

**Omipay Payment**

**Technical Documentation**

|  |  |
| --- | --- |
| Author | Max.Wong |
| Create Date | 2018-7-20 |
| Version | v2.0.6 |

|  |  |  |
| --- | --- | --- |
| Modify Date | Modifier | Description |
| 2018-9-26 | Max.Wong | Add Wechat MiniProgram Payment |

Catalog

[**1．** **Scenes of Payment** 2](#_Toc525724731)

[1.1 In-Store Payment 2](#_Toc525724732)

[a) Merchant QR Code Payment 2](#_Toc525724733)

[b) POS Terminal Payment 2](#_Toc525724734)

[1.2 Online Payment 2](#_Toc525724735)

[a) Web Store Payment 2](#_Toc525724736)

[b) WeChat Store Payment 2](#_Toc525724737)

[c) Alipay Wap/Web Payment 2](#_Toc525724738)

[d) In-App Payment 2](#_Toc525724739)

[e) WeChat MiniProgram Payment 3](#_Toc525724740)

[**2．** **Rules** 3](#_Toc525724741)

[2.1 Protocol Rules 3](#_Toc525724742)

[2.2 Parameter Rules 3](#_Toc525724743)

[2.3 Signature 3](#_Toc525724744)

[2.4 Error Codes 5](#_Toc525724745)

[3． API List 5](#_Toc525724746)

[2.1 Get current exchange rate 5](#_Toc525724747)

[2.2 Make QRCode Order 6](#_Toc525724748)

[2.3 Make JSAPI Order 11](#_Toc525724749)

[2.4 Make Scan Order 16](#_Toc525724750)

[2.5 Make Online Order (Alipay Online) 19](#_Toc525724751)

[2.6 Make APP Order (Alipay Online) 21](#_Toc525724752)

[2.7 Make WeChat MiniProgram Order (WeChat) 22](#_Toc525724753)

[2.8 Query Order Status 25](#_Toc525724754)

[2.9 Payment Success Notification 27](#_Toc525724755)

[2.10 Refund 28](#_Toc525724756)

[2.11 Query Refund 30](#_Toc525724757)

1. **Scenes of Payment**

Omipay supports the payment catalogs as below:

1. In-Store Payment
2. Merchant QR Code Payment

Merchant could present two static QR Codes to the customer ,namely the aggregation payment QR code and the UnionPay payment QR code. The customer should scan aggregation payment QR Code via Alipay/WeChat App/ or any other app which integrates the browser on smartphone and should scan Unionpay payment QR code via app which supports Unionpay. Then customer should input the essential information on the payment page to complete the payment.

Merchant could download the QR Code on Dashboard and print out it.

Merchant QR Code Payment don’t need any API integration.

1. POS Terminal Payment

The customer shows their payment QRCode / barcode in Alipay/WeChat/ App which supports Unionpay, and then the merchant scans it via POS terminal to finish the payment.

The introduction of this scene is Chapter

In other case, if the merchant’s POS terminal has a screen, then merchant can generate a dynamic QRCode and show it in the screen, customer scans it via Alipay/WeChat/App which supports Unionpay to complete the payment.

The introduction of this scene is Chapter

1. Online Payment
2. Web Store Payment

When customer access the store web site on PC/tablet, merchant can generate aggregation payment QRCode or Unionpay payment QR code,and then present it out. Customer scan aggregation payment QRCode via Alipay/WeChat App and scan Unionpay payment QR code via App which supports Unionpay,and then complete the payment directly.

The introduction of this scene is Chapter

1. WeChat Store Payment

When customer access the web store in WeChat browser, merchant can get a payment URL from Omipay, and redirect the WeChat browser to this URL, then customer complete the payment in the payment page.

The introduction of this scene is Chapter

1. Alipay Wap/Web Payment

When customer access the store in smartphone web browser (e.g. Safari/Chrome), they should pay directly via Alipay by this way.

The introduction of this scene is Chapter

1. In-App Payment

Merchant should integrate a SDK in your own Android/iOS app to let customer pay directly in it.

The introduction of this scene is Chapter

1. WeChat MiniProgram Payment

If you have a WeChat MiniProgram, you should call this API to finish the payment.

