	bank_type	Yes	String(16)	CMC	Strings states bank type
Currency Type	fee_type	No	String(16)	CNY	Complies with ISO 4217 standards and uses CNY for Chinese currency by default. For more

					information, see Section 4.2 Currency Type.
Bid price	total_fee	Yes	Int	100	Specifies the total amount for a transaction. For more information, see Section 4.2. Payment Amount .
Cash Type	cash_fee_type	No	String(16)	HKD	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type .
Cash Payment Amount	cash_fee	Yes	Int		Specifies the total cash payment amount of a transaction. For more information, see Section 4.2 Payment Amount .
Mobile Coupon Amount	coupon_fee	No	Int	100	The amount for mobile coupons will be equal to or less than the total order amount. The cash payment amount is the difference between the total order amount and the mobile coupon amount.
Mobile Coupon Amount type	coupon_fee_type	No	String (16)	HKD	Mobile Coupon Amount type
WeChat Payment Order Number	transaction_id	Yes	String(32)	1217752501201407033233368018	Specifies the WeChat payment order id
Vendor Order Number	out_trade_no	Yes	String(32)	1217752501201407033233368018	Specifies the order number created within the Vendors' system, which is consistent with request.
Vendor's Data Package	attach	No	String(128)	123456	Specifies vendor's data package, which is returned as it is.
Payment End Time	time_end	Yes	String(14)	20141030133525	Specifies transaction creation time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. For more information, see Section 4.2 Time Protocol .

Example:

```

<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <device_info><![CDATA[1000]]></device_info>
  <nonce_str><![CDATA[GOp3TRyMXzbMlkun]]></nonce_str>
  <sign><![CDATA[D6C76CB785F07992CDE05494BB7DF7FD]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>
  <is_subscribe><![CDATA[Y]]></is_subscribe>
  <trade_type><![CDATA[MICROPAY]]></trade_type>
  <bank_type><![CDATA[CCB_DEBIT]]></bank_type>

```

```

<total_fee>1</total_fee>
<coupon_fee>0</coupon_fee>
<fee_type><![CDATA[CNY]]></fee_type>
<transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>
<out_trade_no><![CDATA[1415757673]]></out_trade_no>
<attach><![CDATA[Additional order description]]></attach>
<time_end><![CDATA[20141111170043]]></time_end>
</xml>

```

6 Error Codes

Name	Description	Reason	Solution
SYSTEMERROR	API return error	System timed out	Call the Query Order API immediately to check the current status, and use the returned status to decide the next steps for processing the order.
PARAM_ERROR	Parameter error	Requested parameters are not correct	Debug your program based on data returned by the API
ORDERPAID	Order is paid	Duplicate order number	Confirm whether this order has already been processed. If not, submit it as a new order.
NOAUTH	No permissions for vendors	Vendors hasn't enabled Quick Pay	Apply and receive permission to use Quick Pay first. For applying for this payment method, contact customer service.
AUTHCODEEXPIRE	QR Code expired. Refresh and try again.	Payer's bar code has expired	Cashier should ask payer to refresh the bar code/QR code on WeChat and scan the code again. This error is displayed directly to the cashier.
NOTENOUGH	Insufficient balance	Payer has insufficient balance in their payment account	Cashiers should inform the payer to change their payment bank card and scan it again. Note: In this scenario, the Cashier's point of sale terminal should receive a message saying "Insufficient balance in this card. Try another one" from the Vendor's backend.
NOTSUPPORTCARD	Unsupported card type	The type of card used by the payer for their payment is not supported at the moment	Payer will receive a message in WeChat telling them to select another card type Note: In this scenario, the cashier's point of sale terminal should receive a message saying "Unsupported card type. Try another one or link a new card for payment" from the vendor's backend.
ORDERCLOSED	Order is closed	The order is closed	An exception has occurred with this transaction. Create a new order and redo the payment process.

ORDERREVERSED	Order is cancelled	The order is cancelled	An exception has occurred with this transaction. Create a new order and redo the payment process.
BANKERROR	Bank system exceptions	Bank system timed out	Call the Query Order API immediately and check the current transaction status. Process the next steps based on this status.
USERPAYING	Password is required as payers are making their payment	This transaction requires payment password	Wait for 5 seconds and then call the Query Order API again to check current transaction status. Process the next steps based on this status.
AUTH_CODE_ERROR	Authorization parameter error	Requested parameters are not correct	Each QR code can only be used once. Payer should refresh the QR code and try again.
AUTH_CODE_INVALID	Authorization code checking error	The bar or QR code scanned by cashiers is not the one on the Quick Pay page	Scan bar or QR code on the Quick Pay page
XML_FORMAT_ERROR	Invalid XML format	Invalid XML format	Check whether XML parameters are in the correct format
ORDERREVERSED	Order is cancelled	This order is already canceled	Prompt the payer to pay again when this status occurs.
REQUIRE_POST_METHOD	Use post method	Data not transferred via POST method	Check whether data is submitted via POST method
SIGNERROR	Signature error	Incorrect signature result	Check whether the signature parameter and method comply with signature algorithm requirements
LACK_PARAMS	Missing parameter	Required parameter is missing	Check whether all required parameters are complete
NOT_UTF8	Invalid coding format	Specified coding format is not used	Use NOT_UTF8 encoding format
BUYER_MISMATCH	Incorrect payment account	Only one payer is allowed to pay for one transaction.	Check whether the payer is the same person
APPID_NOT_EXIST	APPID does not exist	No APPID in this parameter	Check whether provided APPID is correct
MCHID_NOT_EXIST	MCHID does not exist	No MCHID in this parameter	Check whether provided MCHID is correct
OUT_TRADE_NO_USED	Duplicate vendor order number	The same transaction can't be submitted repeatedly	Check whether the Vendor's order number has already been submitted or used previously
APPID_MCHID_NOT_MATCH	appid does not match mch_id	appid does not match mch_id	Check whether appid belongs to the associated mch_id

2. Cancel Payment API

1 Use Case

When a payment transaction isn't successfully returned or the payment system times out, this API is called to cancel the transaction. If the Payer fails to pay for the order, the WeChat payment system closes the transaction; if the Payer has successfully paid, then the WeChat payment system refunds the order.

