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) Native 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 (Official Account 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) 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.

11) 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 Official Account 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.

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 Official Account 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 or HMAC-SHA256
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

NZD：New Zealand Dollar

KRW：South Korean won

THB: Thailand baht

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.

6) Requirements for field body

Some special standards are required for field "body" when calling the WeChat Payment APIs:

Scenario	Payment Method	Input Standard	Example	Remark
PC web site	Native Quick Pay	Web site title--Product description	Tencent recharge center—QQ member recharging	
WeChat browser	Official Account Payment	Merchant name--Product category	Dangdang-Books	
QR code in stores	Official Account Payment	Store name—Product category	Southern store-Super market	Offline store payment
QR code in stores	Native Quick Pay	Store name—Product category	Southern store-Super market	Offline store payment
Quick Pay in stores	Quick Pay	Store name—Product category	Southern store-Super market	Offline store payment
Third party mobile browser	H5 Payment	Mobile web site title—Product	Tencent recharge center—QQ member	

		description	recharging	
Third party APP	In-App Payment	App name—Product description	Cool Running-Recharging	

3. Access Nodes for Payment API

There are two access nodes for payment API for overseas merchants:

Server deployed in east-southern Asia: https://apihk.mch.weixin.qq.com/

Server deployed in Europe or America: https://apius.mch.weixin.qq.com/

Server deployed in China: https://api.mch.weixin.qq.com/

Note: Merchants should choose the best access node for payment API according to their server deployed area. It is advised to consider the compatibility for other access nodes. Once there is problem about the main node, the system could automatically switch to the other nodes.

4. 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>wxd930ea5d5a258f4f</appid>
    <mch_id>10000100</mch_id>
    <device_info>1000<device_info>
    <body>test</body>
    <nonce_str>ibuaiVcKdpRxkhJA</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	Calling the Revoke Order API and Submit Refund	Double-click to import into a Windows system and enter certificate password as

	by WeChat payment for identity verification	API	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 call APIs, the authenticity of the server called and domain name shall be verified.	This file is the root certificate issued by authorities that sign WeChat payment certificates, which can be used to verify the authenticity of WeChat payment server certificates.	Root certificates are built-n to some tools. For tools without root certificates, the ones provided here may be used.

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.

4. Getting OpenID

WeChat Official Account Admin Platform:

After messaging between a follower and Official Accounts, the Official Accounts can get the follower's OpenID. Each user has a unique OpenID after its WeChat ID is encrypted. User's OpenID varies from different Official Accounts.

Official Accounts can get user's OpenID by the API below. For information about user's alias, photo picture, gender, location, language, and following time, user's authorization is required.
URL: http://mp.weixin.qq.com/wiki/17/c0f37d5704f0b64713d5d2c37b468d75.html

To use OpenIDs from different platforms as one ID for the same user, developers can use the API below:
URL: http://mp.weixin.qq.com/wiki/14/bb5031008f1494a59c6f71fa0f319c66.html

WeChat Open Platform:

Mobile apps can use the API below to get user's OpenID:
https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419317851&token=&lang=zh_CN

Website apps can use the API below to get user OpenID:
https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419316505&token=&lang=zh_CN

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.

Note: The WeChat barcodes are constructed with 18 numbers, with the start value of 10, 11, 12, 13, 14 or 15.

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 1000 CNY;
- 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.

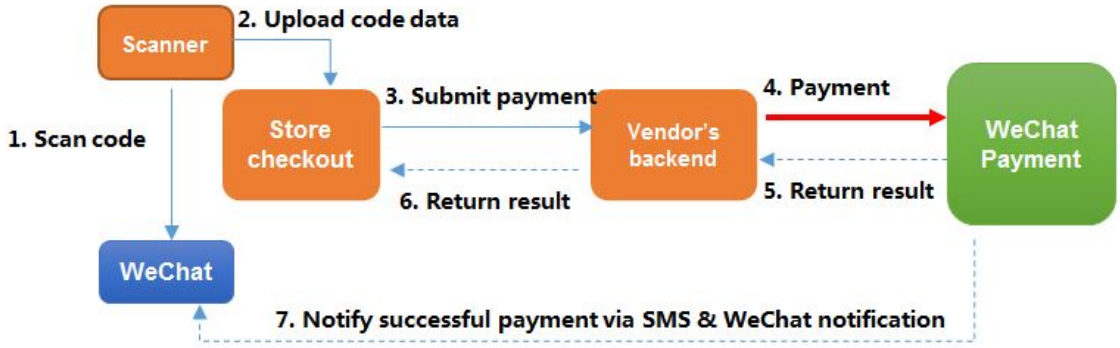

Figure 5.4 Vendors' Backend Access Process

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.

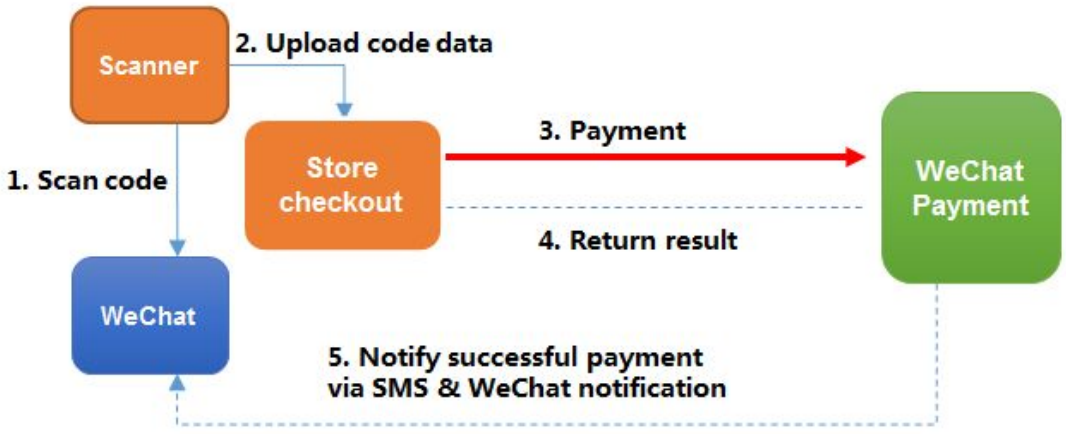

Figure 5.5 Physical Store Access Process

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.

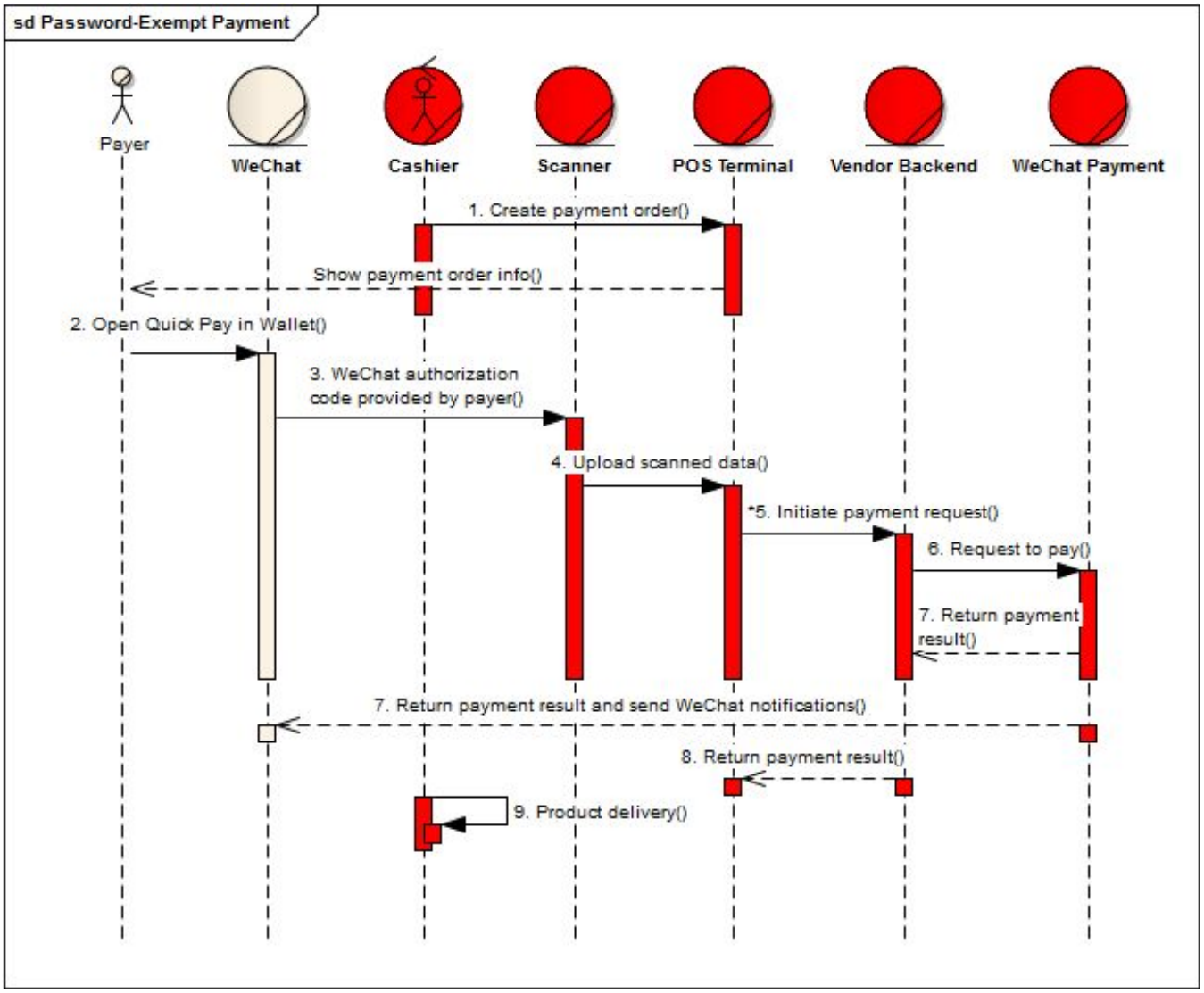

Figure 5.6 Sequence Chart of Password-Exempt Payment

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:

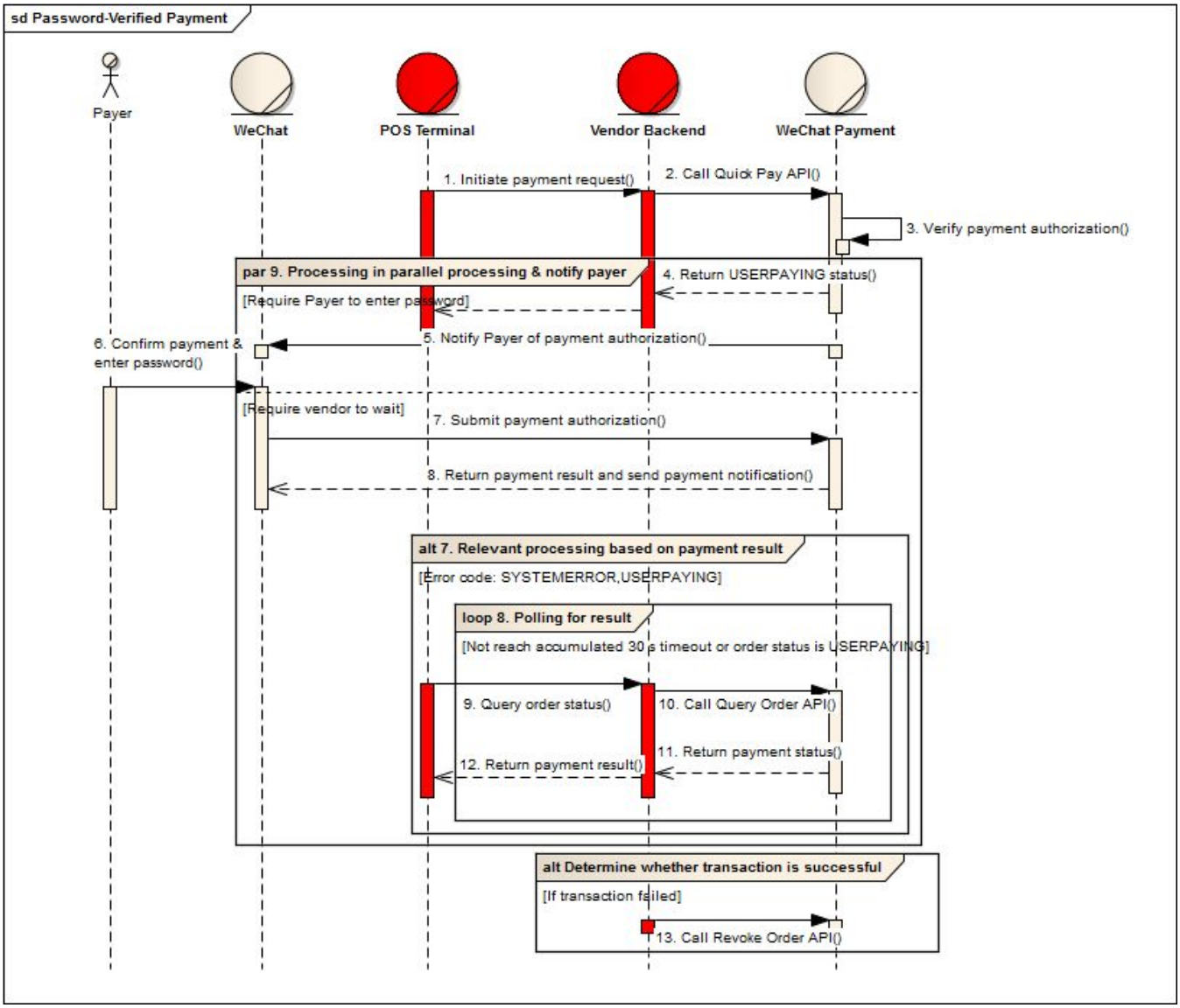

Figure 5.7 Process Diagram for Password-Verified Payment

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.