The introduction of this scene is Chapter

1. Credit card Payment

If you have a web store, and want to integrate the credit card checkout counter in it, you should see Chapter

\*Applications that support UnionPay payments include the following:

China UnionPay Quick Pass / Bank of China application / China Merchants Bank application

1. **Rules**
	1. Protocol Rules

|  |  |
| --- | --- |
| Transafer Mode | HTTPS |
| Submit Mode | POST/GET |
| Receive Data Format | Json |
| Text Encoding | UTF-8 |
| Sign Algorithm | MD5 |
| Data Sending Format | Sending data as URL parameter |
| Http Header | ContentType=application/json |
| Expiring | Signature expires in 5 minutes |

* 1. Parameter Rules

1）Amount

 Amount uses the smallest unit of transaction currency, must be integer value.

e.g. amount 100 of currency AUD would be equivalent to A$1.00.

2）Currency

 The only currency type we support is AUD (Australian dollar).

3）Time zone

 System time zone is AEST (Australia East Standard Time).

4）Timestamp

 Convert current UTC time to milliseconds.

5）Nonce string

 The random string witch length in 10-32 characters，allows the characters in a-z | A-Z | 0-9.

6）Payment platform

 Omipay supports the payment platforms below:

|  |  |
| --- | --- |
| Platform name | The value of parameter *platform*  |
| Wechat Pay | WECHATPAY |
| Alipay | ALIPAY |
| Unionpay International | UPI |
|  |  |

7） language

 Omipay WebAPI supports multi-language. Set the *Accept-Language* value in HTTP　request header to handle it:

|  |  |
| --- | --- |
| Language | The value of parameter *language* |
| English | en-US |
| Simplified Chinese (Default) | zh-CN |

* 1. Signature

Every request to Omipay Payment API must be signed, the sign algorithm is as below:

 1）Parameters

 The parameters for verifying as below:

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| m\_number | String | Merchant number, can be found in merchant Dashboard(<https://dashboard.omipay.com.cn>) |
| timestamp | Long | Time stamp, refer to *Timestamp* in *Parameter Rules* |
| nonce\_str | String | Random string, refer to *Nonce string* in *Parameter Rules* |
| secret\_key | String | API Secret key of merchant, acquired from merchant Dashboard. This parameter is for private, don’t present in any public place. |

 2）Generate signature

 ① Concatenate all the parameters above like this:

 *m\_number&timestamp&nonce\_str&secret\_key*

Make sure it’s in the correct order.

② Use the concatenated string to make a signature string with MD5 algorithm, and then turn it into uppercase.

③ Put the signature string as a part of the query parameters, and then call the web api.

e.g.:

The parameters below:

|  |  |
| --- | --- |
| m\_number | 123456 |
| timestamp | 1482812036067 |
| nonce\_str | 313644f42ecd4758b5e23b80e86efdc4 |
| secret\_key | 0af61531c6c04ac4ac910d0cd59e6238 |

The concatenated string below:

123456&1482812036067&313644f42ecd4758b5e23b80e86efdc4&0af61531c6c04ac4ac910d0cd59e6238

Here are some sample codes of making the sign string:

|  |
| --- |
| **C#** |
| var originString = “123456&1482812036067&313644f42ecd4758b5e23b80e86efdc4&0af61531c6c04ac4ac910d0cd59e6238”;    var md5 = MD5.Create();    var bs = md5.ComputeHash(Encoding.UTF8.GetBytes(originString));   var sb = new StringBuilder();   foreach (byte b in bs)  {     sb.Append(b.ToString("x2"));  }   originString = sb.ToString().ToUpper();  |
| **PHP** |
| $originString = ‘123456&1482812036067&313644f42ecd4758b5e23b80e86efdc4&0af61531c6c04ac4ac910d0cd59e6238’;$sign = strtoupper(md5($originString)); |

Get the signature string:

*sign=”516A3B52F9C8897F52239B19CD8A499”*

and then put it into the parameter string, we can get the parameter below:

*m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499*

Use this string for signature verifying.