Note: This API can be called to cancel a transaction made within 7 days after an order is created, while the Submit Refund API should be used for transactions paid successfully. After a transaction is submitted, **【Query Order API】** is called. When there is no clear query result, **【Cancel Order API】** is called.

2 URL

<https://api.mch.weixin.qq.com/secapi/pay/reverse>

3 Certificate Requirement

This API requires two-way certificates. For more information, see Section [4.3.3 Vendor Certificate](#).

4 Request Parameters

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String (32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String (32)	1900000109	Specifies vendor ID assigned by WeChat Payment
WeChat Order Number	transaction_id	Yes	String (32)	1217752501201407033233368018	WeChat order number is preferred
Vendor Order Number	out_trade_no	No	String (32)	1217752501201407033233368018	out_trade_no is an internal order number within the Vendor's system. transaction_id will be used over out_trade_no if they are both provided by the vendor.
Random String	nonce_str	Yes	String (32)	5K8264ILT KCH16CQ2 502S18ZNM TM67VS	32 characters or fewer or fewer. For more information, see Section 4.3.2 Random String Algorithm
Signature	sign	Yes	String (32)	C380BEC2 BFD727A4 B68451335 19F3AD6	For more information, see Section 4.3.1 Signature Algorithm .

Example:

```
<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <nonce_str>b7ffb16a7150cf08639db472c5f5bdae</nonce_str>
  <out_trade_no>1415717424</out_trade_no>
  <sign>9B2EA16C05A5CEF8E53B14D53932D012</sign>
</xml>
```

Return Data

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String (16)	SUCCESS	SUCCESS or FAIL
Return Data	return_msg	No	String (128)	Signature Failure	If not empty, the returned info is the error description Signature Failure Parameter format checking error

If return_code is SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String (32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String (32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Random String	nonce_str	Yes	String (32)	5K8264ILT KCH16CQ 2502S18ZN MTM67VS	32 characters or fewer
Signature	sign	Yes	String (32)	C380BEC2 BFD727A4 B68451335 19F3AD6	For more information, see Section 4.3.1 Signature
Service Result	result_code	Yes	String (16)	SUCCESS	SUCCESS or FAIL SUCCESS indicates the order was paid for successfully and cannot be paid for again. If the payment is completed, a refund is initiated. 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;
Recall Requirement	recall	Yes	String (1)	Y	Specifies whether calling the Cancel Order API is required or not, Y means yes while N means no.

Example:

```

<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <nonce_str><![CDATA[o5bAKF3o2ypC8hwa]]></nonce_str>
  <sign><![CDATA[6F5080EDDD196FFCDE53F786BBB93899]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <recall><![CDATA[N]]></recall>
</xml>

```

5 Error Code

Name	Description	Reason	Solution
SYSTEMERROR	API return error	System timed out	Try to call the Query Order API again
INVALID_TRANSACTIONID	Invalid transaction_id	Requested parameters are not correct	Parameter error. Check transaction_id again.
PARAM_ERROR	Parameter error	Requested parameters are not correct	Parameter error. Check parameters again.
REQUIRE_POST_METHOD	Use post method	Data is not transferred by post	Check whether data is submitted via POST method
SIGNERROR	Signature error	Incorrect signature result	Check whether signature parameter and method comply with signature algorithm requirements

3. Query Order

1 Use Case

This API allows inquiry of all payment orders made from WeChat. After receiving a status code using this API, vendors can proceed with the next step in service logic. The following are situations when to use the Query Order API:

1. The Vendor doesn't receive any payment due to an exception in the Vendor's backend, network or server;
2. A system error or unknown transaction status is returned after calling the payment interface;
3. USERPAYING status is returned after calling the Quick Pay API;
4. To confirm payment status before calling the Close Order API or Revoke Order API;

2 URL

<https://api.mch.weixin.qq.com/pay/orderquery>

3 Certificate Requirement

No certificate is required.

4 Request Parameters

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
WeChat Order Number	transaction_id	No	String(32)	013467007045764	WeChat order number is preferred
Vendor Order Number	out_trade_no	No	String(32)	1217752501201407033233368018	Specifies an internal order number created by the Vendor's system. This field is required when transaction_id is not provided.
Random String	nonce_str	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm
Signature	sign	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .

Example:

```
<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <nonce_str>ec2316275641faa3aacf3cc599e8730f</nonce_str>
  <transaction_id>1008450740201411110005820873</transaction_id>
  <sign>FDD167FAA73459FD921B144BAF4F4CA2</sign>
</xml>
```

5 Return Data

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL Specifies communicating label instead of transaction label. The status of a transaction is determined by the value oftrade_state.
Return Data	return_msg	No	String(128)	Signature failure	If not empty, this is the error description Signature failure

					Parameter format checking error
--	--	--	--	--	---------------------------------

If return_code is SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
Service Result	result_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL
Error Code	err_code	No	String(32)	SYSTEMERROR	For more information, see Section 5.5.6 Error Code .
Error Code Description	err_code_des	No	String(128)	System error	Describes result data

If return_code and result_code are SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Device ID	device_info	No	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment
User Tag	openid	Yes	String(128)	wxd930ea5d5a258f4f	Specifies the user id of the Payer provided the WeChat system in OpenID format unique to each appid instance
Follows Official Account or not	is_subscribe	Yes	String(1)	Y	Specifies whether the payer follows the associated official account or not, with Y meaning 'follows' and N meaning 'not follows'.
Transaction Type	trade_type	Yes	String(16)	JSAPI	Set to JSAPI, NATIVE, MICROPAY or APP
Transaction Status	trade_state	Yes	String(32)	SUCCESS	SUCCESS: Payment successful REFUND: Order to be refunded NOTPAY: Order not paid CLOSED: Order closed

					REVOKED: Order revoked USERPAYING: Awaiting user to pay PAYERROR: Payment failed (payment status failed to be returned by bank or other reasons)
Payment Bank	bank_type	Yes	String(16)	CMC	String states bank_type
Bid price	total_fee	Yes	Int	100	Specifies the total amount for a transaction. For more information, see Section 4.2. Payment Amount.
Currency Type	fee_type	Yes	String(8)	HKD	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type.
Cash Payment Amount	cash_fee	Yes	Int	100	Specifies the total cash payment amount of a transaction. For more information, see Section 4.2 Payment Amount.
Cash Type	cash_fee_type	No	String(16)	HKD	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type.
Mobile Coupon Amount	coupon_fee	No	Int	100	The amount for mobile coupons will be equal to or less than the total order amount. The cash payment amount is the difference between the total order amount and the mobile coupon amount. For more information, see Section 4.2 Payment Amount.
Mobile Coupon Amount type	coupon_fee_type	No	String (16)	HKD	Mobile Coupon Amount type
Mobile Coupon Quantity	coupon_count	No	Int	1	Specifies the quantity of mobile coupons
Mobile Coupon Batch ID	coupon_batch_id_\$n	No	String(20)	100	Specifies mobile coupon batch ID, where \$n is a suffixed counter ascending from 1
Mobile Coupon ID	coupon_id_\$n	No	String(20)	10000	Specifies mobile coupon ID, where \$n is a suffixed counter ascending from 1
Payment Amount for Each Coupon	coupon_fee_\$n	No	Int	100	Specifies payment amount for each mobile coupon, where \$n is a suffixed counter ascending from 1
Specifies the number of a WeChat	transaction_id	Yes	String(32)	12177525012 01407033233 368018	Specifies the WeChat payment order id number

payment order					
Vendor Order Number	out_trade_no	Yes	String(32)	12177525012 01407033233 368018	Specifies an order number created by a Vendors' system, which is consistent with request.
Vendor's Data Package	attach	No	String(128)	123456	Specifies vendor's data package, which is returned as it is.
Payment End Time	time_end	Yes	String(14)	20141030133 525	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. For more information, see Section 4.2 Time Protocol .
Exchange rate	rate	Yes	String(16)	1.234	Exchange rate

Example:

```

<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <device_info><![CDATA[1000]]></device_info>
  <nonce_str><![CDATA[TN55wO9Pba5yENi8]]></nonce_str>
  <sign><![CDATA[BDF0099C15FF7BC6B1585FBB110AB635]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>
  <is_subscribe><![CDATA[Y]]></is_subscribe>
  <trade_type><![CDATA[MICROPAY]]></trade_type>
  <bank_type><![CDATA[CCB_DEBIT]]></bank_type>
  <total_fee>1</total_fee>
  <fee_type><![CDATA[CNY]]></fee_type>
  <transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>
  <out_trade_no><![CDATA[1415757673]]></out_trade_no>
  <attach><![CDATA[Additional Order description]]></attach>
  <time_end><![CDATA[20141111170043]]></time_end>
  <trade_state><![CDATA[SUCCESS]]></trade_state>
</xml>

```

6 Error Code

Name	Description	Reason	Solution
ORDERNOTEXIST	This order does not exist	This order number does not exist in the query system	This API only helps query successfully paid transactions. The Vendor should check whether the provided transaction ID is correct.
SYSTEMERROR	System error	Exception occurs when data is returned from backend	This is caused by system error. Try to query again.

4.Submit Refund

1 Use Case

For a period after a payment transaction has been completed and a refund is required by either the Payer or Vendor, the Vendor can refund the Payer via this API. After the WeChat payment system receives and verifies the refund request successfully, the Payer will be refunded with the original payment amount according to the refund rules.

Notes:

1. For any transaction completed more than 6 months prior, a refund is not supported;

2. A refund for a transaction can be processed in the form of multiple partial refunds. In this case, the original order number is required and multiple refund numbers must be set. The total refund amount cannot exceed the original payment amount.

