**Bank Payment API of QuickPay**

|  |  |
| --- | --- |
| Writer | Updated |
| Technical Support Dept | 2017-2-17 |

[1. Introduction 3](#_Toc23719)

[1.1. Purpose 3](#_Toc26924)

[1.2. Terms and Abbreviations 3](#_Toc26064)

[1.3. Integration Process 4](#_Toc13496)

[1.4. List of interfaces 4](#_Toc14133)

[2. Interface definition 4](#_Toc26895)

[2.1. Bank payment transaction interface 4](#_Toc21183)

[2.1.1. Description 4](#_Toc22120)

[2.1.2. Definition of interface parameters 4](#_Toc3569)

[2.2. Payment result notification interface 9](#_Toc20923)

[2.2.1. Description 9](#_Toc29022)

[2.2.2. Definition of interface parameters 9](#_Toc4832)

[2.3. Single transaction query interface 11](#_Toc10440)

[2.3.1. Description 11](#_Toc17591)

[2.3.2. Definition of interface parameters 12](#_Toc12588)

[2.3.3. Definition of response parameters 13](#_Toc24437)

[2.4. Appendix 16](#_Toc7246)

[2.4.1. Supported bank code 16](#_Toc26261)

[2.4.2. Definition of error code 17](#_Toc15743)

# Introduction

## 1.1. Purpose

This file defines the following API: China Bank payment API, payment result notice API, and order query API .

## 1.2. Terms and Abbreviations

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

**2) bank\_code**: When a user in the use of Internet Bank Direct, when the user submit the bank code it will redirect to the corresponding bank page.

**3) return\_url**: Page synchronous notification. Several seconds after QuickPay processes the data obtained (or after the user clicks the button), the QuickPay payment page will jump to the return\_url address designated by the merchant, carrying the feedback after the processing is finished. Example: [https://www.xhbill.com/Return\_Url.php](https://www.dinpay.com/Return_Url.php)

**4) notify\_url**: Server asynchronous notification. After QuickPay processes the data obtained, the QuickPay server will positively send a notice to the notify\_url address designated on the merchant’s website, carrying the processed order information. Example: [http://www.xhbill.com/Notify\_Url.php](http://www.dinpay.com/Notify_Url.jsp)

## 1.3. Integration Process

Developed as API files

Register and blind domain

Activate B2C Payment Service

Test on live environment

## 1.4. List of interfaces

|  |  |
| --- | --- |
| **No.** | **Interface Name** |
| 1 | Bank payment interface between merchant’s website and QuickPay payment platform |
| 2 | Payment result notification interface between merchant’s website and QuickPay payment platform |
| 3 | Transaction query interface between merchant’s website and QuickPay payment platform |

# Interface definition

## 2.1. Bank payment transaction interface

### 2.1.1. Description

This section is description of bank payment transaction interface which between merchant's website and QuickPay's gateway. Transaction order data submitted (in form of HTTPS POST) to QuickPay shall be based on this definition of interface specification.

### 2.1.2. Definition of interface parameters

Transaction request address: https://pay.xhbill.com/gateway?input\_charset={coded character set of parameter}

For example: [https://pay.xhbill.com/gateway?input\_charset=UTF-8](https://pay.dinpay.com/gateway?input_charset=UTF-8)