* 1. Error Codes

|  |  |
| --- | --- |
| Error Code | Description |
| MERCHANTNO\_INVALID | The merchant number is wrong, please check the corresponding merchant number from the merchant system. |
| SIGN\_ERROR | Signature verification failed. Please check if the signature is correct |
| SYSTEM\_ERROR | Server error |
| MERCHANTNO\_DISABLE | Merchant is disabled |
| BRANCH\_INVALID | Branch do not exist |

1. API List
2. Get current exchange rate
* Description：

Get the current exchange rate of AUD to CNY, which is provided by WeChat/Alipay, refreshes in a day. This exchange rate is only for reference.

* API address:

<https://www.omipay.com.cn/omipay/api/v2/GetExchangeRate>

* Parameters (except the sign verifying parameters, all the APIs are as the same.)

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| currency | String | AUD | The currency code which follows the ISO-4217 standard.  |
| base\_currency | String | CNY |
| platform | String | WECHATPAY | Not mandatory, defaults ***WECHATPAY*** |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | When execute succeed, returns SUCCESS,it shows the result of API execution. |
| rate | Decimal | 5.01540000 | Current exchange rate |
| platform | String | WECHATPAY |  |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | When execute failed, returns FAIL  |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | System error. | Description of error |

* Error code list:

|  |  |
| --- | --- |
| Error Code | Description |
| CURRENCY\_ERROR | The currency type error may be the unavailability of the exchange rate of the two currency types or the input error of the currency type. |

* Request example:

<https://www.omipay.com.cn/omipay/api/v2/GetExchangeRate>? m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&currency=AUD&base\_currency=CNY

* Success returns example:

{

 "return\_code": "SUCCESS",

 "rate": 5.01540000

}

* Fail returns example

{

 "return\_code": "FAIL",

 "error\_code": "SIGN\_TIMEOUT",

 "error\_msg": "Sign time out."

}

1. Make QRCode Order
* Description:

QRCode order suits for the scene of PC web store payment. Merchant can call this API to get a string for payment, and then show this string to customer as a QRCode; Customer would scan this QRCode with WeChat/Alipay/App which supports Unionpay to finish the pay.

\*Applications that support UnionPay payments include the following:

China UnionPay Quick Pass / Bank of China application / China Merchants Bank application

* Sequence diagram:

* API address:

<https://www.omipay.com.cn/omipay/api/v2/MakeQROrder>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_name | String | Test\_Order | The simple order information. |
| currency | String | AUD | Currency code of the order. |
| amount | int | 100 | Order amount in the base unit of the currency of order. Refers to section 1.2.e.g. if currency = AUD, then 100 means A$1.00 |
| notify\_url | String |  | Notification URL for transaction success. When this order is pay succeed, will send a notification to such URL. Refers to section 2.6.  |
| out\_order\_no | String |  | Out order number. The notification data of transaction would include this field. So, it best be unique, in order to identify the order. |
| platform | String | WECHATPAY | payment platform of this order. |
| o\_number | String | 01 | number of the transaction organization. |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS |
| order\_no | String |  | Omipay order number, 32 characters length.  |
| Qrcode | String |  | String for QRCode.  |
| Pay\_url | String |  | The url of common payment page. When order is created, may redirect to this url to show the payment QRCode.\* Access this page is also needs signing, the sign algorithm is same to other methods. |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | System error. | Description of error |

* Error Code list:

|  |  |
| --- | --- |
| Error Code | Description |
| WEBSTORE\_NOT\_EXISTS | The merchant did not register an online store.The error will occur if the merchant does not register with Omipay. |
| AMOUNT\_INVALID | The amount is incorrect.This error will be prompted when the amount is greater than the maximum amount of a single transaction allowed by the payment platform, or the amount converted by the exchange rate is less than the minimum amount paid by the payment platform |
| CURRENCY\_ERROR | The currency type error may be the unavailability of the exchange rate of the two currency types or the input error of the currency type |
| OUT\_TRANSACTION\_PAID | Out order number has been paid |
| TRANSACTION\_TIMEOUT | Order has expired |

* Request example:

<https://www.omipay.com.cn/omipay/api/v2/MakeQRCode>? M\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&order\_name=Test\_Order&amount=100&notify\_url=http://your\_site.com/receive\_notify.html&out\_order\_no=SEORD000001

* Addition comments:

The flow of QRCode payment:

1. Merchant create order in Web or PC Online-Store;
2. Call Omipay MakeQROrder API;
3. Generate QRCode with the *qrcode* string of API returns, and then show it to customer in application interface;

OR: Redirect to pay\_url and show the QRCode.