The Vendor should review the following Submit Refund flowchart demonstrating when this API is called successfully:

2 URL

URL: <https://api.mch.weixin.qq.com/secapi/pay/refund>

3 Certificate Requirement

Two-way certificate is required

4 Request Parameters

Field Name	ID	Required	Type	Example	Description

Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment. This field should match the value of device_info when the order was created.
Random String	nonce_str	Yes	String(32)	5K8264ILTKC H16CQ2502S I8ZNM67V S	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm .
Signature	sign	Yes	String(32)	C380BEC2BF D727A4B684 5133519F3AD 6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
WeChat Order Number	transaction_id	No	String(28)	12177525012 01407033233 368018	Specifies the WeChat payment order id number
Vendor Order Number	out_trade_no	No	String(32)	12177525012 01407033233 368018	out_trade_no is an internal order number within the vendor's system. transaction_id will be used over out_trade_no if they are both provided by the vendor.
Vendor Refund Number	out_refund_no	Yes	String(32)	12177525012 01407033233 368018	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.
Bid price	total_fee	Yes	Int	100	Specifies the total order amount. The units are expressed in cents and must be an integer. For more information, see Section 4.2 Payment Amount .
Refund Amount	refund_fee	Yes	Int	100	Specifies the total refund amount for a transaction. The units are expressed in cents and shall be an integer. Section 4.2 Payment Amount .
Operator	op_user_id	Yes	String(32)	1900000109	Specifies the Operator ID. This field shows vendor's ID by default.

Example:

```

<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <nonce_str>6cefdb308e1e2e8aabd48cf79e546a02</nonce_str>
  <op_user_id>10000100</op_user_id>
  <out_refund_no>1415701182</out_refund_no>
  <out_trade_no>1415757673</out_trade_no>
  <refund_fee>1</refund_fee>

```

```

<total_fee>1</total_fee>
<transaction_id></transaction_id>
<sign>FE56DD4AA85C0EECA82C35595A69E153</sign>
</xml>

```

5 Return Data

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL
Return Data	return_msg	No	String(128)	Signature failure	If not empty, this is the error description Signature failure Parameter format checking error

If return_code is SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Error Code	err_code	No	String(32)	SYSTEMERROR	For more information, see Section 9.6 Download Transaction History.
Error Code Description	err_code_desc	No	String(128)	System timed out	Describes result data
Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment. The value in this field must match the device_info value used when the order was created.
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer
Signature	sign	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
WeChat Order Number	transaction_id	Yes	String(28)	1217752501201407033233368018	Specifies the WeChat payment order id number
Vendor Order	out_trade_no	Yes	String(32)	12177525012014070332	Specifies an internal order number created by the Vendor's system

Number				33368018	
Vendor Refund Number	out_refund_no	Yes	String(32)	1217752501 2014070332 33368018	Vendor Refund Number
WeChat Refund Number	refund_id	Yes	String(28)	1217752501 2014070332 33368018	WeChat Refund Number
Refund Amount	refund_fee	Yes	Int	100	Specifies the total refund amount expressed in cents and must be an integer. For more information, see Section 4.2 Payment Amount A refund can be processed as multiple partial refunds.
Currency Type	refund_fee_type	No	String(8)	HKD	Complies with ISO 4217 standards , For more information, see Section 4.2 Currency Type .
Bid price	total_fee	Yes	Int	100	Specifies the total order amount expressed in cents and must be an integer. For more information, see Section 4.2 Payment Amount
Order Currency Type	fee_type	Yes	String(8)	CNY	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type .
Cash Payment Amount	cash_fee	Yes	Int	100	Specifies the cash payment amount expressed in cents and must be an integer. For more information, see Section 4.2 Payment Amount
Currency Type	cash_fee_type	No	String(8)	CNY	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type .
Cash Refund Amount	cash_refund_fee	No	Int	100	Specifies the cash refund amount expressed in cents and must be an integer. For more information, see Section 4.2 Payment Amount
Cash Refund Currency Type	cash_refund_fee_type	No	String(8)	HKD	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type .
Mobile Coupon Refund Amount	coupon_refund_fee	No	Int	100	A mobile coupon refund amount equals to the difference between order amount and cash refund amount. Note: Refunds are not supported for certain mobile coupons, such as discounts requiring a certain payment amount.
Mobile Coupon Quantity	coupon_count	No	Int	1	Specifies the quantity of mobile coupons
Mobile Coupon Batch ID	coupon_batch_id_\$n	No	String(20)	100	Specifies mobile coupon batch ID, where \$n is a suffixed counter ascending from 1
Mobile Coupon ID	coupon_id_\$n	No	String(20)	10000	Specifies mobile coupon ID, where \$n is a suffixed counter ascending from 1

Payment Amount for Each Coupon	coupon_fee_\$n	No	Int	100	Specifies payment amount for each mobile coupon, where \$n is a suffixed counter ascending from 1
--------------------------------	----------------	----	-----	-----	---

Example:

```

<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <nonce_str><![CDATA[NjsMFbUFpdbEhPXP]]></nonce_str>
  <sign><![CDATA[B7274EB9F8925EB93100DD2085FA56C0]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>
  <out_trade_no><![CDATA[1415757673]]></out_trade_no>
  <out_refund_no><![CDATA[1415701182]]></out_refund_no>
  <refund_id><![CDATA[2008450740201411110000174436]]></refund_id>
  <refund_channel><![CDATA[]]></refund_channel>
  <refund_fee>1</refund_fee>
  <coupon_refund_fee>0</coupon_refund_fee>
</xml>