5. 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.

Note: a. Trade result will be returned synchronously. Query Order API should be called 5s later if it returns "System error". This API is also required once it returns "USERPAYING", and it should be called continuously until the payment is successful or timeout (advised timeout time is 30s, advised calling frequency is 10s once).

b. Please revoke the order if it returns an uncertain trade status when calling Query Order API. After revoking, the successful order will be refunded, while the failed order will be closed. Please call again if it returns failed when calling Revoke Order API. Please notice that Revoke Order API should be called at least 15s later after the order is created, and it requires passing the certificate.

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	Signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	String(32)	NO	HMAC-SHA256/MD5	Currently support HMAC-SHA256 and MD5, default is MD5. If you choose HMAC-SHA256, sign_type must be pass
Item Description	body	String(32)	Yes	Pay for QQ Coins	Short description of item(s) to be purchased for the Payer, Please refer to Section 4.2, item 6
Item Details	detail	String(6000)	No	{ "goods_detail":[{ "goods_id":"ipho	Detailed product list described in JSON format. Please use CDATA tag to protect the JSON string when generating signature.

				ne6s_16G",
				"wxpay_goods_id":"1001",
				"goods_name":"iPhone6s 16G",
				"goods_num":1,
				"price":528800,
				"goods_category":"123456",
				"body":"苹果手机"
				},
				{
				"goods_id":"iphone6s_32G",
				"wxpay_goods_id":"1002",
				"goods_name":"iPhone6s 32G",
				"quantity":1,
				"price":608800,
				"goods_category":"123789",
				"body":"苹果手机"
				}
				]
				}
					goods_detail []:
					└ goods_id String Required 32 Goods ID
					└ wxpay_goods_id String Optional 32 The unified goods ID defined by WeChat
					└ goods_name String Required 256 Goods name
					└quantity Int Required Goods amount
					└ price Int Required Goods price, unit as cent
					Note: The goods price should be less than total_fee and it should be the favorable price.

Additional Data	attach	String(128)	No	User-Defined Data	Allow vendors an additional field to be returned in the payment notification after submitting a payment or in the Query Order API
Vendor Order Number	out_trade_no	String(32)	Yes	12177525012014 07033233368018	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(8)	Yes	GBP	ISO-4217 standard compliant and be described by three characters based code. 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
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 by scanning a barcode/QR Code on Quick Pay **Note: The WeChat barcodes are constructed with 18 numbers, with the start value of 10, 11, 12, 13, 14 or 15.**

Example：

```
<xml>
   <appid>wx2421b1c4370ec43b</appid>
   <attach>Additional Order Descriptionr</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>
   <nonce_str>8aaee146b1dee7cec9100add9b96cbe2</nonce_str>
   <out_trade_no>1415757673</out_trade_no>
   <spbill_create_ip>14.17.22.52</spbill_create_ip>
   <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 but not 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)	wx888888888888888	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The vendor ID submitted when calling the interface
Device ID	device_info	No	String(32)	013467007045764	The device ID submitted when calling the interface
Random String	nonce_str	Yes	String(32)	Fsdfds1235df231asdfg32145gfdse	32 characters or fewer, returned from Wechat payment
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Signature returned from Wechat payment. For more information, see Section 4.3.1 Signature Algorithm
Service Result	result_code	Yes	String(16)	SUCCESS	Set to SUCCESS or FAIL
Error Code	err_code	No	String(32)	SYSTEMERROR	For more information, please refer to Section 5.5.6 Error Codes
Error Code Description	err_code_des	No	String(128)	System error	The detailed description of error returned

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

Field Name	ID	Required	Type	Example	Description
User Tag	openid	Yes	String (128)	oUpF8uMuAJO_M2pxb1Q9zNjWeS6o	The only identification under the current appid
Follows Official Account or	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

not					account Y: Follows N: Doesn't follow
Transaction Type	trade_type	Yes	String (16)	MICROPAY	The transaction type is MICROPAY(quick pay)
Payment Bank	bank_type	Yes	String (32)	CMC	Strings states bank type
Currency Type	fee_type	Yes	String (8)	GBP	Complies with ISO 4217 standards and uses 3 characters based code. For more information, see Section 4.2 Currency Type.
Total Amount	total_fee	Yes	Int	100	Specifies the total amount for a transaction. The unit is cent and the value must be integer. For more information, see Section 4.2. Payment Amount.
Cash Type	cash_fee_type	Yes	String (8)	CNY	Payer's currency type. 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.
WeChat Payment Order Number	transaction_id	Yes	String (32)	1217752501 2014070332 33368018	The WeChat payment order id
Vendor Order Number	out_trade_no	Yes	String (32)	1217752501 2014070332 33368018	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)	2014103013 3525	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.

Exchange Rate	rate	Yes	String (16)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000

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	Payment Status	Reason	Solution
SYSTEMERROR	API return error	Unknown	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	Failure	Requested parameters are not correct	Debug your program based on data returned by the API
ORDERPAID	Order is paid	Failure	Duplicate order number	Confirm whether this order has already been processed. If not, submit it as a new order.

NOAUTH	No permissions for vendors	Failure	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.	Failure	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	Failure	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.*
NOTSUPORTCARD	Unsupported card type	Failure	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	Failure	The order is closed	An exception has occurred with this transaction. Create a new order and redo the payment process.
ORDERREVERSED	Order is cancelled	Failure	The order is cancelled	The current order is cancelled. Create a new order and redo the payment process.
BANKERROR	Bank system exceptions	Unknown	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 performing their payment	Unknown	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	Failure	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	Failure	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	Failure	Invalid XML format	Check whether XML parameters are in the correct format
REQUIRE_POST_METHOD	Use post method	Failure	Data not transferred via POST method	Check whether data is submitted via POST method
SIGNERROR	Signature error	Failure	Incorrect signature result	Check whether the signature parameter and method comply with signature algorithm requirements
LACK_PARAMS	Missing parameter	Failure	Required parameter is missing	Check whether all required parameters are complete
NOT_UTF8	Invalid coding format	Failure	Specified coding format is not used	Use NOT_UTF8 encoding format
BUYER_MISMATCH	Incorrect payment account	Failure	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	Failure	No APPID in this parameter	Check whether provided APPID is correct
MCHID_NOT_EXIST	MCHID does not exist	Failure	No MCHID in this parameter	Check whether provided MCHID is correct
OUT_TRADE_NO_USED	Duplicate vendor order number	Failure	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	Failure	appid does not match mch_id	Check whether appid belongs to the associated mch_id

6. Query Order API

For more information, see 【Section 9.2 Query Order】.

7. Submit Refund API

For more information, see 【Section 9.4 Submit Refund】.

8. Query Refund API

For more information, see 【Section 9.5 Query Refund】.

9. Revoke Order 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. After revoking, the successful order will be refunded, while the failed order will be closed.

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】 needs to be called. When there is no clear query result, 【Revoke Order API】 needs to be called.

Note: Please notice that Revoke Order API should be called at least 15s later after the order is created.

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	No	String (32)	1217752501201407033233368018	WeChat order number is preferred
Vendor Order Number	out_trade_no	Yes	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)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer or fewer. For more information, see Section 4.3.2 Random String Algorithm
Signature	sign	Yes	String (32)	C380BEC2BFD727A4B6845133519F3AD6	For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	NO	String (32)	HMAC-SHA256/MD5	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This parameter must be submitted if HMAC-SHA256 is chosen

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>
```

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 but not 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, 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	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String (32)	1900000109	The vendor ID submitted when calling the interface
Random String	nonce_str	Yes	String (32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer
Signature	sign	Yes	String (32)	C380BEC2BFD727A4B6845133519F3AD6	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;
Error Code	err_code	No	String (32)	SYSTEMERROR	For more information, please refer to Section 5.9.5 Error Code
Error Code Description	err_code_des	No	String (128)	System error	The detailed description of error returned.
Recall Requirement	recall	Yes	String (1)	Y	Specifies whether recalling the Cancel Order API is required or not, Y means yes while N means no.
Exchange Rate	rate	Yes	String (16)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000

