**Scan a QR code Payment Api**

[1. Overview 3](#_Toc32495)

**[1.1. Purpose](#_Toc6040)** [3](#_Toc6040)

[1.2. Terms and acronyms 3](#_Toc14962)

[1.3. List of interfaces 3](#_Toc30745)

[2. Interface definition 3](#_Toc18034)

**[2.1. WeiXin\alipay\QQ payment transaction interface](#_Toc6057)** [3](#_Toc6057)

[2.1.1. Description 3](#_Toc14506)

[2.1.2. Definition of request parameters 4](#_Toc31591)

[2.1.3. Definition of callback parameters 5](#_Toc22239)

**[2.3. Payment result notification API](#_Toc23197)** [7](#_Toc23197)

[2.3.1. Description 7](#_Toc18788)

[2.3.2. Definition of notification parameters 7](#_Toc6584)

**[2.4. Single order query API](#_Toc29364)** [8](#_Toc29364)

[2.4.1. Description 8](#_Toc27137)

[2.4.2. Definition of query parameters 8](#_Toc7367)

[2.4.3. Definition of response parameters 9](#_Toc7200)

**[2.5. Appendix](#_Toc29190)** [11](#_Toc29190)

[2.5.1. Definition of error code 11](#_Toc25518)

# Overview

## **1.1. Purpose**

This file defines the following four interfaces: WeChat\Alipay\ payment API between merchant website and QuickPay payment platform, server asynchronous notification API and transaction query API between merchant website and QuickPay payment platform.

## 1.2. Terms and acronyms

**Merchant Code:** The unique ID for the QuickPay platform registered by a merchant.

**Backend Notification:** Server asynchronous notification. After QuickPay processes the data obtained, the QuickPay server will positively send a notice to the address(notify\_url) designated by the merchant website, carrying the order information processed. The correct practice is managing and operating website orders here. Example address:http://www.xhbill.com/notify\_url.jsp.

**Scope of application :** **WeChat payment、Alipay payment 、QQ** **payment、Union payment、JingDong payment.**

## 1.3. List of interfaces

|  |  |
| --- | --- |
| **Serial number** | **Name of interface** |
| 1 | Wechat\Alipay\QQ payment API between merchant website and QuickPay payment platform |
| 2 | QuickPay platform qr code information sent to the merchant website interface |
| 3 | Payment result notification interface between merchant website and QuickPay payment platform |
| 4 | Transaction query interface between merchant website and QuickPay payment platform |

# Interface definition

## **2.1. Scan payment transaction interface**

### 2.1.1. Description

Define the Wechat\Alipay\QQ payment transaction interface between merchant website and QuickPay payment platform,the mercant website should submit transaction order data to QuickPay payment gateway based on the definition of the interface specification ,and the data should be submitted in form of HTTPS POST.

### 2.1.2. Definition of request parameters

**Transaction request address**: **[https://api.xhbill.com/](https://pay.dinpay.com/gateway?input_charset)gateway/api/scanpay**

The request parameter should be encoded by UTF-8.(Currently only supports this encoding format)

The payment request parameters are defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type(length)** | **Required** | **Notes** |
| **Basic parameter** | | | |
| merchant\_code | String(10) | √ | The sole merchant code distributed by QuickPay payment platform during signing contract with merchant. |
| service\_type | String(10) | √ | Fix value:  alipay\_scan（alipay scan）  weixin\_scan（wechat scan）  qq\_scan（qq scan）  union\_scan（union scan）  jd\_scan（jingdong scan） |
| notify\_url | String(200) | √ | The address receives the notification from QuickPay server.  for example:  http://www.xhbill.com/notify\_url |
| interface\_version | String(10) | √ | Fix value:V3.1(Capital) |
| client\_ip | String(15) | √ | Consumers to use the client IP, such as: 192.168.1.25 |
| sign\_type | String(10) | √ | Parameter Name: Signature type  support value: RSA or RSA-S, not participate in signature,please choose one from them  RSA: encrypt data by certification  RSA-S:encrypt data by String Keys |
| sign | String | √ | Parameter Name: Signature  QuickPay sample code will tell you how to get the value of it |
| **Service parameter** | | | |
| order\_no | String(64) | √ | Oreder number,generated from merchant website system,should be unique and consist of letters and numbeers. for example:  1000201666 |
| order\_time | Date | √ | Oreder time of merchant,  format:yyyy-MM-dd HH:mm:ss  for example:  2013-11-01 12:34:58 |
| order\_amount | Number(13,2) | √ | It should be in unit of RMB,and two digits after the decimal point.  for example:12.01 |
| product\_name | String(100) | √ | No more than 100 characters.for expmple:  ASUSG750Y47JX-BL |
| product\_code | String(60) | × | No more than 60 characters,  for example:  ASUSB85M-ER2.0 |
| product\_num | Number(10) | × | For example:5 |
| product\_desc | String(300) | × | Product description,no more than 300 characters |
| extra\_return\_param | String(100) | × | If such parameter is delivered in payment request,then this parameter will be returned when notifying the merchant that the payment is made successfully . |
| extend\_param | String | × | Format:parameter name 1^parameter value 1|parameter name 2^parameter value 2,multiple data should be separated with “|”,for example:  name Zhang San|sex^Male |