The coded character is in consistency with the value of the input\_charset request parameter, UTF-8 or GBK.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Required** | **Notes** |
| merchant\_code | String(10) | √ | Parameter Name: Merchant ID  QuickPay assigned to the merchant's identity when they establish the cooperation relations. |
| service\_type | String(10) | √ | Parameter Name: Service type  It has a fixed value: direct\_pay |
| notify\_url | String(200) | √ | Parameter Name: The address of server asynchronous notification.  After payment is made successfully, QuickPay will actively notice to merchant via this URL. Please make sure that it's effective. |
| interface\_version | String(10) | √ | Parameter Name: Interface version  Fixed value: V3. 0 (Capital) |
| input\_charset | String(5) | √ | Parameter Name: Coded character set of parameter  Value: UTF-8 or GBK (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 |
| return\_url | String(200) | × | Parameter Name: The address of page synchronous notification  After payment is made successfully, return to merchant’s website in form of page return. |
| pay\_type | String(10) | × | Parameter Name: Pay type  Values are as follows（The value must be lowercase, when multiple selections separated by commas）  b2c（B2C）,express（express）,weixin（wechat scan）,alipay\_scan（alipay scan）,qq\_scan（qq scan）,union\_scan（union scan）,jd\_scan（jingdong scan） |
| client\_ip | String(15) | × | Parameter Name: Client IP  The IP of machine used by user during creating transaction, for example: 192.168.1.25, the maximal length includes 15 characters. |
| order\_no | String(100) | √ | Parameter Name: Merchant's order No.  1.This order number is generated by merchant’s website, which should be uniqueness.  2.It's composed of letters, numbers or underscores, which shall be no  more than 64 characters. |
| order\_time | Date | √ | Parameter Name: Order time of merchant  Format: yyyy-MM-dd HH:mm:ss 2013-11-01 12:34:58 |
| order\_amount | Number(13,2) | √ | Parameter Name: Total amount of merchant orders  The total amount of such order shall be in unit of RMB, and to two digits after the decimal point  For example: 12.01 |
| bank\_code | String(10) | × | Parameter Name: Bank code  By using it, merchant website can be redirected to bank page directly. |
| redo\_flag | Int(1) | × | Parameter Name: Whether to allow repeat orders  The value 1 are not allowed to repeat submit merchant order number;  The value 0 or empty when allowed to repeat submit order number |
| product\_name | String(100) | √ | Parameter Name: Product name  No more than 100 characters |
| product\_code | String(60) | × | Parameter Name: Product code  No more than 60 characters |
| product\_num | Number(10) | × | Parameter Name: Product number  Must be number |
| product\_desc | String(300) | × | Parameter Name: Product description  No more than 300 characters |
| extra\_return\_param | String(100) | × | Parameter Name: Public use return parameter  If such parameter is delivered in payment request, then such parameter will be returned when notifying the merchant that the payment is successfully made. |
| extend\_param | String | √ | Parameter Name: Public use service extend parameter  Format of parameter: parameter name1^parameter  value|parameter name2^parameter value2  Multiple data shall be separated with"|"  For example: name^ZhangSan|sex^Male  Cross-border (Outside China) merchants are required, non-cross-border (Inside China) merchant can have the option to choose. |
| show\_url | String(200) | × | Parameter Name: Product show URL  No more than 200 characters |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Parameter Name** | **Type** | **Required** |
| Information of receiver | | | |
| ship\_to\_name | Name of receiver | String(50) | × |
| ship\_to\_email | E-mail of receiver | String(60) | × |
| ship\_to\_phone | Telephone of receiver | String(20) | × |
| ship\_to\_state | Province of receiving address | String(100) | × |
| ship\_to\_city | City of receiving address | String(100) | × |
| ship\_to\_street | Detailed address of receiver | String(200) | × |
| ship\_to\_zip | Zip code of receiving address | String(20) | × |
| Information of Customer | | | |
| customer\_email | Customer e-mail | String(100) | × |
| customer\_name | Customer name  This parameter participates in cross-border authentication verification. Dindin Merchant have the option to choose. For non Dindin and cross-border (Outside China) merchant this is required, for non Dindin and non-cross-border (Inside China) merchant can have the options to choose | String(50) | √ |
| customer\_phone | Customer telephone or cell phone number | String(20) | × |
| customer\_state | Province of customer | String(100) | × |
| customer\_city | City of customer | String(100) | × |
| customer\_street | Detailed address of customer | String(200) | × |
| customer\_zip | Zip code of address | String(20) | × |
| customer\_cardNumber | Consumer Card | String(50) | × |
| customer\_idNumber | Consumer ID number  This parameter participates in cross-border authentication verification. Dindin Merchant have the option to choose. For non Dindin and cross-border (Outside China) merchant this is required, for non Dindin and non-cross-border (Inside China) merchant can have the options to choose | String(30) | √ |
| orders\_info | Related information storage sub-orders, limit 20 products | String(4000) | × |

## 2.2. Payment result notification interface

### 2.2.1. Description

When the consumer finished the transaction, both return\_url and notify\_url (you can find them on request parameters) will receive notice from QuickPay server.