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>
```

6 Error Code

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
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 method	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

10. Download Transaction History API

For more information, see 【Section 9.6 Download Transaction History】

11. Report Speed Testing

To improve our payment service, we recommend vendors report payment interface delays to the WeChat payment backend via the Report Speed Testing API. For more information, see 【Section 9.8 Report Speed Testing API】.

6. NATIVE QUICK PAY PROGRAMMING GUIDE

1. Use Case

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

Step 1： The Vendor creates a QR Code for their products correspondingly based on WeChat payment rules, as shown in Figure 6.1.

Step 2：The Payer scans the QR Code with their WeChat to access the Vendor’s product data and proceeds with the transaction, as shown in Figure 6.2. The Payer then makes their payment as instructed, as shown in Figure 6.3.

Figure 6.1 Payment QR Code

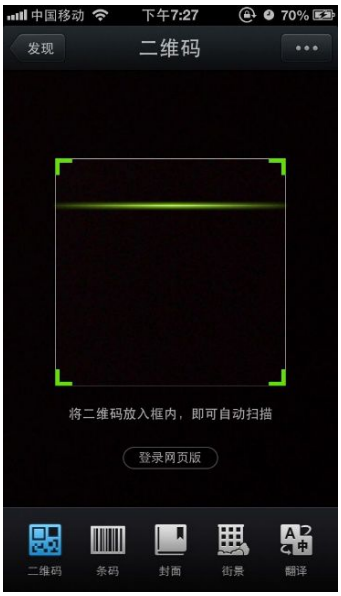

Figure 6.2 Scan QR Code on WeChat

Figure 6.3 Confirm Payment

Step 3：The Payer confirms their transaction and enters their payment password, as shown in Figure 6.4.

Step 4: The Payer is prompted of a successful payment after completing their payment, as shown in Figure 6.5. The Vendor delivers the paid products to the Payer after receiving a notification about the successful payment.

Figure 6.4: Enter Payment Password

Figure 6.5 Prompt after Successful Payment

2. Participating Vendors

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

Offline: Shopin physical stores

Online: JD.com, Ctrip, etc.

3. Process for Vendors

1 Enable Payment Authorization

On the left navigation bar of the WeChat Official Account Admin Platform, choose “WeChat Payment（微信支付）”-> “Developer Settings (开发配置)” and click “Edit (修改)”, as shown in Figure 6.6.

Figure 6.6 Parameter Settings of Native Payment

Check the checkbox for “Native Payment (Native 原生支付)” under Official Account Payment and enter the Vendor's payment backend URL in the “Payment Callback URL (支付回调 URL)” field.

Figure 6.7 Authorization Settings and Payment Callback URL for Native Payment

2 Select a Mode as Required

There are two modes for Quick Pay for vendors to choose from.

Mode 1: The Vendor's backend creates a QR Code based on WeChat payment rule URL containing a fixed parameter "productid" (specifies product label or order number). After the Payer scans the QR Code, the WeChat payment system sends the productid and OpenID (Payer's identity) to the Vendor’s backend. The Vendor's backend initiates a transaction based on the productid. The WeChat payment system starts the payment process on the Payers' WeChat.

Mode 2: The Vendor's backend calls the WeChat payment system 【Unified Order API】 in order to generate an advance transaction and creates a QR Code based on the URL returned by the API. The Payer enters their password to complete the transaction.

Note: The advance transaction is valid for 2 hours and cannot be paid for once it expires.

4. Mode 1

Before proceeding with this mode, the Vendor needs to set their payment callback URL on the WeChat Official Account Admin Platform. This URL is used to receive productid and the payer's OpenID via a callback by the WeChat payment system after the Payer scans a QR Code. For more information, see Section 6.3.1 Enable Payment Authorization.

1 Sequence Chart

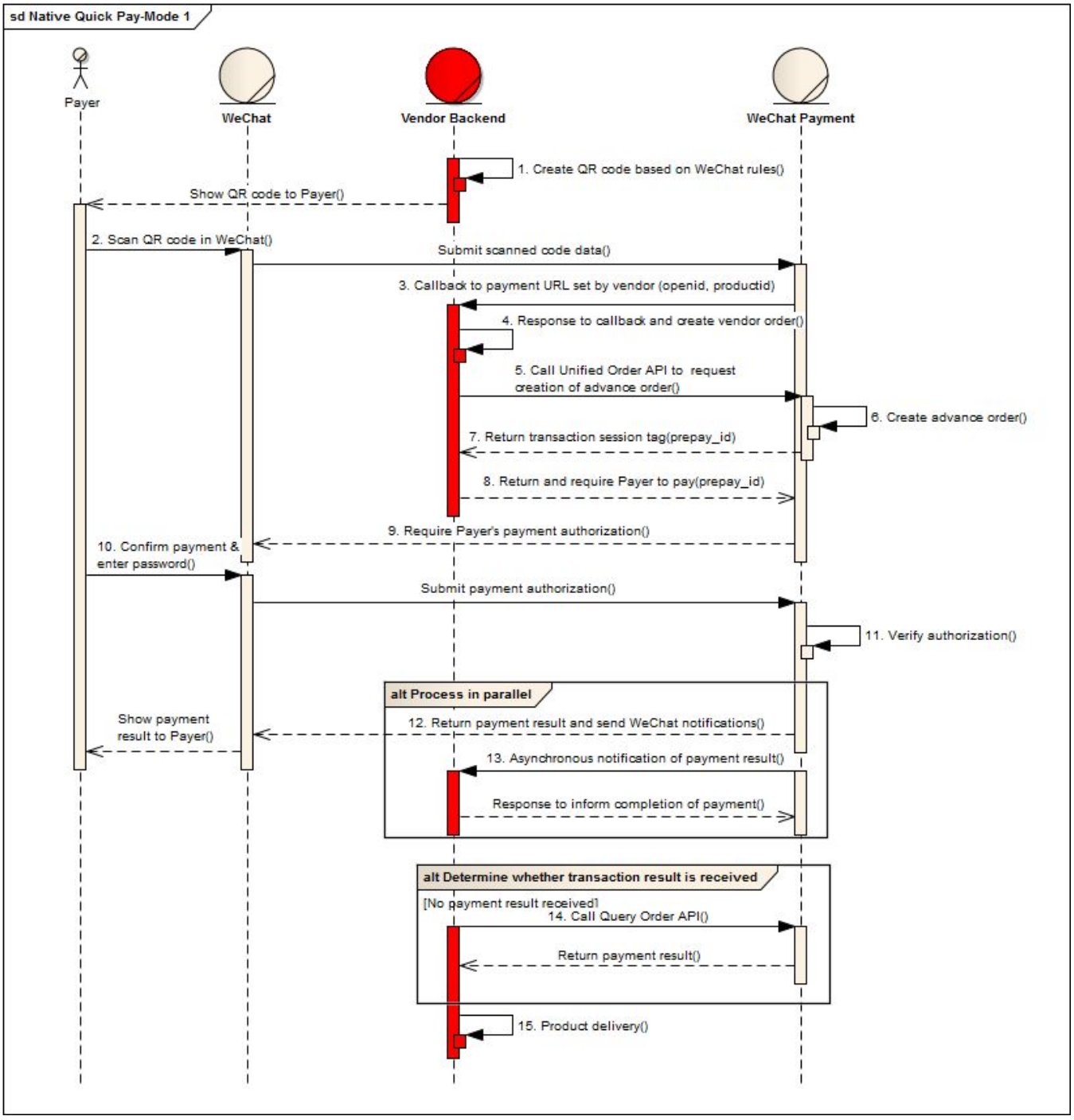

Figure 6.8 Native Payment API of Mode 1

Service steps:

1) The Vendor's backend creates a QR Code based on the format specified by WeChat payment rules (for more information, see Section 6.4.2 QR Code Generation Rules) and displays it to the Payer to be scanned.

2) Payer opens “Scan QR Code” in WeChat to scan the QR Code. The scanned data is sent to the WeChat payment system from the Payer's WeChat.

3) After receiving this request from the Payer’s WeChat, the WeChat payment system calls the payment callback URL on the Vendor's backend, containing productid and the user’s OpenID. The Vendor's system is required to return prepay_id (【Unified Order API】 returns the transaction session tag which is valid for up to 2 hours) before the Payer can make a payment.

4) Vendor's backend receives the callback request from the WeChat payment system and creates an order in the Vendor's backend based on productid.

5) Vendor's backend calls the WeChat payment system 【Unified Order API】 and requests a transaction session tag.

6) WeChat payment system creates an advance transaction based on the above request and returns the transaction session tag prepay_id.

7) Vendor's backend receives the transaction session tag prepay_id.

8) Vendor's backend returns prepay_id to WeChat payment system.

9) WeChat payment system initiates an authorization for payment on the Payer’s WeChat based on prepay_id.

10) The Payer enters their payment password and confirms payment on WeChat. The payment authorization is submitted from WeChat to the WeChat payment system.

11) WeChat payment system verifies and makes deductions to complete the transaction.

12) WeChat payment system returns the transaction result to the Payer's WeChat via SMS and prompts after the payment is done. The Payer can see the payment result in WeChat.

13) WeChat payment system sends an asynchronous message to inform the Vendor's backend of the transaction result. The Vendor's backend replies to the WeChat payment system that the payment is completed.

14) The Vendor backend polls the 【Query Order API】 if no payment message is received.

15) The Vendor confirms the order and delivers products to the Payer.

2 QR Code Generation Rules

The content of a QR Code is a URL in the following format:

weixin://wxpay/bizpayurl?**sign**=XXXXX&**appid**=XXXXX&**mch_id**=XXXXX&**product_id**=XXXXXX&**time_stamp**=XXXXXX&**nonce_str**=XXXXX

where XXXXX are the required fields for the Vendor. The Vendor should create a QR Code based on these rules. This format is required if the QR Code is to be displayed properly. A Vendor can call a third plug-in to create the QR Code. The following are parameter specifications.

Table 6.1 Parameters for generating QR Code

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx888888888888888	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Timestamp	time_stamp	Yes	String(10)	1414488825	Specifies the current system time. For more information, see Time Protocol in Section 4.2 Parameter Specifications.
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm.
Product ID	product_id	Yes	String(32)	88888	Specifies product ID or order ID defined by the Vendor
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.

Example：

weixin://wxpay/bizpayurl?appid=wx2421b1c4370ec43b&mch_id=10000100&nonce_str=f6808210402125e30663234f94c87a8c&product_id=1&time_stamp=1415949957&sign=512F68131DD251DA4A45DA79CC7EFE9D

3 Payment Callback URL

The Vendor should provide a payment callback URL (for more information, see Section 6.3.1 Enable Payment Authorization). The URL is used to receive data sent by the WeChat payment system after the Payer scans the QR Code. The Vendor's backend then creates a payment order based on the data, and calls the 【Unified Order API】 to submit the payment transaction.

1. Input Parameters

Table 6.2 Input Parameters

Field Name	ID	Required	Type	Example	Description

Official Account ID	appid	Yes	String (32)	wx888888888888888	Specifies Official Account ID assigned by WeChat
User Tag	openid	Yes	String (128)	o8GeHuLAsgefS_80exEr1cTqekUs	Specifies the user id of the Payer provided by the WeChat system in OpenID format and is a unique tag unique to each appid instance
Vendor ID	mch_id	Yes	String (32)	1900000109	Specifies vendor ID assigned by WeChat Payment
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’.
Random String	nonce_str	Yes	String (32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer or fewer. For more information, see Section 4.3.2 Random String Algorithm.
Product ID	product_id	Yes	String (32)	88888	Specifies product ID or order ID defined by the Vendor
Signature	sign	Yes	String (32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.

2. Output Parameters

Table 6.3 Output Parameters

Field Name	ID	Required	Type	Example	Description
Return Status Code	return_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL. Specifies communicating label but not 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 Signature failure Parameter format checking error
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
Advance Transaction ID	prepay_id	Yes	String(64)	wx201410272009395522657a690389285100	Specifies the advance transaction ID created by calling the Unified Order API.
Service Result	result_code	Yes	String(16)	SUCCESS	Set to SUCCESS or FAIL
Error Description	err_code_des	No	String(128)		When result_code is FAIL, the Vendor should display the error information to the Payer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.

5. Mode 2

Compared with Mode 1, Mode 2 is a simpler process. It doesn't require use of the callback payment URL. In Mode 2, the Vendor's backend calls the Unified Order API and the WeChat payment system returns code_url to the Vendor's backend to create a QR Code. The Payer scans the QR Code in WeChat to initiate payment.

Note: code_url is valid for up to 2 hours. The Payer can't scan the QR Code to pay again once it has expired.

1 Sequence Chart

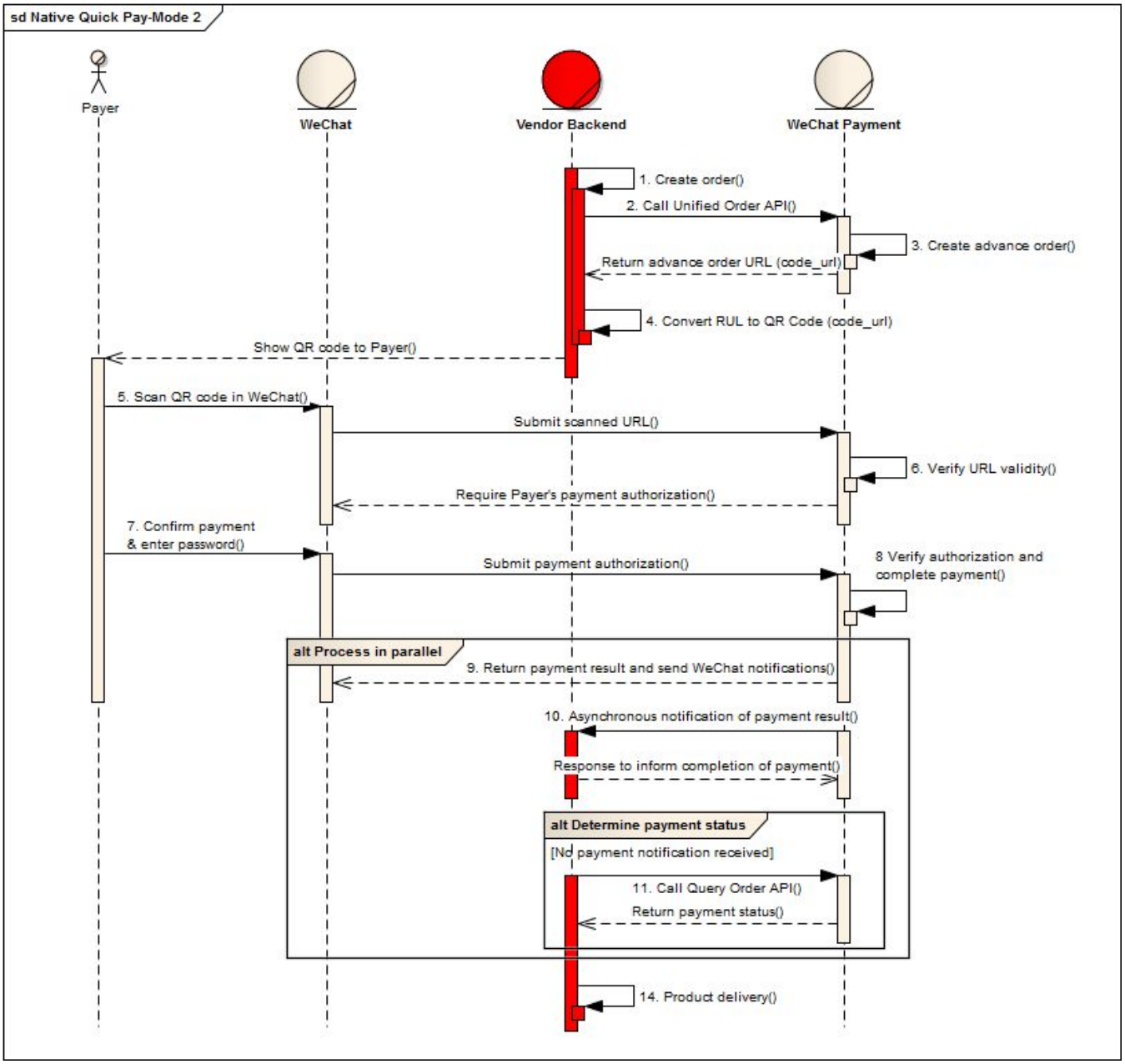

Figure 6.9 Native Payment API of Mode 2

Service steps:

1) The Vendor's backend creates an order based on the product selected by the Payer.

2) The Payer confirms payment and the Vendor calls the WeChat payment system 【Unified Order API】 to create an advance payment transaction.

3) WeChat payment system creates an advance transaction bill upon receiving this request, and returns code_url.

4) The Vendor’s backend creates a QR Code based on code_url.