1. Customer scan this QRCode with WeChat/Alipay APP on mobile phone.
2. Customer finish the payment.
3. Omipay send transaction notification data to merchant.
* Success returns example:

{

 “return\_code”: “SUCCESS”,

 “order\_no”: “bc112874260946a2af2b7107825e6ce2”,

 “qrcode” : “………………. “

}

* Fail returns example:

{

 “return\_code”: “FAIL”,

 “error\_code”: “SIGN\_TIMEOUOT”,

 “error\_msg”: “Sign timeout.”

}

* Description of payment page:
1. When API returns, may lead the browser redirect to the *pay\_url* address. Which page shows the information of order and the payment QRCode. Customer scans this QRCode with Alipay/Wechatpay app in mobile phone to finish the transaction. When payment is done, browser would redirect to the *redirect\_url* address which in the call parameter.
2. For accessing the *pay\_url*, the following parameters are required.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| m\_number | String |  |  |
| timestamp | String | 1482812036067 | Time stamp, refer to Timestamp in *Parameter Rules* |
| noncestring | String |  | Random string, refer to Nonce string in Parameter Rules |
| sign | String |  | Sign string |
| redirect\_url | String |  | Browser will redirect to such url when payment done. |

1. Make JSAPI Order
* Description:

JSAPI order suits for the scenes below:

1. WeChat web browser based payment. Merchant could call this API to get a URL for payment, and then lead customer to access this URL to pay.
2. Common pay QRCode for transaction. Merchant could use the pay\_url generate a QRCode, and show it to customer. Customer could scan this code with either Wechat or Alipay client app on mobile phone to pay.
* Sequence diagram:

* API address:

<https://www.omipay.com.cn/omipay/api/v2/MakeJSAPIOrder>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_name | String | TEST\_ORDER | The simple order information. |
| Currency | String | AUD | Currency code of the order. |
| Amount | int | 100 | Order amount in the base unit of the currency of order. Refers to section 1.2.e.g. if currency = AUD, then 100 means A$1.00 |
| notify\_url | String |  | Notification URL for transaction success. When this order is pay succeed, will send a notification to such URL. Refers to section 2.6.  |
| redirect\_url | String |  | Redirecting URL for transaction success. When this order is pay succeed, web browser will auto redirect to such URL. |
| Out\_order\_no | String |  | Out order number. The notification data of transaction would include this field. So, it best be unique, in order to identify the order. |
| direct\_pay | Int | 1 | Not mandatory, if this parameter transferred, the order will be paid directly when the pay\_url is open. |
| Show\_pc\_pay\_url | Int | 1 | Not mandatory, if this parameter transferred, API response object would include a parameter that shows a PC payment page URL . You can redirect to this URL and show the order OQCode to customer.See ***Description of payment page*** section to get more details. |
| o\_number | String | 01 | Not mandatory, number of transaction organization. |
| pos\_no | String |  | Not mandatory, POS device number. |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS |
| order\_no | String |  | Omipay order number, 32 characters length.  |
| pay\_url | String |  | URL for payment. Merchant should lead user to access this URL.  |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | System error | Description of error |

* Error Code list:

|  |  |
| --- | --- |
| Error Code | Description |
| WEBSTORE\_NOT\_EXISTS | The merchant did not register an online store.The error will occur if the merchant does not register with Omipay. |
| AMOUNT\_INVALID | The amount is incorrect.This error will be prompted when the amount is greater than the maximum amount of a single transaction allowed by the payment platform, or the amount converted by the exchange rate is less than the minimum amount paid by the payment platform |
| CURRENCY\_ERROR | The currency type error may be the unavailability of the exchange rate of the two currency types or the input error of the currency type |
| OUT\_TRANSACTION\_PAID | Out order number has been paid |
| TRANSACTION\_TIMEOUT | Order has expired |
| POS\_NOT\_REGISTERED | Merchant POS terminal is not registered |
| POS\_NOT\_BIND | The Pos machine is not tied to the store |

* Request example:

<https://www.omipay.com.cn/omipay/api/v2/MakeJSAPIOrder>? m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&order\_name=TEST\_ORDER&redirect\_url=http://your\_site.com/redirect\_to.html&amount=100&notify\_url=http://your\_site.com/receive\_notify.html&out\_order\_no=SEORD000001&language=EN

* Success returns example:

{

 "return\_code": "SUCCESS",

 "order\_no": "bc112874260946a2af2b7107825e6ce2",

 "pay\_url" : "………………. "

}

* Fail returns example:

{

 "return\_code": "FAIL",

 "error\_code": "SIGN\_TIMEOUOT",

 "error\_msg": "Sign timeout."