```

6 Error Code

Name	Description	Reason	Solution
SYSTEMERROR	API return error	System timed out	Call this API again using the same parameters
INVALID_TRANSACTIONID	Invalid transaction_id	Requested parameters are not correct	Incorrect request parameters. Check whether the original transaction ID exists or whether data failed to be returned from the payment interface.
PARAM_ERROR	Parameter error	Requested parameters are not correct	Incorrect request parameters. Check the parameters and call the Submit Refund API again.
APPID_NOT_EXIST	APPID DOES NOT EXIST	No APPID in this parameter	Check whether the provided APPID is correct
MCHID_NOT_EXIST	MCHID DOES NOT EXIST	No MCHID in this parameter	Check whether the provided MCHID is correct
APPID_MCHID_NOT_MATCH	appid does not match mch_id	appid does not match mch_id	Check whether appid belongs to the associated mch_id
REQUIRE_POST_METHOD	Use post method	Data is not transferred by post	Check whether data is submitted by POST method
SIGNERROR	Signature error	Incorrect signature result	Check whether the signature parameter and method complies with signature algorithm requirements
XML_FORMAT_ERROR	INVALID XML FORMAT	INVALID XML FORMAT	Check whether XML parameters are in the correct format

5. Query Refund

1 Use Case

After submitting Submit Refund, this API can be called to check the refund status. After submitting a refund, there may be a delay in processing the refund: 20 minutes for refunding to Balance and 3 working days for refunding to a bank card.

2 URL

URL: <https://api.mch.weixin.qq.com/pay/refundquery>

3 Certificate Requirement

No certificate is required.

4 Request Parameters

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm .
Signature	sign	Yes	String(32)	C380BEC2BF727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
WeChat Order Number	transaction_id	No	String(28)	1217752501201407033233368018	Specifies the WeChat payment order id number
Vendor Order Number	out_trade_no	No	String(32)	1217752501201407033233368018	Specifies an internal order number created by the Vendor's system
Vendor Refund Number	out_refund_no	No	String(32)	1217752501201407033233368018	Vendor Refund Number
WeChat Refund Number	refund_id	No	String(28)	1217752501201407033233368018	WeChat Refund Number This field will supply the refund_id, out_refund_no, out_trade_no, or

					transaction_id. Their priority is as shown below: refund_id>out_refund_no>transaction_id>out_trade_no
--	--	--	--	--	--

Example:

```
<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <nonce_str>0b9f35f484df17a732e537c37708d1d0</nonce_str>
  <out_refund_no></out_refund_no>
  <out_trade_no>1415757673</out_trade_no>
  <refund_id></refund_id>
  <transaction_id></transaction_id>
  <sign>66FFB727015F450D167EF38CCC549521</sign>
</xml>
```

5 Return Data

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL
Return Data	return_msg	No	String(128)	Signature failure	If not empty, this is the error description Signature failure Parameter format checking error

If return_code is SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Service Result	result_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL SUCCESS indicates that Submit Refund is received successfully. The refund result can be queried via the Query Refund API. FAIL
Error Code	err_code	Yes	String(32)	SYSTEMERROR	For more information, see Section 9.6 Download Transaction History .
Error Description	err_code_des	Yes	String(32)	System error	Describes result data
Official Account ID	appid	No	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat

Vendor ID	mch_id	Yes	String(32)	190000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	Yes	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment. The value in this field must match the device_info value used when the order was created.
Random String	nonce_str	Yes	String(28)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
WeChat Order Number	transaction_id	Yes	String(32)	1217752501201407033233368018	Specifies the WeChat payment order id number
Vendor Order Number	out_trade_no	Yes	String(32)	1217752501201407033233368018	Specifies an internal order number created by the Vendor's system
Refund Count	refund_count	Yes	Int	1	Specifies recorded refund counts
Vendor Refund Number	out_refund_no_\$n	Yes	String(32)	1217752501201407033233368018	Vendor Refund Number
WeChat Refund Number	refund_id_\$n	Yes	String(28)	1217752501201407033233368018	WeChat Refund Number
Refund Channel	refund_channel_\$n	No	String(16)	ORIGINAL	ORIGINAL: Refund to original payment account BALANCE: Refund to Balance
Refund Amount	refund_fee_\$n	Yes	Int	100	Specifies the total refund amount expressed in cents and must be an integer. For more information, see Section 4.2 Payment Amount A refund can be processed as multiple partial refunds.
Exchange rate	rate	Yes	String(16)	1.234	Exchange rate
Currency Type	refund_fee_type	No	String(8)	HKD	Complies with ISO 4217 standards , For more information, see Section 4.2 Currency Type .
Cash Coupon Refund Amount	coupon_refund_fee_\$n	No	Int	100	The mobile coupon refund amount will be equal to or less than the refund amount. That is, the cash amount is the difference between the refund amount and the mobile coupon refund amount. For more information, see Section 10 Mobile

					Coupon
Refund Status	refund_status_\$n	Yes	String(16)	SUCCESS	<p>Refund Status:</p> <p>SUCCESS: Refunded successfully</p> <p>FAIL: Refund failed</p> <p>PROCESSING: Refund is pending</p> <p>NOTSURE: Require the Vendor to call the Submit Refund API again with the original refund number</p> <p>CHANGE: Refund can't be processed as the Payer's bank card is either revoked or blocked. As a consequence, the refund will be transferred to the Vendor's cash account. In this case, the refund must be processed offline via the help of the vendor's customer support staff or by transferring the refund amount from the Vendor to the Payer via Tenpay.</p>

Example:

```

<xml>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <nonce_str><![CDATA[TeqCIE3i0mvm3DrK]]></nonce_str>
  <out_refund_no_0><![CDATA[1415701182]]></out_refund_no_0>
  <out_trade_no><![CDATA[1415757673]]></out_trade_no>
  <refund_count>1</refund_count>
  <refund_fee_0>1</refund_fee_0>
  <refund_id_0><![CDATA[2008450740201411110000174436]]></refund_id_0>
  <refund_status_0><![CDATA[PROCESSING]]></refund_status_0>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <sign><![CDATA[1F2841558E233C33ABA71A961D27561C]]></sign>
  <transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>
</xml>

```

6 Error Code

Name	Description	Reason	Solution
SYSTEMERROR	API return error	System timed out	Try to call the API again
INVALID_TRANSACTIONID	Invalid transaction_id	Requested parameters are not correct	Incorrect request parameters. Check whether the original transaction ID exists or whether data failed to be returned from the payment interface.
PARAM_ERROR	Parameter error	Requested parameters are not correct	Incorrect request parameters. Check parameters and call the Submit Refund API again.