5) The Payer opens “Scan QR Code” in WeChat and scans the QR code. The scanned data is sent to the WeChat payment system from the Payer's WeChat.

6) The WeChat payment system receives the request from WeChat and verifies the URL. If the URL is verified to be valid, payment is initiated and requires the Payer's authorization.

7) The Payer enters their payment password and confirms payment in WeChat. The payment authorization is submitted to the WeChat payment system from WeChat.

8) The WeChat payment system completes the transaction based on the Payer's authorization.

9) The WeChat payment system returns the transaction result to the Payer's WeChat via SMS after the payment is done. The Payer can view the payment result in their WeChat.

10) The WeChat payment system sends an asynchronous message to inform the Vendor's backend of the payment result. Then Vendor's backend replies to inform the WeChat payment system the payment is completed.

11) The Vendor’s backend polls the 【Query Order API】 if no payment message is received.

12) The Vendor confirms the order and delivers products to the Payer.

2 QR Code Generation Rules

The URL should follow the format weixin://wxpay/bizpayurl?sr=XXXXX. The Vendor can use a third plug-in to create a QR Code from code_url. If printed on the checkout bill, this QR Code is more readable as the URL is shorter.

Take the QR Code created from weixin://wxpay/s/An4baqw as an example. We can see a clearer pattern in Figure 6.10.

Figure 6.10 QR Code Example in Native Payment API using Mode 2

6. **Unified Order API**

NATIVE is required for the field of transaction type (trade_type)

For more information, see 【Section 9.1 Unified Order】.

7. **Query Order API**

For more information, see 【Section 9.2 Query Order】.

8. **Close Order API**

For more information, see 【Section 9.3 Close Order】.

9. **Submit Refund API**

For more information, see 【Section 9.4 Submit Refund】.

10. **Query Order API**

For more information, see 【Section 9.5 Query Refund】.

11. **Download Transaction History API**

For more information, see 【Section 9.6 Download Transaction History】

12. **Report Speed Testing**

To improve our payment service, we recommend vendors report payment interface delays to the WeChat payment backend via the Report Speed Testing API. For more information, see 【Section 9.8, Report Speed Testing API】.

13. **Learn More about QR Codes**

Relevant links:

http://www.thonky.com/qr-code-tutorial/

http://coolshell.cn/articles/10590.html

7. OFFICIAL ACCOUNT PAYMENT

1. **Use Case**

The Payer opens the vendor's HTML5 page in WeChat by clicking messages or scanning a related QR Code, and enters WeChat payment to complete their transaction.

Step 1: The Vendor sends a rich media message or a self-defined menu to the Payer, as shown in Figure 7.1.

Step 2: The Payer clicks the message or selects a menu option to enter the Vendor's page and selects product(s).

Figure 7.1 Vendor's Rich Media Message and Self-Defined Menu

Figure 7.2 Vendor's Product Page

Step 3: The Payer calls WeChat payment and enters their payment password as prompted, as shown in Figure 7.3.

Step 4: The payment is completed after the password has been verified successfully. The Vendor's backend receives the payment result notification, as shown in Figure 7.4.

Figure 7.3: Enter Payment Password

Figure 7.4 Prompt after Successful Payment

Step 5: The Vendor returns a page displaying that the payment was successful. This page is designed by the Vendor, as shown in Figure 7.5.

Step 6: The Vendor's official account sends a message to inform the Payer in WeChat of delivery, as shown in Figure 7.6. This step is optional.

Figure 7.5 Return Vendor's Page

Figure 7.6 Inform Payer of Delivery

Notes: Vendors can convert their product URL into a QR Code so that payers can scan them to buy and pay quickly.

Detailed steps are as follows:

Please carefully read the instructions below before designing product pages:

(1) The Payer opens the Vendor's product page and confirms a transaction. The getBrandWCPayRequest API is called via JavaScript on the page to initiate a WeChat payment request. This initiates the Payer's payment process.

(2) After the Payer clicks to pay successfully and completes the transaction, the Vendor's front-end receives a value returned within the JavaScript. The Vendor can then directly redirect to a static page indicating successful payment.

(3) The Vendor's backend receives a callback call from the WeChat Open Platform, indicating a result reporting the transaction was paid successfully.

Note:The triggering times of steps (2) and (3) are not strictly in order. The returned value of the JSAPI is the event that should trigger the Vendor's page redirect. However, the Vendor's backend should process the payment result data only after a successful payment call is received from the WeChat payment system.

2. Participating Vendors

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

Offline: New-version Ubox vending machines, etc.

Online: JD.com, Yixun Official Account, etc.

3. Programing Instructions

1 Set Testing Directory

You can configure a testing directory on the WeChat Official Account Admin Platform, as shown in Figure 7.7. In the "Payment Testing" section, you can configure a testing directory in the "Testing Directory (测试目录)" field and add a testers' WeChat IDs to the white list. Ensure that the testing directory is matches the directory for initiating payment, otherwise the payment can't be performed later. In addition, this payment URL shall be sent to the corresponding official account session so as to start the payment testing properly.

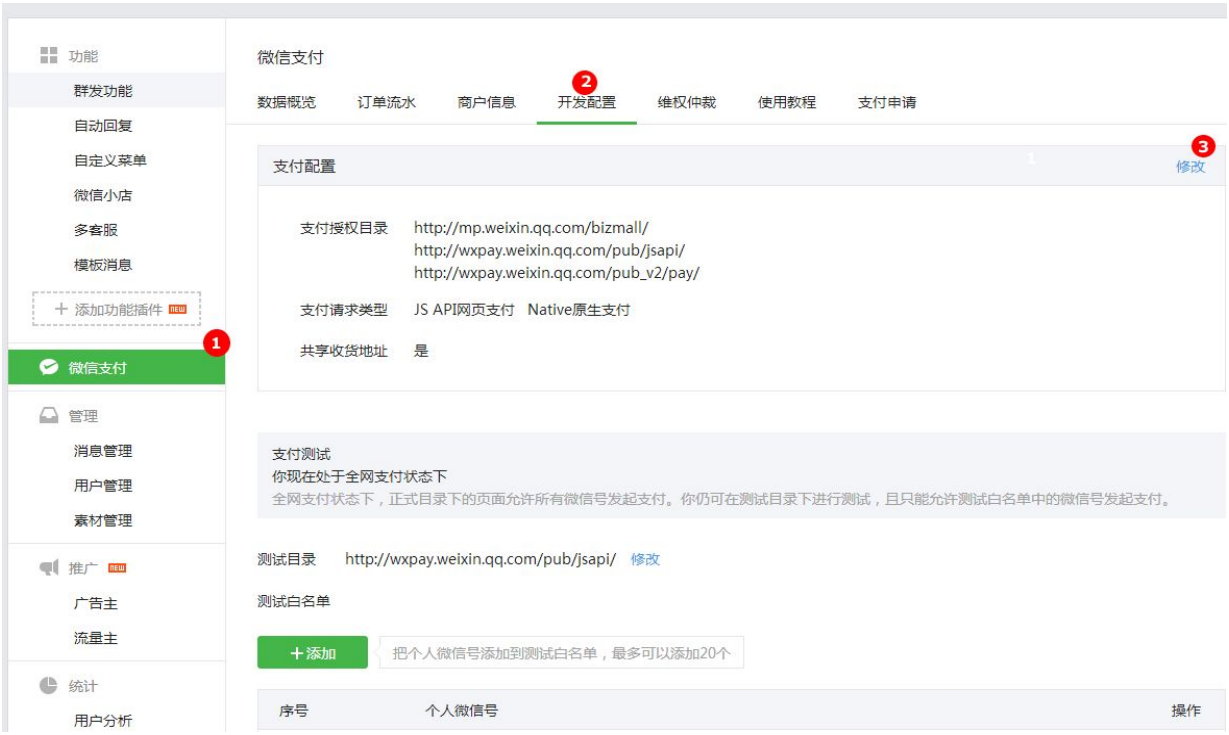

Figure 7.7 Official Account Payment Settings

2 Set Official Payment Directory

You can edit the above settings after configurations, as shown in Figure 7.8. You can enable the JSAPI by checking the checkbox and setting the payment authorization directory. Ensure that the directory matches the directory for initiating payment, as payment can't be started from a sub-directory.

Figure 7.8 Official Account Payment Authorization Directory Settings

4. Sequence Chart

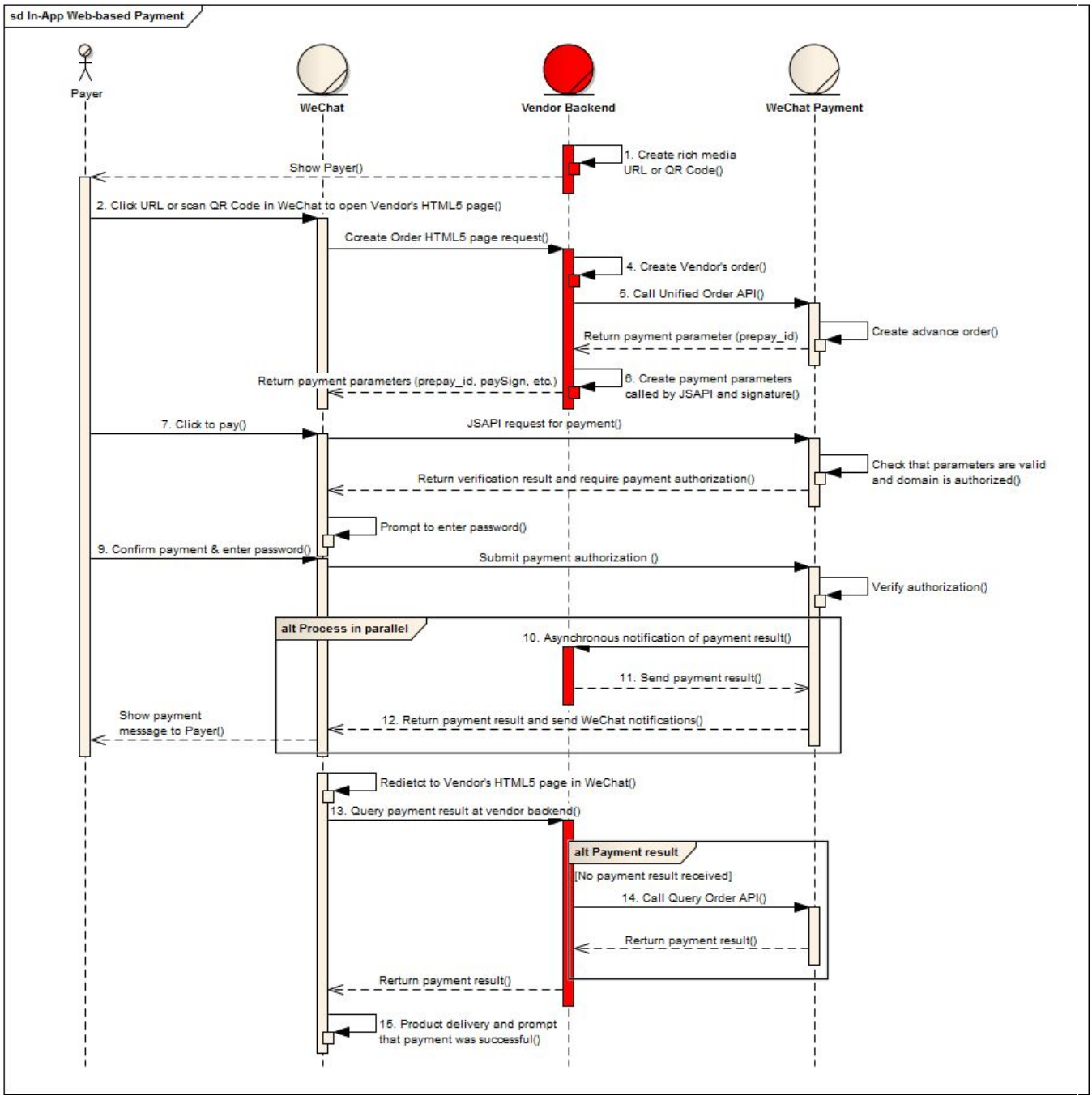

Figure 7.9 Official Account Payment Sequence Chart

How the Vendor's system interacts with the WeChat payment system:

1. The Vendor's server calls the Unified Order API to create an order. For more information, see 【Section 9.1 Unified Order.】
2. The Vendor's server receives payment notifications. For more information, see 【Section 9.7 General Payment Result Notification.】

3. The Vendor's server queries payment result. For more information, see 【Section 9.2 Query Order.】

5. WeChat Version Requirement

WeChat supports the payment feature in versions 5.0 and greater, so payers using versions prior to version 5.0 can't access WeChat payment. As a consequence, we suggest the Vendor confirm the Payer's version via the user agent string before using WeChat payment features. Using an iPhone WeChat client as an example, the Vendor can check the Payer's WeChat version via user agent below:

"Mozilla/5.0(iphone;CPU iphone OS 5_1_1 like Mac OS X) AppleWebKit/534.46(KHTML,like Geocko) Mobile/9B206 MicroMessenger/5.0"

where 5.0 indicates the Payer's WeChat version. Vendors can parse the above HTTP header to check whether the WeChat version is greater than or equal to 5.0.

6. WeChat Security Payment Subtitle in Page Header

For transaction security reasons, the Payer can see WeChat Security Payment subtitle in the page header of their payment page in WeChat. We suggest vendors use this subtitle for their WeChat payment transaction.

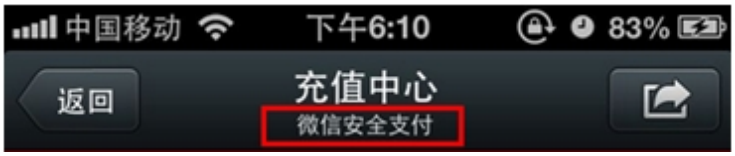

Figure 7.10 WeChat Security Payment Subtitle

To display the subtitle, the Vendors should add "showwxpaytitle=1" to the end of the original URL. After doing so, their page will show the "WeChat Security Payment" subtitle. For example, for the original URL (http://weixin.qq.com), the Vendor can change it to the one with the subtitle (http://weixin.qq.com?showwxpaytitle=1).

As a result, when the Payer opens http://weixin.qq.com in WeChat, they won't see the subtitle. However, if they go with http://weixin.qq.com?showwxpaytitle=1, the subtitle will be displayed in the header of their opened page.

7. Call Payment API from HTML5 Websites

The JSAPI is called when the Payer opens an HTML5 website with a WeChat-supported browser. The input and output API data is in JSON format.

Notes: WeixinJSBridge built-in objects are invalid in other browsers and parameter names in the list are case-sensitive.

For "getBrandWCPayRequest" parameters and return value definitions, see Table 7.1. For return value description, see Table 7.2.

Table 7.1 HTML5-based API Parameters

Field Name	ID	Required	Type	Example	Description
Official Account ID	appId	Yes	String(16)	wx8888888888888888	This ID is issued after vendors apply for official account supporting WeChat payment.
Timestamp	timeStamp	Yes	String(32)	1414561699	Specifies the current time. For more information, see Timestamp in Section 4.2 Parameter Specifications.
Random String	nonceStr	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm.
Order Extension String	package	Yes	String(128)	prepay_id=123456789	Specifies the parameter value (prepay_id) returned by the Unified Order API. The submission format is "prepay_id=***".
Signature	paySign	Yes	String(64)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	signType	Yes	String(32)	MD5	Currently HMAC-SHA256 and MD5 are supported

Table 7.2 Web-based Payment API Return Data Description

Return Value	Description
get_brand_wcpay_request:ok	Payment successful
get_brand_wcpay_request:cancel	Payment canceled
get_brand_wcpay_request:fail	Payment failed

"get_brand_wcpay_request:ok" is returned by the JSAPI only when the Payer completes the payment. For front-end logic, "get_brand_wcpay_request:cancel" or "get_brand_wcpay_request:fail" can be handled as payment exceptions.

Example:

```
function onBridgeReady(){
   WeixinJSBridge.invoke(
      'getBrandWCPayRequest', {
          "appId" : "wx2421b1c4370ec43b",     //Official Account name transferred by vendors
```

```
            "timeStamp":" 1395712654",          //Timestamp since 1.1.1970 UTC
            "nonceStr" : "e61463f8efa94090b1f366cccfbbb444", //Random string
            "package" : "prepay_id=u802345jgfjsdfgsdg888",
            "signType" : "MD5",          //WeChat signature type:
            "paySign" : "70EA570631E4BB79628FBCA90534C63FF7FADD89" //WeChat signature
        },
        function(res){
            if(res.err_msg == "get_brand_wcpay_request:ok" ) {}      // Use the above method to determine values returned to the front-end. Please note that "res.err_msg" returns"ok" after a successful payment. However, WeChat doesn't guarantee its reliability.
        }
    );
}
if (typeof WeixinJSBridge == "undefined"){
    if( document.addEventListener ){
        document.addEventListener('WeixinJSBridgeReady', onBridgeReady, false);
    }else if (document.attachEvent){
        document.attachEvent('WeixinJSBridgeReady', onBridgeReady);
        document.attachEvent('onWeixinJSBridgeReady', onBridgeReady);
    }
}else{
    onBridgeReady();
}
```

8. Unified Order API

JSAPI is required for the field of transaction type (trade_type)

For more information, see 【Section 9.1 Unified Order】.

9. Query Order API

For more information, see 【Section 9.2Query Order】.

10. Close Order API

For more information, see 【Section 9.3 Close Order】.

11. Submit Refund API

For more information, see 【Section 9.4 Submit Refund】.

12. Query Refund API

For more information, see 【Section 9.5 Query Refund】.

13. Download Transaction HistoryTransaction History API

For more information, see 【Section 9.6 Download Transaction History】.

8. IN-APP PAYMENT

1. Use Case

This method is applicable to WeChat payments integrated by vendors into mobile apps.

The Vendor's app calls the SDK provided by WeChat to use the WeChat payment module, and redirects to WeChat to pay a transaction. After completing the transaction, the WeChat reopens the vendor's app and a page containing the payment result is displayed.

At present, WeChat supports iOS, Android and Windows Phone.

Detailed steps are as below:

Step 1: Payer enters the Vendor's app, selects products and confirms the transaction to proceed with payment. The Vendor’s service backend creates a payment order and signs it, and relevant data is transferred to the Vendor's app, as shown in Figure 8.1.

Step 2: The Payer clicks to confirm the payment and which opens the payment page within WeChat to pay for the order, as shown in Figure 8.2.

Step 3: The Payer confirms the payee and the amount, and clicks to pay. A page is then displayed that prompts the Payer to enter their payment password. The Payer can select to pay with a bank card or via Balance, as shown in Figure 8.3.

Figure 8.1 Example of Vendor's app screen

Figure 8.2 Redirect to the payment page within WeChat

Figure 8.3: Payer enters their payment password

Step 4: The Payer enters their payment password to complete the transaction. If paid successfully, a page containing the payment result is displayed on the payer's WeChat, as shown in Figure 8.4.

Step 5: The page reopens the Vendor's app, which will show the order process result based on payment results.

Figure 8.4 Prompt after successful payment

Figure 8.5 Prompt for reopening the vendor's app

2. Participating Vendors

This payment method is now supported on the JD and Yixun apps.

3. Service process

This payment method is explained below. The Unified Order API, Query Order API, and accepting order notifications require signatures, which are created on the Vendor's service backend, as shown in Figure 8.6.

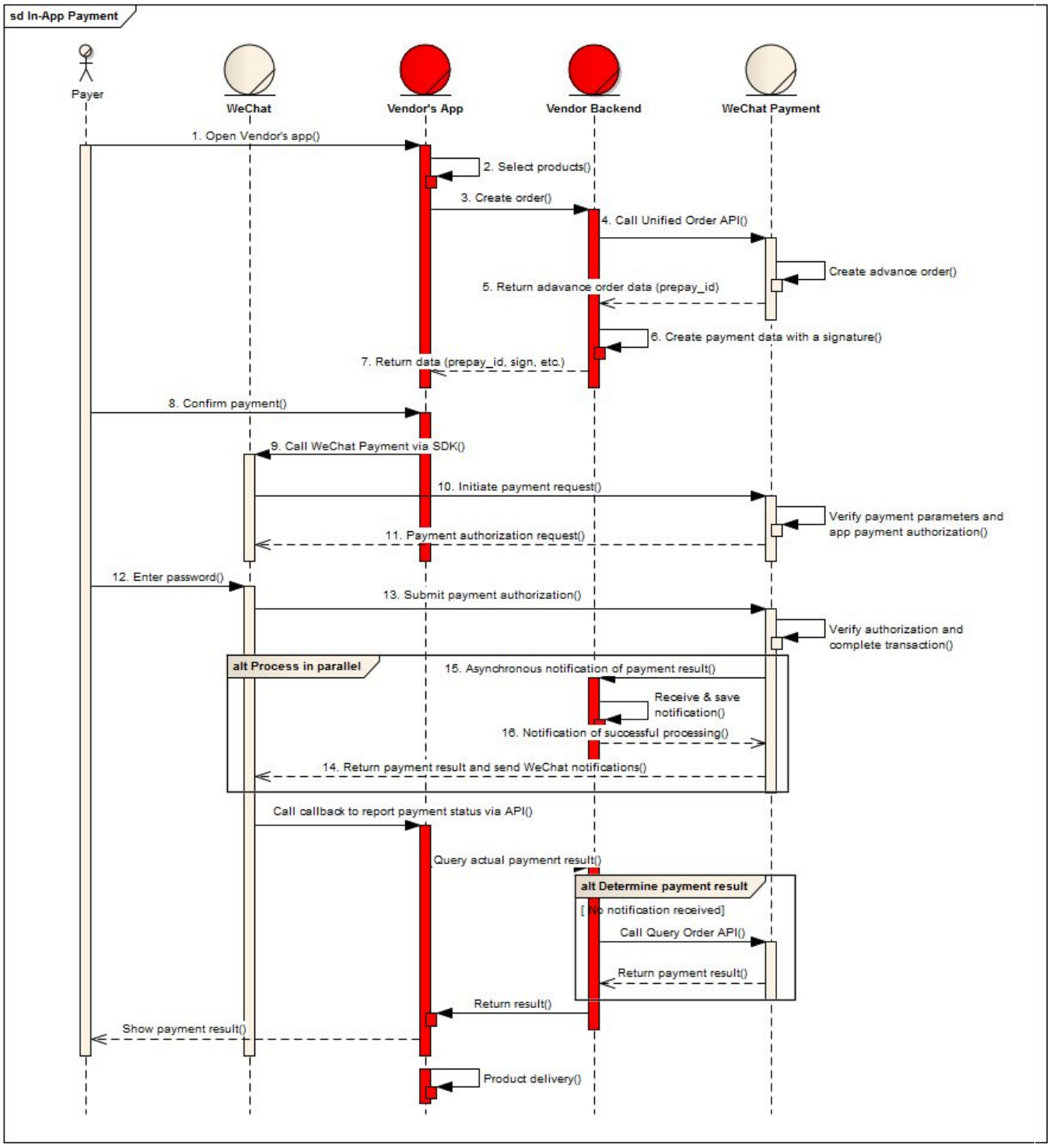

Figure 8.6 In-app payment sequence chart

Below shows how the vendor's backend interacts with the WeChat payment system:

Step 1: Payer selects products within the Vendor's app, submits the order, and chooses WeChat payment.

Step 2: The Vendor receives payer's payment transaction and calls the Unified Order API. For more information, see Section 9.1【Unified Order】.

Step 3: The Unified Order API returns a normal prepay_id, and creates a signature based on the

signature rules. The relevant data is transferred to the Vendor's app. Fields contained in the signature include cappId, partnerId, prepayId, nonceStr, timeStamp, and package.

Note: the value format of package is Sign=WXPay

Step 4: The Vendor's app uses the SDK to open WeChat payment within WeChat. For more information, see Section 8.5【App-based Development Guide】.

Step 5: The Vendor's backend receives payment notifications. For more information, see Section 9.7【General Payment Result Notification】.

Step 6: The Vendor's backend queries the payment result. For more information, see Section 9.2【Query Order】.

4. Learn More about this API

https://open.weixin.qq.com/zh_CN/htmledition/res/dev/document/sdk/ios/index.html

5. Call Payment API from APP

About the detailed App-based development process, please refer to Chapter 8.6.
Call Payment API form APP

Field Name	ID	Required	Type	Example	Description
APP Application ID	appid	Yes	String(32)	wx8888888888888888	This ID is issued after vendors have register their APP on WeChat open platform.
Vendor ID	partnerid	Yes	String(32)	1900000109	Specifies vendor ID assigned by WeChat Payment
Prepaid Trading ID	prepayid	Yes	String(32)	WX1217752501201407033233368018	Specifies the parameter value (prepay_id) returned by the Unified Order API.
Order Extension String	package	Yes	String(128)	Sign=WXPay	Specify as the static value “Sign=WXPay”
Random String	noncestr	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm.
Time stamp	timestamp	Yes	String(10)	1414561699	Specifies the current time. For more information, see Timestamp in Section 4.2 Parameter Specifications.

Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.

Data returned：

Return Value	Description	Solution
0	Payment successful	Show the payment successful page
-1	Payment error	Possible reasons: sign error, unregistered appid, incorrect project appid, registered appid not match to the set one, etc.
-2	Payment cancelled	No need to settle. User gives up the payment, etc.

6. App-based Development Guide

1 Instructions for iOS

We will use Xcode10.0 running an iOS 7.0 environment as an example to illustrate the process.

1) APPID for Project Settings

After the Vendor has successfully applied for an App in the WeChat Open Platform, the Platform will provide an unique APPID to the Vendor. When creating a project in Xcode, the developer should enter the APPID value in the “URL Schemes” field, as marked in red in Figure 8.7.

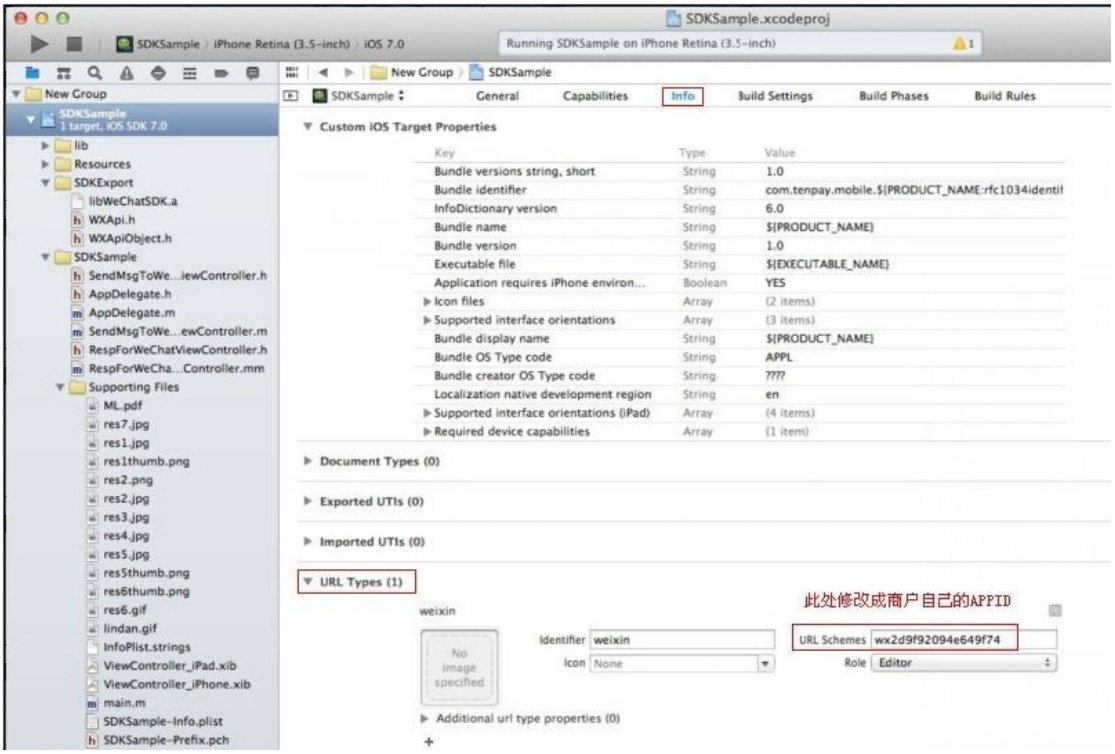

Figure 8.7

2) APPID for Registration

The WeChat SDK “lib” and “head” files should be imported into the Xcode project. Before calling the API, you should register your APPID with WeChat, as shown below:

```
[WXApi registerApp:@"wxd930ea5d5a258f4f" withDescription:@"demo 2.0"];
```

3) Call Payment

The Vendor's server calls the Unified Order API (for more information, see Section 9.1 Unified Order) to create an advance transaction. After obtaining prepay_id and signing relevant parameters, the advance transaction data is transferred to the App to start a payment. See below for an example on how to do this:

```
PayReq *request = [[[PayReq alloc] init] autorelease];
request.partnerId = @"10000100";
request.prepayId= @"1101000000140415649af9fc314aa427";
request.package = @"Sign=WXPay";
request.nonceStr= @"a462b76e7436e98e0ed6e13c64b4fd1c";
request.timeStamp= @"1397527777";
request.sign= @"582282d72dd2b03ad892830965f428cb16e7a256";
[WXApi safeSendReq:request];
```

4) Payment Result Callback

As shown in the sample included in the SDK, onResp() can be added to WXPayEntryActivity. After completing payment, WeChat will be redirected to the Vendor's app and do a callback using onResp(). The Developer receives notifications in this function and determines the returned error code if necessary. If the payment is successful, the payment result shall be queried from the WeChat payment system and shown to

Payer. The payment result is subject to the payment notifications from the WeChat payment system and the result returned to the Payer after querying the API. See below for an example on how to do this:

```
- (void)onResp:(BaseResp *)resp {
    if ([resp isKindOfClass:[PayResp class]]) {
        PayResp *response = (PayResp *)resp;
        switch (response.errCode) {
            case WXSuccess:
                //Prompt of successful payment according to server-based query result or API returned data
                NSlog(@"Payment Successful");
                break;
            default:
                NSlog(@"Payment Failed，  retcode=%d",resp.errCode);
                break;
            }
    }
}
```

errCode value list:

Name	Description	Solution
0	Success	Displays the success page
-1	Error	This may be caused by signature error, unregistered APPID, incorrect APPID in project settings, unmatched APPID in the registration and project settings, or other exceptions.
-2	Canceled by user	This occurs when the Payer cancels payment and returns to the App. In this case, no further steps are required.

2 Instructions for Android

1) Backend Settings

After the Vendor has successfully applied for an App in the WeChat Open Platform, the Platform will provide an unique APPID to the Vendor. For payment security reasons, the Vendor's app package name and signature must match in the Platform. Only when these are configured properly can payment be initiated. This can be done in “App Platform (应用平台)”->”Android App (Android 应用)” on the WeChat Open Platform, as marked in red in Figure 8.8.

Figure 8.8

The App package name is the same as the one set in the configuration file "AndroidManifest.xml" of the app's project settings. For an example, look at the package name "net.sourceforge.simcpux" in Figure 8.9.

The App signature is the keystore used for compiling according to project's app package name, which should be a 32-bit md5 string generated by a signature tool. If developers install the signature tool on a testing phone, they can run it to generate the app's signature string, as shown in the strings in green in Figure 8.9. Download a signature tool at the following URL:
https://open.weixin.qq.com/zh_CN/htmledition/res/dev/download/sdk/Gen_Signature_Android.apk

Figure 8.9

2) APPID for Registration

The WeChat JAR package should be imported into the App project. Before calling the API, you need to register your APPID with WeChat, as shown below:

Before calling the API, you should register your APPID with WeChat, as shown below:

```
final IWXAPI msgApi = WXAPIFactory.createWXAPI(context, null);
```

// Register this app with WeChat

```
msgApi.registerApp("wxd930ea5d5a258f4f");
```

3) Call Payment

The Vendor's server calls the Unified Order API (for more information, see Section 9.1 Unified Order) to create an advance transaction. After obtaining prepay_id and signing relevant parameters, the advance transaction data is transferred to the App to start a payment. See below for an example of how to do this:

```
IWXAPI api;
PayReq request = new PayReq();
request. appId = "wxd930ea5d5a258f4f";
request. partnerId = "1900000109";
request.prepayId= "1101000000140415649af9fc314aa427",;
request. packageValue = "Sign=WXPay";
request.nonceStr= "1101000000140429eb40476f8896f4c9";
request.timeStamp= "1398746574";
request.sign= "7ffecb600d7157c5aa49810d2d8f28bc2811827b";
api.sendReq(req);
```

4) Payment Result Callback

As shown in the sample included in the SDK, onResp() can be added to WXPayEntryActivity. After completing payment, WeChat will be redirected to Vendor's app and do a callback using onResp(). The Developer receives notifications in this function and determines the returned error code if necessary. If the

payment is successful, the payment result shall be queried from the WeChat payment system and shown to Payer. The payment result is subject to the payment notifications from the WeChat payment system and the result returned to the Payer after querying the API. See below for an example on how to do so:

```
public void onResp(BaseResp resp) {
        if (resp.getType() == ConstantsAPI.COMMAND_PAY_BY_WX) {
                Log.d(TAG, "onPayFinish, errCode = " + resp.errCode);
                AlertDialog.Builder builder = new AlertDialog.Builder(this);
                builder.setTitle(R.string.app_tip);
        }
}
```

errCode value list:

Name	Description	Solution
0	Success	Displays the success page
-1	Error	This may be caused by signature error, unregistered APPID, incorrect APPID in project settings, unmatched APPID in the registration and project settings, or other exceptions.
-2	Canceled by users	This happens occurs when the Payer cancels payment and returns to App. In this case, no further steps are required.

7. Unified Order API

APP is required for the field of transaction type (trade_type)

For more information, see 【Section 9.1 Unfied Order】.

8. Query Order API

For more information, see 【Section 9.2 Query Order】.

9. Close Order API

For more information, see 【Section 9.3 Close Order】.

10. Submit Refund API

For more information, see 【Section 9.4 Submit Refund】.

11. Query Refund API

For more information, see 【Section 9.5 Query Refund】.

12. Download Transaction History API

For more information, see 【Section 9.6 Download Transaction History】.

9. 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. Unified Order

1 Use Case

For scenarios other than the Quick Pay method, the Vendor's backend calls this API to create an advance transaction in the WeChat payment service backend, and initiates the payment process via payment by QR Code. JSAPI, App and other payment methods after the order is submitted successfully.

2 URL

URL: https://api.mch.weixin.qq.com/pay/unifiedorder

3 Certificate Requirement

No certificate is required.

4 Request Parameters

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx888888888888888	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 is defined by the Vendor. *Note: If the payment is performed based on PC web page or WeChat Web-based page, please submit the field value as WEB.*
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)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	NO	String(32)	HMAC-SHA256	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This field is only required when sign_type is HMAC-SHA256.
Item Description	body	Yes	String(128)	iPad Mini in white with 16G memory	Short description of item(s) to be purchased for the Payer, Please refer to Section 4.2, item 6
Item Details	detail	No	String(6000)	{ "goods_detail":[{ "goods_id":"iphone6s_16G", "wxpay_goods_id":"1001", "goods_name":"iPhone6s 16G", "goods_num":1, "price":528800, "goods_category":"123456", "body":"苹果手机" }, { "goods_id":"iphone6s_32G",	Detailed product list described in JSON format. Please use CDATA tag to protect the JSON string when generating signature. goods_detail []: └ goods_id String Required 32 Goods ID └ wxpay_goods_id String Optional 32 The unified goods ID defined by WeChat └ goods_name String Required 256 Goods name └quantity Int Required Goods amount └ price Int Required Goods price, unit as cent *Note: The goods price should be less*

				"wxpay_goods_id":"1002", "goods_name":"iPhone6s 32G", "quantity":1, "price":608800, "goods_category":"123789", "body":"苹果手机" }] }	*than total_fee and it should be the favorable price.*
Additional Data	attach	No	String(128)	Additional description	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	Yes	String(32)	1217752501201407033233368018	32 alphanumeric characters or less. For more information, see Section 4.2 Vendor's Order Number.
Currency Type	fee_type	Yes	String(16)	GBP	ISO-4217 standard compliant and be described by three characters based code. For more information, see Section 4.2 Currency Type.
Total Amount	total_fee	Yes	Int	888	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.
Terminal IP	spbill_create_ip	Yes	String(16)	8.8.8.8	For Native Quick Pay, specifies the machine IP that calls WeChat Payment API. While for Official Account Payment or In-App Payment, specifies the client terminal IP
Transaction Start Time	time_start	No	String(14)	20091225091010	Specifies the transaction creation time in the format yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10 (UTC+08). For more information, see Section4.2 Time Protocol.
Transaction	time_expire	No	String(14)	20091227091010	Specifies the transaction end time in the format yyyyMMddHHmmss, such

End Time					as 20091227091010 for Dec 27, 2009 09:10:10 (UTC+08). The shortest transaction invalid time should be larger than 5m. For more information, see Section4.2 Time Protocol.
Item Label	goods_tag	No	String(32)	WXG	Specifies the label of goods, which is a parameter in the coupon feature for businesses. For more information, see Section 10 Mobile Coupon.
Notification URL	notify_url	Yes	String(256)	http://www.baidu.com/	Specifies the callback address for receiving WeChat payment notifications
Transaction Type	trade_type	Yes	String(16)	JSAPI	Set to JSAPI, NATIVE, or APP
Product ID	product_id	No	String(32)	12235413214070356458058	This field is only required when trade_type is NATIVE. This ID contains the product ID as set by the Vendor.
Specified Payment Method	limit_pay	No	String(32)	no_credit	no_credit: Using credit card for payment is not allowed
User Tag	openid	No	String(128)	oUpF8uMuAJO_M2pxb1Q9zNjWeS6o	This field is only required when trade_type is JSAPI. It is the only user identification under the current appid. About how to get the openid, please refer to http://admin.wechat.com/wiki/index.php?title=User_Profile_via_Web

Example：

```
<xml>
   <appid>wx2421b1c4370ec43b</appid>
   <attach>Payment Testing</attach>
   <body>JSAPI Payment Testing</body>
   <mch_id>10000100</mch_id>
   <nonce_str>1add1a30ac87aa2db72f57a2375d8fec</nonce_str>
   <notify_url>http://wxpay.weixin.qq.com/pub_v2/pay/notify.v2.php</notify_url>
   <openid>oUpF8uMuAJO_M2pxb1Q9zNjWeS6o</openid>
   <out_trade_no>1415659990</out_trade_no>
   <spbill_create_ip>14.23.150.211</spbill_create_ip>
   <total_fee>1</total_fee>
   <trade_type>JSAPI</trade_type>
   <sign>0CB01533B8C1EF103065174F50BCA001</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	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, 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	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface
Device ID	device_info	No	String(32)	013467007045764	The Device ID submitted when calling the interface
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Signature returned. 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 both return_code and result_code are SUCCESS, return data will also include the following fields:

Field Name	ID	Required	Type	Example	Description
Transaction Type	trade_type	Yes	String(16)	JSAPI	The transaction type submitted. The value could be JSAPI, NATIVE, or APP
Advance Transaction ID	prepay_id	Yes	String(64)	wx201410272009395522657a690389285100	Specifies the advance transaction ID created by WeChat. It is used to call the Query Order API later. Validity is 2 hours.
QR Code URL	code_url	No	String(64)	URI： weixin://wxpay/s/An4baqw	This field is returned when trade_type is NATIVE. This parameter should be used to create a QR Code that is displayed to the Payer later.