The public use service extend parameters are defined as follows: (**extend\_param**)

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type(length)** | **Required** |
| **Information of receiver** | | |
| ship\_to\_name | String(50) | × |
| ship\_to\_email | String(60) | × |
| ship\_to\_phone | String(20) | × |
| ship\_to\_state | String(100) | × |
| ship\_to\_city | String(100) | × |
| ship\_to\_street | String(200) | × |
| ship\_to\_zip | String(20) | × |
| **Customer information** | | |
| customer\_email | String(100) | × |
| customer\_name | String(50) | × |
| customer\_phone | String(20) | × |
| customer\_state | String(100) | × |
| customer\_city | String(100) | × |
| customer\_street | String(200) | × |
| customer\_zip | String(20) | × |
| customer\_cardNumber | String(50) | × |
| customer\_idNumber | String(30) | × |

### 2.1.3. Definition of callback parameters

The callback parameters of request are defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type(length)** | **Required** | **Notes** |
| resp\_code | String(5) | √ | processing code ,came back from QuickPay payment platform.  SUCCESS,It means processing is successful .  Other value,please refer to the error code definition. |
| resp\_desc | String | √ | The description about the relevant resp\_code. |
| qrcode | String(179) | × | Qr code information content,it must be required when the value of resp\_code is SUCCESS. |
| sign\_type | String | √ | RSA or RSA-S(Capital), not participate in signature |
| sign | String | √ | Signature data |

When the tansaction is successful,the returnde XML data is as follows:

|  |
| --- |
| <? xml version=”1.0” encoding=”UTF-8” ?>  <dinpay>  <resonse>  <resp\_code>SUCCESS</resp\_code>  <resp\_desc>成功</resp\_desc>  <sign\_type>RSA-S</sign\_type>  <sign>56ae9c3286886f76e57e0993625c71fe</sign>  <trade>  <qrcode>weixin://wxpay/bizpayurl?pr=BANyrOG</qrcode>  </trade>  </response>  </dinpay> |

When the tansaction is unsuccessful,the returnde XML data is as follows:

|  |
| --- |
| <? xml version=”1.0” encoding=”UTF-8” ?>  <dinpay>  <resonse>  <resp\_code>OREDER\_NO\_IS\_TOO\_LONG</resp\_code>  <resp\_desc>商家订单号太长</resp\_desc>  <sign\_type>RSA-S</sign\_type>  <sign>56ae9c3286886f76e57e0993625c71fe</sign>  <trade></trade>  </response>  </dinpay> |

## **2.3. Payment result notification API**

### 2.3.1. Description

When the consumer finished the transaction, notfiy\_url(you can find it on request parameters) will receive notice from QuickPay server. QuickPay server won’t send notice if transaction failed.

### 2.3.2. Definition of notification parameters

The notification API parameters are defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type**  **(length)** | **Required** | **Notes** |
| **Basic parameter** | | | |
| merchant\_code | String(10) | √ | The sole merchant code distributed by QuickPay payment platform during signing contract with merchant. |
| notify\_type | String(14) | √ | Fix value:offline\_notify |
| notify\_id | String(100) | √ | It is that after receiving such notification information, the merchant system use such  verification ID to verify the legitimacy of such notification at QuickPay payment platform, it shall consist of 32 digits of non-repeated numbers and letters, for example: 235dst58fd1dwe21354fdafdaesfdsaf.  Note:This function stops check at the current version, parameters are still reserved. |
| interface\_version | String(10) | √ | Fix value:V3.0(Capital) |
| sign\_type | String(10) | √ | RSA or RSA-S(Capital), not participate in signature |
| sign | String | √ | Signature data,for more details,please refer to the following signature rule in the appendix |
| **Service parameter** | | | |
| order\_no | String(64) | √ | Oreder number,generated from merchant website system,should be unique and consist of letters and numbers. for example:1000201666 |
| order\_time | Date | √ | Oreder time of merchant,  format:yyyy-MM-dd HH:mm:ss  for example:  2013-11-01 12:34:58 |
| order\_amount | Number(13,2) | √ | It should be in unit of RMB,and two digits after the decimal point.  for example:12.01 |
| extra\_return\_param | String(100) | × | If such parameter is delivered in payment request,then this parameter will be returned when notifying the merchant that the payment is made successfully . |
| trade\_no | String(30) | √ | QuickPay transaction order code,  for example: 1000004817 |
| trade\_time | Date | √ | QuickPay transaction order time  format: yyyy-MM-dd HH:mm:ss, for example: 2013-12-1 12:23:34. |
| trade\_status | String(7) | √ | Transaction status of the order .  SUCCESS, the transaction is successful |
| bank\_seq\_no | String | × | Sequence number of bank transaction, for example: 2013060911235456. |