APPID_NOT_EXIST	APPID DOES NOT EXIST	No APPID in this parameter	Check whether provided APPID is correct
MCHID_NOT_EXIST	MCHID DOES NOT EXIST	No MCHID in this parameter	Check whether provided MCHID is correct
APPID_MCHID_NOT_MATCH	appid does not match mch_id	appid does not match mch_id	Check whether appid belongs to the associated mch_id
REQUIRE_POST_METHOD	Use post method	Data is not transferred via POST method	Check whether data is submitted by POST method
SIGNERROR	Signature error	Incorrect signature result	Check whether signature parameter and method comply with signature algorithm requirements
XML_FORMAT_ERROR	INVALID XML FORMAT	INVALID XML FORMAT	Check whether XML parameters are in correct format

6. Download Transaction History

1 Use Case

This API is used to help the Vendors download their transaction record history including missing orders, and can be used to find unmatched data between vendors and WeChat caused by system error. By downloading transaction history and doing reconciliation, the Vendor can correct the status of their orders in a timely fashion.

Notes:

1. Transactions that are not ordered successfully are excluded in this reconciliation form, while orders that are revoked after successful payment will still be included. Included orders shall be consistent with the original payment bill number, that is, the value of bill_type will be set to REVOKED;

2. Each day's transaction history download is created at 9:00 AM the following day. Therefore, vendors can download the history after 10:00 AM (UTC+08);

3. The currency unit used in the reconciliation form is CNY (Chinese yuan).

2 URL

<https://api.mch.weixin.qq.com/pay/downloadbill>

3 Certificate Requirement

No certificate is required.

4 Request Parameters

Field Name	ID	Required	Type	Example	Description
Official	appid	Ye	String(32)	wx88888888	Specifies Official Account ID assigned by

Account ID		s		88888888	WeChat
Vendor ID	mch_id	Yes	String(32)	190000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment. This field specifies the transactions related to this device.
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502S18ZNM67VS	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm .
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
Reconciliation Start Date	bill_date	Yes	String(8)	20140603	Specifies the date of the transactions to be downloaded in the format of yyyyymmdd, such as 20140603 for June 3, 2014.
Bill Type	bill_type	No	String(8)	ALL	ALL: Return all order data from the specified date. bill_type is set to ALL by default. SUCCESS: Return successfully paid orders only REFUND: Return refunded orders only REVOKED: Return revoked orders revoked only

```

<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <bill_date>20141110</bill_date>
  <bill_type>ALL</bill_type>
  <mch_id>10000100</mch_id>
  <nonce_str>21df7dc9cd8616b56919f20d9f679233</nonce_str>
  <sign>332F17B766FC787203EBE9D6E40457A1</sign>
</xml>

```

5 Return Data

If the request has failed, the following fields will be returned.

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String(16)	FAIL	FAIL
Return Data	return_msg	No	String(128)	Signature failure	If not empty, this is the error description Signature failure Parameter format checking error

					The order date is not created
--	--	--	--	--	-------------------------------

When the query has succeeds, the returned data will include a header row containing the data fields of the subsequent rows.

The first row will be the header row, with the value based upon the Bill Type (as specified by bill_type) requested by the Vendor and containing the included data fields of the subsequent row. Each bill_type category will contain the following fields:

All orders from specified day

transaction time,wechat order number,merchant order number,payment type ,transaction status,order amount,order currency,foreign exchange amount,foreign exchange currency,customer payment amount,customer payment currency,applying refund time,successful refund time,wechat refund order number,merchant refund order number,refund amount,refund currency,foreign exchange refund amount,foreign exchange refund currency,customer refund amount,customer refund currency ,transaction description,exchange rate,fees,attach,appid,merchant id,sub merchant id,device,open id, coupon amount,coupon refund amount,coupon currency,trade type,refund type

total count,total foreign exchange amount,total foreign exchange refund amount

SUCCESS:

transaction time,wechat order number,merchant order number,payment type ,transaction status,order amount,order currency,foreign exchange amount,foreign exchange currency,customer payment amount,customer payment currency,applying refund time,successful refund time,wechat refund order number,merchant refund order number,refund amount,refund currency,foreign exchange refund amount,foreign exchange refund currency,customer refund amount,customer refund currency ,transaction description,exchange rate,fees,attach,appid,merchant id,sub merchant id,device,open id, coupon amount,coupon refund amount,coupon currency,trade type,refund type

total count,total foreign exchange amount

REFUND:

transaction time,wechat order number,merchant order number,payment type ,transaction status,order amount,order currency,foreign exchange amount,foreign exchange currency,customer payment amount,customer payment currency,applying refund time,successful refund time,wechat refund order number,merchant refund order number,refund amount,refund currency,foreign exchange refund amount,foreign exchange refund currency,customer refund amount,customer refund currency ,transaction description,exchange rate,fees,attach,appid,merchant id,sub merchant id,device,open id, coupon amount,coupon refund amount,coupon currency,trade type,refund type

total count,total foreign exchange refund amount

REVOKED:

transaction time,wechat order number,merchant order number,payment type ,transaction status,order amount,order currency,foreign exchange amount,foreign exchange currency,customer payment amount,customer payment currency,applying refund time,successful refund time,wechat refund order number,merchant refund order number,refund amount,refund currency,foreign exchange refund amount,foreign exchange refund currency,customer refund amount,customer refund currency ,transaction description,exchange rate,fees,attach,appid,merchant id,sub merchant id,device,open id, coupon amount,coupon refund amount,coupon currency,trade type,refund type