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[IITRi8Iabbblz1Jc]]></nonce_str>
  <sign><![CDATA[7921E432F65EB8ED0CE9755F0E86D72F]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <prepay_id><![CDATA[wx201411101639507cbf6ffd8b0779950874]]></prepay_id>
  <trade_type><![CDATA[JSAPI]]></trade_type>
</xml>
```

6 Error Code

Name	Description	Reason	Solution
NOAUTH	Vendor doesn't have permission to use this API	Vendors hasn't enabled authorization for this API	The Vendor should apply for permission to use this API
NOTENOUGH	Insufficient balance	The Payer has an insufficient balance in their payment card	Inform the Payer to add funds to their account or to try another payment card
ORDERPAID	Order is paid	Order is already paid and cannot be paid for again	The order has already been paid and no further action is required
ORDERCLOSED	Order is closed	The current order is closed and cannot be paid for	The current order has already been closed. The Payer should be told to create a new

		again	order.
SYSTEMERROR	System error	System has timed out	A system exception has occurred. Call the API again using the same parameters.
APPID_NOT_EXIST	APPID DOES NOT EXIST	Provided APPID in this parameter does not exist	Check whether provided APPID is correct
MCHID_NOT_EXIST	MCHID DOES NOT EXIST	Provided MCHID in this parameter does not exist	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
LACK_PARAMS	Missing parameter	Required parameter is missing	Check whether parameter is provided
OUT_TRADE_NO_USED	Duplicate vendor order number	The same transaction can't be submitted repeatedly	Check whether vendor's order number has already been submitted or used previously
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
REQUIRE_POST_METHOD	Use post method	Data is not transferred by post	Check whether data is submitted via POST method
POST_DATA_EMPTY	post data is empty	post data can’t be empty	Check whether post data is empty
NOT_UTF8	Invalid coding format	Specified coding format is not used	Use NOT_UTF8 coding format

2. **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	Choose one to submit	String(32)	1346700704 5764	WeChat order number is preferred
Vendor Order Number	out_trade_no		String(32)	12177525012 01407033233 368018	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)	C380BEC2B FD727A4B68 45133519F3 AD6	32 characters or fewer. For more information, see Section 4.3.2 Random String Algorithm
Signature	sign	Yes	String(32)	5K8264ILTK CH16CQ250 2SI8ZNMTM 67VS	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	No	String(32)	HMAC-SHA2 56	Currently support HMAC-SHA256 and MD5, default is MD5. This field is only required when sign_type is HMAC-SHA256

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 of trade_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	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 characters or fewer
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Signature returned. 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 both return_code and result_code are SUCCESS, return data will also include the following fields:

Note: If trade_state is not SUCCESS, only out_trade_no and attach will be returned.

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 by the WeChat system in OpenID format unique to each appid instance
Follows Official Account or not	is_subscribe	No	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
Total Amount	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)	GBP	ISO-4217 standard compliant and be described by three characters based code. For more information, see Section 4.2 Currency Type.
Cash Payment	cash_fee	Yes	Int	100	Specifies the total cash payment amount of a transaction. For more information, see Section

Amount					4.2 Payment Amount.
Cash Type	cash_fee_type	No	String(16)	CNY	ISO-4217 standard compliant and be described by three characters based code. For more information, see Section 4.2 Currency Type.
Specifies the number of a WeChat payment order	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 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)	20141030133525	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)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000

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[TN55wO9Pba5yENl8]]></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.

3. Close Order

1 Use Case

This API needs to be called before a vendor wants to create a new order due to the Payer failing to pay for an order. The original order will be closed in order to avoid repeat payment. After an order is created in the WeChat payment system, if the Payer doesn't pay within the required time, no further operation can be performed in the system. To prevent the Payer from proceeding with their order, this API is called to close the order.

Note: The Close Order API might be used no shorter than 5 minutes after the order is created.

2 URL

https://api.mch.weixin.qq.com/pay/closeorder

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

Vendor Order Number	out_trade_no	Yes	String(32)	121775250120140703323336 8018	Specifies an internal order number created by the Vendor's system
Random String	nonce_str	Yes	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	32 alphanumeric characters or fewer. For more information, see Section 4.2 Vendor's Order Number.
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	NO	String(32)	HMAC-SHA256	Currently support HMAC-SHA256 and MD5, default is MD5. This field is only required when sign_type is HMAC-SHA256

Example:

```
<xml>
   <appid>wx2421b1c4370ec43b</appid>
   <mch_id>10000100</mch_id>
   <nonce_str>4ca93f17ddf3443ceabf72f26d64fe0e</nonce_str>
   <out_trade_no>1415983244</out_trade_no>
   <sign>59FF1DF214B2D279A0EA7077C54DD95D</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
Official Account ID	appid	Yes	String(32)	wx8888888888888888	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface

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.
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

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[BFK89FC6rxKCOjLX]]></nonce_str>
   <sign><![CDATA[72B321D92A7BFA0B2509F3D13C7B1631]]></sign>
   <result_code><![CDATA[SUCCESS]]></result_code>
</xml>
```

6 Error Code

Name	Description	Reason	Solution
ORDERPAID	Order is paid	Order is paid and cannot be closed	Order is already paid and cannot be closed. No further operation is required.
SYSTEMERROR	System error	System error	System exception has occurred. Call this API again.
ORDERNOTEXIST	Order does not exist	This order does not exist in the system	Don't attempt to close this order yet as it is still a pending transaction
ORDERCLOSED	Order is closed	Order is closed and cannot be done again.	Order is already closed and no further operation is required
SIGNERROR	Signature error	Incorrect signature result	Check whether signature parameter and method comply with signature algorithm requirements
REQUIRE_POST_MET	Use post	Data is not transferred by post	Check whether data is submitted by

HOD	method		POST method
XML_FORMAT_ERROR	INVALID XML FORMAT	INVALID XML FORMAT	Check whether XML parameters are in correct format

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 3 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.

Note: If the refund request is failed, please use the same out_refund_no for retrying.

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	nonce_str	Yes	String(32	5K8264ILTKC	32 characters or fewer. For more information,

String			)	H16CQ2502SI8ZNMTM67VS	seeSection 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.
Sign type	sign_type	NO	String(16)	HMAC-SHA256	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This field is only required when sign_type is HMAC-SHA256
WeChat Order Number	transaction_id	Choose one to submit	String(32)	1217752501201407033233368018	Specifies the WeChat payment order id number
Vendor Order Number	out_trade_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.
Vendor Refund Number	out_refund_no	Yes	String(32)	1217752501201407033233368018	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.
Total Amount	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.
Currency Type	refund_fee_type	No	String(8)	GBP	ISO-4217 standard compliant and be described by three characters based code. The refund currency type must be same with the bid currency type. For more information, see Section 4.2 Currency Type.
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
Result Code	result_code	Yes	String(16)	SUCCESS	SUCCESS/FAIL SUCCESS: Receives refund application successfully. To get refund status, calling Query Refund API is required. FAIL: Submitting refund application failed.
Error Code	err_code	No	String(32)	SYSTEMERROR	For more information, see Section 9.6 Download Transaction History.
Error Code Description	err_code_des	No	String(128)	System timed out	Describes result data
Official Account ID	appid	Yes	String(32)	wx888888888888888	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface
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(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
Vendor Refund Number	out_refund_no	Yes	String(32)	1217752501201407033233368018	Vendor Refund Number
WeChat Refund Number	refund_id	Yes	String(32)	1217752501201407033233368018	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	Yes	String(8)	GBP	ISO-4217 standard compliant and be described by three characters based code. The refund currency type must be same with the bid currency type. For more information, see Section 4.2 Currency Type.
Total Amount	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)	GBP	ISO-4217 standard compliant and be described by three characters based code. 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
Payment Currency Type	cash_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 Refund Amount	cash_refund_fee	Yes	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)	CNY	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type.
Exchange Rate	rate	Yes	String(16)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000

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[NfsMFbUFpdbEhPXP]]></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
USER_ACCOUNT_ABNORMAL	Refund request failed	User's account cancelled	This error code means refund request failed, merchants need to handle the refund by themselves
NOTENOUGH	Not enough unsettled fund for refund	There is not enough unsettled fund for refund	This error code means refund request failed, merchants need to recall the refund API once there is enough unsettled fund, or retry it continuously
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

Note: All subscripts such as $n are started from 0.

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	Yes	String(32)	wx888888888888888	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)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	NO	String(32)	HMAC-SHA256	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This field is only required when sign_type is HMAC-SHA256
WeChat Order Number	transaction_id	Choose one to submit	String(32)	1217752501201407033233368018	Specifies the WeChat payment order id number
Vendor Order Number	out_trade_no		String(32)	1217752501201407033233368018	Specifies an internal order number created by the Vendor's system
Vendor Refund Number	out_refund_no		String(32)	1217752501201407033233368018	Vendor Refund Number
WeChat Refund Number	refund_id		String(32)	1217752501201407033233368018	WeChat Refund Number. Returned from Submit Refund API. 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
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	Yes	String(32)	wx888888888888888	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface

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(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	The WeChat payment order id number
Vendor Order Number	out_trade_no	Yes	String(32)	1217752501201407033233368018	An internal order number created by the Vendor's system
Total Amount	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)	GBP	ISO-4217 standard compliant and be described by three characters based code. 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
Payment Currency Type	cash_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.
Refund Amount	refund_fee_$n	Yes	Int	100	Specifies the bid amount of refund 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. The bid currency type

					should be consistent with the bid amount.
Refund Count	refund_count	Yes	Int	1	Record counts of refund
Vendor Refund Number	out_refund_no_$n	Yes	String(32)	12177525012014070332333680 18	Vendor refund number
WeChat Refund Number	refund_id_$n	Yes	String(32)	12177525012014070332333680 18	Wechat refund number
Refund Channel	refund_channel_$n	No	String(16)	ORIGINAL	ORIGINAL: Refund to original payment account BALANCE: Refund to Balance OTHER_BALANCE: Refund to balance of other WeChat account since the original account is abnormal OTHER_BANKCARD: Refund to other bank card since the original bank card is abnormal
Refund Status	refund_status_$n	Yes	String(16)	SUCCESS	Refund Status: SUCCESS: Refunded successfully REFUNDCLOSE: Refund failed PROCESSING: Refund is pending NOTSURE: Require the Vendor to call the Submit Refund API again with the original refund number 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.
Account Received	refund_recv_accout_$n	Yes	String(64)	招商银行信用卡 0403	The account which finally received the refund:

Refund					1.Back to bankcard: {bank name}{bank type}{tail numbers} 2.Back to balance: 支付用户零钱 3.Back to merchant 商户基本账户 Merchants' basic account 商户结算银行账户 Merchants' bank account
Refund Success Time	refund_success_time_$n	No	String(20)	2016-07-25 15:26:26	Refund success time, only returned when the refund the successfully complete.
Exchange Rate	rate	Yes	String(16)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000

Example:

```
<xml>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <nonce_str><![CDATA[TeqClE3i0mvn3DrK]]></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
REFUNDNOTEXIST	Failed to query refund	The refund number is wrong or the order status is not in the refund	Please check whether the refund number or transaction ID is correct or whether

		process	the order is paid or in refund process.
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 bid currency unit. (Like Chinese yuan, US dollar).