## **2.4. Single order query API**

### 2.4.1. Description

Merchants can use this API to query the order detail ,including failed and successful orders.At present,just supporting the orders within 12 hours from order\_time.

### 2.4.2. Definition of query parameters

Request address: [https://query.xhbill.com/query](https://query.dinpay.com/query)

The parameters are defind as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type**  **(length)** | **Required** | **Notes** |
| **Basic parameter** | | | |
| service\_type | String(18) | √ | Fix value:  single\_trade\_query |
| merchant\_code | String(10) | √ | The sole merchant code distributed by QuickPay payment platform during signing contract with merchant |
| interface\_version | String(10) | √ | Fix value:V3.0(Capital) |
| sign\_type | String(10) | √ | Parameter Name: Signature type  support value: RSA or RSA-S, not participate in signature,please choose one from them  RSA: encrypt data by certification  RSA-S:encrypt data by String Keys |
| sign | String | √ | Parameter Name: Signature  QuickPay sample code will tell you how to get the value of it | |
| **Service parameter** | | | | |
| order\_no | String(64) | √ | Order number,generated from merchant website system,should be unique and consist of letters and numbeers. for example:  1000201666 |
| trade\_no | String(30) | × | QuickPay transaction order code, for example: 1000004817 |

### 2.4.3. Definition of response parameters

Return response data in synchronization in the format of XML

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type**  **(length)** | **Required** | **Notes** |
| **Basic parameter** | | | | |
| is\_success | String(1) | √ | T:query is successful  F:failing to query |
| sign\_type | String(10) | √ | RSA or RSA-S(Capital), not participate in signature |
| sign | String | √ | Signature data |
| error\_code | string | × | It will return an error code when the query fails.for example:TRADE\_IS\_NOT\_EXIST |
| **Service parameter(the value of is\_succcess is T)** | | | | |
| merchant\_code | String(10) | √ | The sole merchant code distributed by QuickPay payment platform during signing contract with merchant |
| order\_no | String(64) | √ | Oreder number,generated from merchant website system,should be unique and consist of letters and numbeers.  for example:1000201666 |
| order\_time | Date | √ | Oreder time of merchant,  format:yyyy-MM-dd HH:mm:ss  for example:  2013-11-01 12:34:58 |
| order\_amount | Number(13,2) | √ | It should be in unit of RMB,and two digits after the decimal point.  for example:12.01 |
| trade\_no | String(30) | √ | QuickPay transaction order code,  for example: 1000004817 |
| trade\_time | Date | √ | QuickPay transaction order time  format: yyyy-MM-dd HH:mm:ss,  for example: 2013-12-01 12:23:34. |
| trade\_status | String(7) | √ | Transaction status of the order:  SUCCESS, the transaction is successful  UNPAY, the payment is not made. |

When query is successful, the returned XML data is as follows:

|  |
| --- |
| <version="1.0" encoding="UTF-8" ?>  <dinpay>  <response>  <is\_success>T</is\_success>  <sign\_type>RSA-S</sign\_type>  <sign>56ae9c3286886f76e57e0993625c71fe</sign>  <trade>  <merchant\_code>2181230245</merchant\_id>  <order\_no>210023569</order\_no>  <order\_time>2013-05-10 11:18:00</order\_time>  <order\_amount>100.00</order\_amount>  <trade\_no>128600</trade\_no>  <trade\_time>2013-05-10 11:20:01</trade\_time>  <trade\_status>SUCCESS</trade\_status>  </trade>  </response>  </dinpay> |

When query is successful, the returned XML data is as follows:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <dinpay>  <response>  <is\_success>F</is\_success>  <error\_code>TRADE\_IS\_NOT\_EXIST</error\_code>  </response>  </dinpay> |