6 Error Code

Name	Description	Reason	Solution
SYSTEMERROR	API return error	System timed out	Try to query again
INVALID_TRANSACTIONID	Invalid transaction_id	Requested parameters are not correct	Parameter error. Check transaction_id and try again.
PARAM_ERROR	Parameter error	Requested parameters are not correct	Parameter error. Check parameters and try again.

7. General Payment Result Notification

1 Use Case

After completing a payment, the WeChat payment system will send the relevant payment result and user information to the Vendor. When this happens, the Vendor's backend will need to receive the result and return a reply to the WeChat payment system.

When interacting with this API, if the WeChat payment system does not receive a notification from the Vendor backend indicating success or timeout, the WeChat payment system will consider it as an unreceived notification and initiate further payment result notifications at a regular interval, such as 8 times in 1440 minutes, so as to ensure successful receipt. However, the WeChat payment system cannot ensure successful receipt of payment notifications in every case.

As payment result notifications may be sent from the WeChat payment system to the Vendor's backend multiple times, a single payment result might be notified to the Vendor's backend multiple times. For this reason, the Vendor's system must be able to handle duplicate notifications properly.

The best practices for dealing with such case: when a notification is received and processed, the enclosed payment data should be checked first to confirm whether the payment result has previously been processed or not. If so, return the processed result; if no, proceed with processing the result first before returning. Before checking the payment data, a transaction lock shall be used for concurrency to prevent data corruption caused by transaction race conditions.

Developers can log in to the WeChat payment system for vendors and join the API warning group.



2 URL

This URL may be configured via `notify_url`, a parameter submitted via the [【Unified Order API】](#), If vendors can't open this URL, they will not be able to receive any notifications sent from the WeChat payment system.

3 Certificate Requirement

No certificate is required.

4 Notification Parameters

Field Name	ID	Required	Type	Example	Description
Return Status Code	<code>return_code</code>	Yes	String(16)	SUCCESS	SUCCESS or FAIL Specifies communicating label instead of transaction label. The status of a transaction is determined by the value of <code>result_code</code> .
Return Data	<code>return_msg</code>	No	String(128)	Signature failure	If not empty, this is the error description Signature failure

					Parameter format checking error
--	--	--	--	--	---------------------------------

If return_code is SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx8888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	Specifies the terminal device ID assigned by WeChat Payment
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNM67VS	32 characters or fewer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm .
Service Result	result_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL
User Tag	openid	Yes	String(128)	wxd930ea5d5a258f4f	Specifies the user id of the Payer provided the WeChat system in OpenID format unique tag on vendor's appid unique to each appid instance
Follows Official Account or not	is_subscribe	Yes	String(1)	Y	Specifies whether the payer follows the associated official account or not, with Y meaning 'follows' and N meaning 'not follows'.
Transaction Type	trade_type	Yes	String(16)	MICROPAY	MICROPAY
Payment Bank	bank_type	Yes	String(16)	CMC	Use strings for bank type. For more information, see attachments.
Bid price	total_fee	Yes	Int	100	Specifies the total amount for a transaction in cents. For more information, see Section 4.2. Payment Amount .
Currency Type	fee_type	No	String(8)	HKD	Complies with ISO 4217 standards . For more information, see Section 4.2 Currency Type
Cash Payment Amount	cash_fee	Yes	Int	100	Specifies the total amount of a transaction. For more information, see Section 4.2 Payment Amount .
Cash Type	cash_fee_type	No	String(16)	HKD	Complies with ISO 4217 standards . For more information, see Section 4.2 Currency Type

Mobile Coupon Amount	coupon_fee	No	Int		The amount for mobile coupons will be equal to or less than the total order amount. The cash payment amount is the difference between the total order amount and the mobile coupon amount. For more information, see Section 4.2 Payment Amount .
Mobile Coupon Quantity	coupon_count	No	Int	1	Specifies the quantity of mobile coupons
Mobile Coupon Batch ID	coupon_batch_id_\$n	No	String(20)	100	Specifies mobile coupon batch ID, where \$n is a suffixed counter ascending from 1
Mobile Coupon ID	coupon_id_\$n	No	String(20)	10000	Specifies mobile coupon ID, where \$n is a suffixed counter ascending from 1
Payment Amount for Each Coupon	coupon_fee_\$n	No	Int	100	Specifies payment amount for each mobile coupon, where \$n is a suffixed counter ascending from 1
Specifies the number of a WeChat payment order	transaction_id	Yes	String(32)	121775250 120140703 323336801 8	Specifies the WeChat payment order id number
Vendor Order Number	out_trade_no	Yes	String(32)	121232121 120140703 356811232 2	Specifies an order number created by a Vendors' system, which is consistent with request.
Vendor's Data Package	attach	No	String(128)	123456	Specifies vendor's data package, which is returned as it is.
Payment End Time	time_end	Yes	String(14)	201410301 33525	Specifies the time of completing payment in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. For more information, see Section 3.2 Time Protocol .