QuickPay server won’t send notice if transaction failed. Merchants also can set up the return\_url and notify\_url on QuickPay merchant system as flows:



It means the setting up will take effect when you see “OFF” and QuickPay server will send notice to the url address if you have set up on admin panel. Admin panel have the priority than the request.

### 2.2.2. Definition of interface parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | required | **Notes** |
| **Basic parameter** | | | |
| merchant\_code | String(10) | √ | Parameter Name: Merchant ID  QuickPay assigned to the merchant's identity when they establish the cooperation relations. |
| notify\_type | String(14) | √ | Parameter Name: Notification type  1. Page return notification: page\_notify  2. Asynchronous notification of server: offline\_notify |
| notify\_id | String(100) | √ | Parameter Name: Notify ID  This parameter is no longer used for verified, but still retained.  It'll return the data in form of 32 characters, which as shown below:  e722dceae317466bbf9cc5f1254b8b0a |
| interface\_version | String(10) | √ | Parameter Name: Interface version  Fixed value: V3. 0 (Capital) |
| sign\_type | String(10) | √ | Parameter Name: Signature type  support value: RSA, RSA-S, not participate in signature |
| sign | String | √ | Parameter Name: Signature  QuickPay sample code will tell you how to use it |
| **Service parameter** | | | |
| order\_no | String(64) | √ | Parameter Name: Merchant's order No.  1.This order number is generated by merchant’s website, which should be uniqueness.  2.It's composed of letters, numbers or underscores, which shall be no  more than 64 characters. |
| order\_time | Date | √ | Parameter Name: Order time of merchant  Format: yyyy-MM-dd HH:mm:ss 2013-11-1 12:34:58 |
| order\_amount | Number(13,2) | √ | Parameter Name: Total amount of merchant orders  The total amount of such order shall be in unit of RMB, and to two digits after the decimal point  For example: 12.01 |
| extra\_return\_param | String(100) | × | Parameter Name: Public use return parameter  If such parameter is delivered in payment request, then such parameter will be returned when notifying the merchant that the payment is successfully made. |
| trade\_no | String(30) | √ | Parameter Name: QuickPay transaction order code  For example: 1000004817 |
| trade\_time | Date | √ | Parameter Name: QuickPay transaction order time  Format: yyyy-MM-dd HH:mm:ss 2013-11-1 12:34:58 |
| trade\_status | String(7) | √ | Parameter Name: Transaction status  SUCCESS, the transaction is successful |
| bank\_seq\_no | String | × | Parameter Name: Internet bank transaction sequence number  For example: 2013060911235456 |

## 2.3. Single transaction query interface

### 2.3.1. Description

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

### 2.3.2. Definition of query parameters

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

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | required | Notes |
| **Basic parameter** | | | |
| service\_type | String(18) | √ | Parameter Name: Service type  It has a fixed value: single\_trade\_query |
| merchant\_code | String(10) | √ | Parameter Name: Merchant ID  QuickPay assigned to the merchant's identity when they establish the cooperation relations |
| interface\_version | String(10) | √ | Parameter Name: Interface version  Fixed 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) | √ | Parameter Name: Merchant's order No.  1.This order number is generated by merchant’s website, which should be uniqueness.  2.It's composed of letters, numbers or underscores, which shall be no  more than 64 characters. |
| trade\_no | String(30) | × | Parameter Name: QuickPay transaction order code  For example: 1000004817 |