4. This API can only download the transactions within 3 months.

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 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 specifies the transactions related to this device.
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)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	NO	String(32)	HMAC-SHA256	This field is only required when sign_type is HMAC-SHA256. Default is MD5.
Reconciliation Start Date	bill_date	Yes	String(8)	20140603	Specifies the date of the transactions to be downloaded in the format of yyyymmdd, 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
Compress Type	tar_type	No	String(8)	GZIP	Fixed value: GZIP. It will return a .gzip package if this parameter is set, unless it will return the data flow.

```
<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.

Transaction records begin from the second row, and transaction fields are comma-separated values and begin with the “`” character (to the right of the “1” key on a standard keyboard) and in the same order as the fields listed in the header row of the returned data.

The second to last row will contain descriptive headers of the transaction history statistics and the last row will contain the statistical values of the returned transactions:

Total transactions, Total order amount, Total refunded amount, Total mobile coupon amount, Total mobile coupon refunds

An example of returned data follows:

Transaction time, Official account ID(appid), Vendor ID(mch_id), Sub vendor ID(sub_mch_id), Device ID(Device_info), Wechat order number(transaction_id), Vendor order number(out_transaction_id), User tag(openid), Transaction type(trade_type), Transaction status(trade_state), Payment bank(bank_type), Currency type(fee_type), Total amount(total_fee), Coupon amount(coupon_fee), Wechat refund number(refund_id), Vendor refund number(out_refund_no), Refund amount(refund_fee), Coupon refund amount(coupon_refund_fee_$n), Refund type, Refund status(refund_status_$n), Product name, Vendor's data package(attach), Fee, Rate, Payment Currency type(Cash_fee_type), Cash payment amount(Cash_fee), Settlement currency type, Settlement currency amount, Exchange rate, Refund exchange rate, Payer's Refund amount, Payer's Refund currency type, Refund currency type, Refund settlement currency type, Refund settlement amount

2016-01-2812:09:57,`wx77f7fce3722c712b,`1277115701,`0,`1002200845,`10002210082016012829496583
32,`212412016012552075176 5,`orB2gVv8fX95-ghb8ws3Eunw,`MICROPAY,`SUCCESS,`GDB_CREDIT,`H
KD,`198.00,`0.00,`0,`0,`0,`0,`,`,`商品,`,`1.19000,`0.60%,`CNY,`168.01,`HKD,`198.00,`84853800,`0,`0,`,`,`,`0

At the end of each transaction history form, there is the statistic information, please refer to the following fields for detailed information:

Total transaction count, Total transaction amount, Total refund amount, Total coupon refund amount, Total commission amount

An example is as following:

Total transaction count	*Total transaction amount*	*Total refund amount*	*Total coupon refund amount*	*Total commission amount*
`110	`6096.73	`0.00	`0.00	`152.45000

6 Error Code

Name	Description	Reason	Solution
SYSTEMERROR	API return error	System timed out	Try to query again
invalid bill_type	Parameter error	Requested parameters are not correct	Parameter error. Check parameters and try again.
data format error			
missing parameter			
SIGN ERROR	Signature error	Signature error	Please check the signature
Bill Creating	The report is not created	There is no successful transactions for the	Please check whether there are successful

	yet	specified date or the report is generating	transactions in the specified data. If there is, please, please call the API after 10 a.m. GTM+8.
CompressGZip Error	Compressing failed	Compressing failed, please retry later	Compressing failed, please retry later
UnCompressGZip Error	Uncompressing failed	Uncompressing failed, please retry later	Uncompressing failed, please retry later

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 30 minutes, so as to ensure successful receipt. However, the WeChat payment system cannot ensure successful receipt of payment notifications in every case. (Notification frequency: 15/15/30/180/1800/1800/1800/1800/3600s)

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.

Note: To avoid receiving fake notifications, merchants must verify the signature of the notification and check whether the order amount in the notification is the same with the one in merchant's system.

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 the URL is inaccessible, the vendors 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	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 of result_code.
Return Data	return_msg	No	String(	Signature	If not empty, this is the error description

			128)	failure	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	The Official Account ID assigned by WeChat
Vendor ID	mch_id	Yes	String(32)	1900000109	The vendor ID assigned by WeChat Payment
Device ID	device_info	No	String(32)	013467007045764	The terminal device 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.
Sign Type	sign_type	No	String(32)	HMAC-SHA256	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This field is only required when sign_type is HMAC-SHA256.
Service Result	result_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL
Error Code	error_code	No	String(32)	SYSTEMERROR	The error code returned
Error Code Description	error_code_des	No	String(128)	System error	The detailed error description returned.
User Tag	openid	Yes	String(128)	wxd930ea5d5a258f4f	The user id of the Payer provided by the WeChat system in OpenID format as unique tag on vendor's appid. Also it is unique to each appid instance.
Follows Official Account or not	is_subscribe	No	String(1)	Y	Specifies whether the payer follows the associated official account or not, with Y meaning 'follows' and N meaning 'not follows'.
Transaction	trade_type	Yes	String(16	JSAPI	Set as JSAPI, NATIVE, or APP

Type			)		
Payment Bank	bank_type	Yes	String(16)	CMC	Use strings for bank type. For more information, see attachments.
Total Amount	total_fee	Yes	Int	100	Specifies the total amount for a transaction in cents as integer. For more information, see Section 4.2. Payment Amount.
Currency Type	fee_type	Yes	String(8)	GBP	ISO-4217 standard compliant and be described by three characters based code. For more information, see Section 4.2 Currency Type
Cash Payment Amount	cash_fee	Yes	Int	100	Specifies the total payment amount of a transaction. For more information, see Section 4.2 Payment Amount.
Cash Type	cash_fee_type	Yes	String(16)	CNY	Complies with ISO 4217 standards and uses CNY (Chinese yuan) by default. For more information, see Section 4.2 Currency Type
Specifies the number of a WeChat payment order	transaction_id	Yes	String(32)	1217752501201407033233368018	The WeChat payment order id number
Vendor Order Number	out_trade_no	Yes	String(32)	1212321211201407033568112322	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)	20141030133525	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.
Exchange Rate	rate	Yes	String(16)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000

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. Report Speed Testing

1 Use Case

This API is used to assist the Vendors in improving overall service quality when they call the WeChat payment APIs. With this API, vendors obtain relevant returned data and response duration. Based on the

Vendor's uplink network speed, the WeChat payment system optimizes its networking deployment, so as to continuously improve service reliability and speed.

2 URL

https://api.mch.weixin.qq.com/payitil/report

3 Certificate Requirement

No

4 Input Parameter

Field Name	ID	Required	Type	Example	Description
Official Account ID	appid	No	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 is defined by vendors.
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)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign Tyep	Sign_type	No	String(32)	HMAC-SHA256	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This field is only required when sign_type is HMAC-SHA256.
Interface URL	interface_url	Yes	String(127)	https://api.mch.weixin.qq.com/pay/unifiedorder	Complete interface URL to be reported is similar to the following: https://api.mch.weixin.qq.com/pay/unifiedorder When Quick Pay is used, in order to analyze the overall duration of a single service with vendors, WeChat requires a report for each Quick Pay service call made in either mode 1 or mode 2. The report URL is as below:

					https://api.mch.weixin.qq.com/pay/micropay/total For more information about mode 2, see Section 5.4 Process for vendors. For other payment methods, calling and reporting is only required once.
API Execute Duration	execute_time_	Yes	Int	1000	Specifies the duration for calling the API. The units are expressed in milliseconds.
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 of trade_state.
Return Data	return_msg	No	String(128)	Signature failure	If not empty, this is the error description Signature failure Parameter format checking error
Service Result	result_code	Yes	String(16)	SUCCESS	SUCCESS or FAIL
Error Code	err_code	No	String(32)	SYSTEMERROR	ORDERNOTEXIST: Order does not exist SYSTEMERROR: System error
Error Code Description	err_code_des	No	String(128)	System error	Describes result data
Vendor Order Number	out_trade_no	No	String(32)	1217752501201407033233368018	Specifies an order number created by the Vendor's system. The Vendor can report this order number to WeChat payment system in order to help improve payment services.
Access Interface IP	user_ip	Yes	String(16)	8.8.8.8	Specifies the machine IP used when calling this interface
Vendor Report Time	time	No	String(14)	20091227091010	Specifies system time in the format of yyyyMMddHHmmss, such as 20091227091010 for Dec 27, 2009 09:10:10. For more information, see Section4.2 Time Protocol.

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 of result_code.
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

6 Error Code

None

9. Short URL Conversion

1 Use Case

This API is used for the URL embedded in a QR code when using native payment method 1 and converts it into short URL (weixin://wxpay/s/XXXXXX), so as to reduce data volume, speed up scanning, and improve accuracy.

2 URL

https://api.mch.weixin.qq.com/tools/shorturl

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

Long URL	long_url	Yes	String(512)	weixin://wxpay/bizpayurl?sign=XXXXX&appid=XXXXX&mch_id=XXXXX&product_id=XXXXXX&time_stamp=XXXXXX&nonce_str=XXXXX	Specifies the URL to be converted. The signature keeps original string and uses URL encode for transferring data.
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)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.
Sign type	sign_type	NO	String(32)	HMAC-SHA256	Currently HMAC-SHA256 and MD5 are supported, default is MD5. This field is only required when sign_type is HMAC-SHA256

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 of result_code.
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	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface
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)	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	SYSTEMERROR: System error URLFORMATERROR: Incorrect URL format
Short URL	short_url	Yes	String(64)	weixin://wxpay/s/XXXXXX	Specifies the converted URL

6 Error Code

Name	Description	Reason	Solution
SIGNERROR	Signature error	Incorrect signature result	Check whether the signature parameter and method comply with signature algorithm requirements
REQUIRE_POST_METHOD	Use post method	Data is not transferred by post	Check whether data is submitted by POST method
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

LACK_PARAMS	Missing parameter	Required parameter is missing	Check whether parameter is provided
XML_FORMAT_ERROR	INVALID XML FORMAT	INVALID XML FORMAT	Check whether XML parameters are in correct format
POST_DATA_EMPTY	post data is empty	post data can't be empty	Check whether POST data is empty

10. QUERYING SETTLED FUNDS

1 Use Case

This API helps query the details of settled funds.

2 URL

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

3 Certificate Requirement

No certificate is required.

4 Request Parameters

Field Name	ID	Type	Required	Example	Description
Official Account ID	appid	String(32)	Yes	Wx1378acui7865dt65	Specifies Official Account ID assigned by WeChat
Vendor ID	mch_id	String(32)	Yes	12567876549	Specifies vendor ID assigned by WeChat Pay
Settlement Status	usetag	Int	Yes	1	Indicates that the fund has been settled or is still outstanding: 1 - settled 2 - outstanding
Random String	nonce_str	String(32)	Yes	Uihu276jjghxlehu3876440932	A random string. 32 characters or fewer.

				2	
Offset	offset	Int	Yes	0	Returned query data starts from this offset value. Example: if there is 10 records and offset is set as 2, it will return the value from the third row
Max Records	limit	Int	Yes	10	The max amount 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). (This field is optional when inquiring the outstanding records)
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). (This field is optional when inquiring the outstanding records)
Signature	sign	String(32)	Yes	8439EBD8AE7422590E2004D0C5EBFFBC	For details about signature, see Section 3.2

5. Return Data

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)	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	The Vendor ID submitted when calling the interface
Random String	nonce_str	Yes	String(32)	Indicates a random string of less than 32 characters
Service Result	result_code	Yes	String(16)	SUCCESS/FAIL
Error Code	err_code	No	String(32)	Please refer to Error code
Error Code Description	err_code_des	No	String(128)	Describes detailed error information
Return Data Lines	record_num	Yes	String(10)	Indicates records amount of return data
The following fields will be returned when both return_code and result_code are **SUCCESS**. In case of multiple records, the data will be repeated.				
Exchange Payment Batch Number	fbatchno	Yes	String(32)	Indicates the batch number of exchange 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)	Comply with ISO 4217 standards and use CNY for Chinese currency by default. Description: CNY：Chinese yuan GBP: Great Britain pound HKD: Hong Kong dollar USD: United States dollar JPY: Japanese yen CAD: Canadian dollar AUD: Australian dollar EUR: Euro THB: Thailand baht Notes: The currency type for payment and refund shall be the same.
Payment Amount	pay_fee	Yes	Int	Priced in foreign currency at the minimum trading unit
Refund Amount	refund_fee	Yes	Int	Priced in foreign currency at the minimum trading unit
Net Payment Amount	pay_net_fee	Yes	Int	Priced in foreign currency at the minimum trading unit
Charge Amount	poundage_fee	Yes	Int	Priced in foreign currency at the minimum trading unit

Example:

```
<xml>
  <return_code>SUCCESS</return_code>
  <result_code>SUCCESS</result_code>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <record_num>10</record_num>
  <nonce_str>ec231627564 1faa3aacf3cc599e8730f</nonce_str>
  <fbatchno>10</fbatchno>
  <date_settlement>20150807</date_settlement>
```

```
<date_start>20150807</date_start>
<date_end>20150807</date_end>
<sign>FDD167FAA73459FD921B144BAF4F4CA2</sign>
<settlement_fee>1000</settlement_fee>
<unsettlement_fee>0</unsettlement_fee>
<settlement_type>USD</settlement_type>
<pay_fee>1000</pay_fee>
<refund_fee>0</refund_fee>
<pay_net_fee>1000</pay_net_fee>
<poundage_fee>0</poundage_fee>
</xml>
```

11. QUERY EXCHANGE RATE API

1. Use Case

The interface could inquiry the exchange rate which Wechat Payment used in real time. The rate is updated once each day at 10 a.m. in the time zone (GMT+8).

2. URL

https://api.mch.weixin.qq.com/pay/queryexchagerate

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
Sub Vendor ID	sub_mch_id	No	String(32)	1230000100	Specifies Sub Vendor ID assigned by WeChat
Currency Type	fee_type	Yes	String(10)	USD	Foreign currency type. ISO-4217 standard compliant and be described by three characters based code.
Date	date	Yes	String(14)	20150807	In the format of yyyyMMdd. For example, December 25, 2009 is

					expressed as 20091225 in the time zone (GMT+8).
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.

Example:

```
<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <fee_type>CNY</fee_type>
  <date>20150807</date>
  <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 of result_code.
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	The Official Account ID submitted when calling the interface
Vendor ID	mch_id	Yes	String(32)	1900000109	The Vendor ID submitted when calling the interface
Sub Vendor ID	sub_mch_id	No	String(32)	1230000100	The Sub Vendor ID submitted when calling the interface
Currency Type	fee_type	Yes	String(10)	USD	Foreign currency type. ISO-4217 standard compliant and be described by three characters based code.
Exchange Rate Time	rate_time	Yes	String(14)	20150807	In the format of yyyyMMdd. For example, December 25, 2009 is

					expressed as 20091225 in the time zone (GMT+8).
Exchange Rate	rate	Yes	String(32)	650000000	The value is 10 to the 8th power times of the exchange rate from foreign currency to RMB. For example, the exchange rate from foreign currency to RMB is 6.5, the value will be 650000000
Signature	sign	Yes	String(32)	C380BEC2BFD727A4B6845133519F3AD6	Specifies a signature. For more information, see Section 4.3.1 Signature Algorithm.

Example:

```
<xml>
  <appid>wx2421b1c4370ec43b</appid>
  <mch_id>10000100</mch_id>
  <fee_type>USD</fee_type>
  <rate_time>20150807</rate_time>
  <rate>650000000</rate>
  <sign>FDD167FAA73459FD921B144BAF4F4CA2</sign>
</xml>
```

12. DOWNLOAD PAYMENT SDKS

Download the Quick Pay SDK
Download the Official Account Payment SDK
Download the App-based Payment SDK

13. FAQ AND ADDITIONAL REMARKS

6.SSL Minimum Version Requirement for Servers

As lower-versions of SSL encryption are more susceptible to vulnerabilities, developers should confirm whether their SSL version requires an update. For more information, see the update notice in the WeChat Official Account Admin Platform：
https://mp.weixin.qq.com/cgi-bin/announce?action=getannouncement&key=1414562353&version=11&lang=zh_CN

Java-based developers can also refer to the instructions on the Oracle website: [Oracle Java disables SSL3](http://www.oracle.com/technetwork/java/javase/documentation/cve-2014-3566-2342133.html)

7.IPv6 Requirements

If you have enabled IPv6, the IP of your site on as stored on our servers will be an IPv6 address. As IPv6 is not supported throughout the Internet globally, you may face potentially long connection timeouts on IPv6.

We suggest you use IPv4 to resolve addresses when calling payment APIs.

How to use curl for PHP programing:

```
if(defined('CURLOPT_IPRESOLVE') && defined('CURL_IPRESOLVE_V4')){
    curl_setopt($ch, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4);
```

14. CONTACT US

1. Follow the "WeChat Payment for Vendors" official account and send us your feedback;
2. Contact us via email at weixinpay@tencent.com；