## **2.5. Appendix**

### 2.5.1. Definition of error code

|  |  |
| --- | --- |
| **Error code** | **Notes** |
| ILLEGAL\_URL\_FORMAT | Meaning:format of notification url is incorrect  Reason:the format of the value of payment parameter notify\_url is incorrect  Correct example:http://www.xhbill.com/notify\_url |
| UNKNOW\_ERROR | Meaning: unknown error  Reason: please refer to resp\_desc and result\_desc |
| OTHER\_ERROR | Meaning: other errors  Reason: reference result\_desc description |
| csp.orderService.order-is-already-exist | Meaning:order already exists  Reason:order\_no is submitted repeatly |
| csp.orderService.prepay-create-order-failure | Meaning:generate the order failed  Reason:System fails to generate order |
| ILLEGAL\_REQUEST | Reason:parameter error |
| ILLEGAL\_PAY\_BUSINESS | Meaning:unknow payment business  Reason:the payment business is not open,please contact our  salesman |
| SYSTEM\_ERROR | Reason:something is wrong with the system,like network timeout and so on |
| NOTIFY\_URL\_IS\_NULL | Reason:it can not be null with the notify\_url |
| NOTIFY\_URL\_IS\_TOO\_LONG | Reason:the length of notify\_url is too long |
| ORDER\_NO\_IS\_NULL | Reason:it can not be null with the order\_no |
| ORDER\_NO\_IS\_TOO\_LONG | Reason:the length of order\_no is too long |
| ORDER\_TIME\_IS\_NULL | Reason:it can not be null with the order\_time |
| ORDERNO\_IS\_ALREADY\_EXISTS | The order number has been in existence |
| ILLEGAL\_ORDER\_TIME\_FORMAT | Reason:the format of order\_time is incorrect  Correct example:2013-12-01 12:23:34 |
| ORDER\_AMOUNT\_IS\_NULL | Reason:it can not be null with the order\_amount |
| ILLEGAL\_ORDER\_AMOUNT\_FORMAT | Reason:the format of order\_amount is incorrect  Correct example:12.01 |
| ORDER\_AMOUNT\_IS\_TOO\_LARGE | Reason:the length of order\_amount can not exceed 13 figures |
| PRODUCT\_NAME\_IS\_NULL | Reason:it can not be null with the product\_name |
| PRODUCT\_NAME\_IS\_TOO\_LONG | Reason:the length of procuct\_name can not exceed 100 characters |
| ILLEGAL\_PRODUCT\_NUM\_FORMAT | Reason:the format of product\_num is incorrect  correct example:10 |
| PRODUCT\_CODE\_IS\_TOO\_LONG | Reason:the length of procuct\_code can not exceed 60 characters |
| PRODUCT\_DESC\_IS\_TOO\_LONG | Reason:the length of procuct\_desc can not exceed 300 characters |
| EXTRA\_RETURN\_PARAM\_IS\_TOO\_LONG | Reason:the length of extra\_return\_param can not exceed 100 characters |
| ILLEGAL\_EXTEND\_PARAM\_FORMAT | Reason:the format of extend\_param is incorrect  correct example:name Zhang San|sex^Male |
| ILLEGAL\_SERVICE\_TYPE | Reason: incorrect service\_type or service\_type is null |
| MERCHANT\_CODE\_IS\_NULL | Reason:it can not be null with the merchant\_code |
| INTERFACE\_VERSION\_IS\_NULL | Reason:it can not be null with the interface\_version |
| ILLEGAL\_INTERFACE\_VERSION | Reason:incorrect interface\_version,it must be V3.0 |
| SIGN\_TYPE\_IS\_NULL | Reason:it can not be null with the sign\_type |
| ILLEGAL\_SIGN\_TYPE | Reason:unsupported sign\_type,it must be RSA-S or RSA |
| SIGN\_IS\_NULL | Reason:it can not be null with the sign |
| ILLEGAL\_SIGN | Meaning:the signature is incorrect,signature data is inconsistent  Possible reason:  1)Some requierd parameters don’t paticipate in singnature.  2)The sequence of signature section is incorrect,it must be in the sequence of a~z  3)Wrong private key or the private key is inconsistent of the public key saved in QuickPay merchant system |
| ILLEGAL\_PARAMS | Reason:your request parameter contains illegal characters |
| VIOLATE\_DAY\_MAXAMOUNT | In violation of the risk control rules: merchants maximum daily transaction amount |
| VIOLATE\_DAY\_MAXCOUNT | In violation of the risk control rules: stores the daily maximum number of transactions |
| VIOLATE\_MONTH\_MAXCOUNT | In violation of the risk control rules: merchants largest transaction amount per month |
| VIOLATE\_SING\_MAXAMOUNT | In violation of the risk control rules: exceed the maximum amount limitation of the order |
| VIOLATE\_WITHCARD\_SINGLE\_OVER | Violation of the rules: the same card number single order amount limitation |
| GET\_QRCODE\_FAILED | Failed to get qr code |