### 2.3.3. Definition of response parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | required | Notes |
| **Basic parameter** | | | |
| is\_success | String(1) | √ | This parameter just means whether the query is successful:  If query is successful, is\_success = 'T'  If query is unsuccessful, is\_success = 'F' |
| sign\_type | String(10) | √ | Parameter Name: Signature type  support value: RSA, RSA-S, not participate in signature |
| sign | String | √ | Parameter Name: Signature |
| error\_code | string | × | Only return error code when query is failed.  For example: 128600 |
| **Service parameter** | | | |
| merchant\_code | String(10) | √ | Parameter Name: Merchant ID  QuickPay assigned to the merchant's identity when they establish the cooperation relations |
| order\_no | String(64) | √ | Parameter Name: Merchant's order No.  1.This order number is generated by merchant’s website, which should be uniqueness.  2.It's composed of letters, numbers or underscores, which shall be no more than 64 characters |
| order\_time | Date | √ | Parameter Name: Order time of merchant  Format: yyyy-MM-dd HH:mm:ss 2013-11-1 12:34:58 |
| order\_amount | Number(13,2) | √ | Parameter Name: Total amount of merchant orders  The total amount of such order shall be in unit of RMB, and to two digits after the decimal point  For example: 12.01 |
| trade\_no | String(30) | √ | Parameter Name: QuickPay transaction order code  For example: 1000004817 |
| trade\_time | Date | √ | Parameter Name: QuickPay transaction order time  Format: yyyy-MM-dd HH:mm:ss 2013-11-1 12:34:58 |
| trade\_status | String(7) | √ | Parameter Name: Transaction status  1. SUCCESS, the transaction is successful  2. UNPAY, the transaction is failed |

The xml data which QuickPay returns is as shown below:

In the case of successful query:

|  |
| --- |
| <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> |

In the case of unsuccessful query:

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

## 2.4. Appendix

### 2.4.1. Supported bank code

bank code comparison table is as follows:

|  |  |
| --- | --- |
| **bank name** | **bank code** |
| Agricultural Bank of China | ABC |
| Industrial and Commercial Bank of China | ICBC |
| China Construction Bank | CCB |
| Bank of Communication | BCOM |
| Bank of China | BOC |
| China Merchants Bank | CMB |
| China Minsheng Bank | CMBC |
| China Ever Bright Bank | CEBB |
| Bank of Beijing | BOB |
| Bank of Shanghai | SHB |
| Bank of Ningbo | NBB |
| Hua Xia Bank | HXB |
| China Industrial Bank | CIB |
| Postal Saving Bank of China | PSBC |
| Shenzhen Ping An Bank | SPABANK |
| Shanghai Pudong Development Bank | SPDB |
| China Citic Bank | ECITIC |
| Bank of HangZhou | HZB |
| China Guangfa Bank | GDB |

### 2.4.2. 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 |
| csp.orderService.order-is-already-exist | Meaning: order already exists  Reason: order\_no is submitted repeatedly |
| 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: unknown 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 cannot 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 cannot 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 cannot be null with the order\_time |
| 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 cannot 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 cannot exceed 13 figures |
| PRODUCT\_NAME\_IS\_NULL | Reason: it cannot be null with the product\_name |
| PRODUCT\_NAME\_IS\_TOO\_LONG | Reason: the length of procuct\_name cannot 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 cannot exceed 60 characters |
| PRODUCT\_DESC\_IS\_TOO\_LONG | Reason: the length of procuct\_desc cannot exceed 300 characters |
| EXTRA\_RETURN\_PARAM\_IS\_TOO\_LONG | Reason: the length of extra\_return\_param cannot 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 cannot be null with the merchant\_code |
| INTERFACE\_VERSION\_IS\_NULL | Reason: it cannot be null with the interface\_version |
| ILLEGAL\_INTERFACE\_VERSION | Reason: incorrect interface\_version, it must be V3.0 |
| SIGN\_TYPE\_IS\_NULL | Reason: it cannot 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 cannot be null with the sign |
| ILLEGAL\_SIGN | Meaning: the signature is incorrect; signature data is inconsistent  Possible reason:  1)Some required parameters don’t participate in signature.  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 |
| CONCUMER\_NAME\_IS\_NULL | It cannot be null with consumer name |
| CONCUMER\_IDNO\_IS\_NULL | It cannot be null with consumer idno |
| CONCUMER\_IDNO\_IS\_INVALID | Wrong consumer idno |
| ILLEGAL\_CONCUMER\_IDNO | Consumer idno is wrong format or invalid |
| INVALID\_ID\_CARD | The Card ID does not pass the verification |
| RECEIVER\_IDNO\_IS\_INVALID | Wrong consignee idno |
| ILLEGAL\_RECEIVER\_IDNO | Consignee idno is wrong format or invalid |