Example:

```

<xml>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <attach><![CDATA[Payment Testing]]></attach>
  <bank_type><![CDATA[CFT]]></bank_type>
  <fee_type><![CDATA[CNY]]></fee_type>
  <is_subscribe><![CDATA[Y]]></is_subscribe>
  <mch_id><![CDATA[10000100]]></mch_id>
  <nonce_str><![CDATA[5d2b6c2a8db53831f7eda20af46e531c]]></nonce_str>
  <openid><![CDATA[oUpF8uMEb4qRXf22hE3X68TekukE]]></openid>
  <out_trade_no><![CDATA[1409811653]]></out_trade_no>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <sign><![CDATA[B552ED6B279343CB493C5DD0D78AB241]]></sign>
  <sub_mch_id><![CDATA[10000100]]></sub_mch_id>

```

```

<time_end><![CDATA[20140903131540]]></time_end>
<total_fee>1</total_fee>
<trade_type><![CDATA[JSAPI]]></trade_type>
<transaction_id><![CDATA[1004400740201409030005092168]]></transaction_id>
</xml>

```

5 Return Data

Parameters returned to the WeChat payment system by the Vendor's backend immediately after the Vendor has processed the payment result notification:

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL SUCCESS indicates that Vendor has received payment result notification and has verified the payment successfully
Return Data	return_msg	No	String(128)	OK	If not empty, this is the error description: Signature failure Parameter format checking error

Example:

```

<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
</xml>

```

8. Querying Settled Funds

1 Use Case

This API helps query the details of settled funds.

2 URL Entry

<https://api.mch.weixin.qq.com/pay/settlementquery>

3 Request Parameters

Field Name	ID	Type	Required	Example	Description
Official Account ID	appid	String(32)	Yes	Wx1378 acui786 5dt65	Specifies Official Account ID assigned by WeChat

Vendor ID	mch_id	String(10)	Yes	12567876 549	Specifies vendor ID assigned by WeChat Pay
Settlement Status	usetag	Int	Yes	1	Indicates the fund has been settled or is still outstanding: 1 - settled 2 - outstanding
Random String	nonce_str	String(32)	Yes	Uihu276j ghxlehu38 76440932 2	A random string of less than 32 chars
Offset	offset	Int	Yes	1	Returned query data starts from this offset value
Max Records	limit	Int	Yes	10	The number of returned records (generally less than 10)
Start Date	date_start	String(14)	Yes	20150807	In the format of yyyyMMdd. For example, December 25, 2009 is expressed as 20091225 in the time zone (GMT+8).
End Date	date_end	String(14)	Yes	20150807	In the format of yyyyMMdd. For example, December 25, 2009 is expressed as 20091225 in the time zone (GMT+8).
Signature	sign	String(32)	Yes	8439EB D8AE74 22590E2 004D0C 5EBFFB C	For details about signature, see Section 3.2

4 Return Parameters

Field Name	ID	Required	Type	Description
Return Status Code	return_code	Yes	String(16)	SUCCESS/FAIL Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.

Return Data	return_msg	No	String(128)	If returned data is not empty, the returned data is the error description of the following: - Signature Failure - Parameter format checking error
If return_code is SUCCESS, the return data also includes the following fields:				
Official Account ID	appid	Yes	String(32)	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	Specifies vendor ID assigned by WeChat Pay
Random String	nonce_str	Yes	String(32)	Indicates a random string of less than 32 chars
Service Result	result_code	Yes	String(16)	SUCCESS/FAIL
Error Code	err_code	No	String(32)	fail
Error Code Description	err_code_des	No	String(128)	Describes result data
Return Data Lines	record_num	Yes	String(10)	Indicates number of lines of return data
The following fields will be returned when result_code is SUCCESS . In case of multiple records, the data will be repeated.				
Foreign Payment Batch Number	FBatchNo	Yes	String(32)	Indicates the batch number of foreign payment
Settlement Date	date_settlement	Yes	String(14)	In the format of yyyyMMdd. For example, December 25, 2009 is expressed as 20091225 in the time zone (GMT+8).
Transaction Start Date	date_start	Yes	String(14)	In the format of yyyyMMdd. For example, December 25, 2009 is expressed as 20091225 in the time zone (GMT+8).
Trasanction End Date	date_end	Yes	String(14)	In the format of yyyyMMdd. For example, December 25, 2009 is expressed as 20091225 in the time zone (GMT+8).
Remit Amount	settlement_fee	Yes	String(32)	Priced in foreign currency at the minimum trading unit

Non-Remit Amount	unsettlement_fee	Yes	String(32)	Priced in foreign currency at the minimum trading unit
Settlement Currency	settlementfee_type	Yes	String(8)	<p>Comply with ISO 4217 standards and use CNY for Chinese currency by default.</p> <p>Description:</p> <p>CNY: Chinese yuan</p> <p>GBP: Great Britain pound</p> <p>HKD: Hong Kong dollar</p> <p>USD: United States dollar</p> <p>JPY: Japanese yen</p> <p>CAD: Canadian dollar</p> <p>AUD: Australian dollar</p> <p>EUR: Euro cent</p> <p>Notes: The currency type for payment and refund shall be the same.</p>
Payment Amount	pay_fee	Yes	String(8)	Priced in foreign currency at the minimum trading unit
Refund Amount	refund_fee	Yes	String(8)	Priced in foreign currency at the minimum trading unit
Net Payment Amount	pay_net_fee	Yes	String(8)	Priced in foreign currency at the minimum trading unit
Charge Amount	poundage_fee	Yes	String(8)	Priced in foreign currency at the minimum trading unit