}

* Additional:

JSAPI order request should includes redirect\_url parameter. And when customer finished the payment, mobile browser would redirect to such URL.

* Description of payment page:
1. If the request parameter includes the parameter ***show\_pc\_pay\_url*,** and its value is 1 (show\_pc\_pay\_url = 1), then the API response data would contain a property as ***pc\_pay\_url,*** which is the pay ur l for PC browser. Merchant may lead the PC browser redirect to such address, which page show the information of order an the payment QRCode. Customer scans this QRCode with Alipay/Wechatpay app in mobile phone to finish the transaction. When payment is done, browser would redirect to the *redirect\_url* address which in the call parameter.
2. The PC payment page is like:



1. For accessing the ***pc\_pay\_url***, the following parameters are required.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| m\_number | String |  |  |
| timestamp | String | 1482812036067 | Time stamp, refer to Timestamp in *Parameter Rules* |
| noncestring | String |  | Random string, refer to Nonce string in Parameter Rules |
| sign | String |  | Sign string |
| redirect\_url | String |  | Browser will redirect to such url when payment done. |

1. Make Scan Order
* Description:

Scan order is suits the scene of local payment. Customer shows the Payment code in WeChat Wallet/Alipay/China UnionPay Quick Pass client, merchant scans the code by POS device, AND then call API to create order.

* API address：

<https://www.omipay.com.cn/omipay/api/v2/MakeScanOrder>

* Sequence diagram:

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_name | String | TEST\_ORDER | The simple order information. |
| currency | String | AUD | Currency code of the order. |
| amount | int | 100 | Order amount in the base unit of the currency of order. Refers to section 1.2.e.g. if currency = AUD, then 100 means A$1.00 |
| notify\_url | String |  | Notification URL for transaction success. When this order is pay succeed, will send a notification to such URL. Refers to section 2.6. |
| qrcode | String |  | Customer’s payment QRCode.  |
| pos\_no | String |  | POS Device number. |
| out\_order\_no | String |  | Out order number. The notification data of transaction would include this field. So, it best be unique, in order to identify the order. |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS：Pay succeed  |
| order\_no | String |  | Omipay order number, 32 characters length. |
| out\_order\_no | String |  | Out order number. |
| order\_time | String |  | Order pay time, in format of yyyyMMddHHmmss |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | Server error | Description of error |
| order\_no | String |  | If error\_code = “USERPAYING”, then returns this field. Refers to Addition comments. |

* Error Code list:

|  |  |
| --- | --- |
| Error Code | Description |
| CURRENCY\_ERROR | The currency type error may be the unavailability of the exchange rate of the two currency types or the input error of the currency type |
| AMOUNT\_INVALID | The amount is incorrect.This error will be prompted when the amount is greater than the maximum amount of a single transaction allowed by the payment platform, or the amount converted by the exchange rate is less than the minimum amount paid by the payment platform |
| USERPAYING | The user is paying |
| ORDER\_PAID | The payment order has been paid |
| POS\_NOT\_REGISTERED | Merchant POS terminal is not registered |
| POS\_NOT\_BIND | The Pos terminal is not tied to the branch |
| CARD\_NOT\_SUPPORT | Type of card not supported |
| ORDER\_CLOSED | Order closed |
| OUT\_TRANSACTION\_PAID | Out order number has been paid |
| TRANSACTION\_TIMEOUT | Order has expired |

* Request example:

<https://www.omipay.com.cn/omipay/api/v2/MakeScanOrder>? m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&order\_name=Test\_Order&amount=100&&notify\_url=http://your\_site.com/receive\_notify.html&qrcode=123abc123&pos\_no=posno\_123&out\_order\_no=SEORD000001&language=EN

* Addition comments：

The flow of Scan payment:

1. Merchant create the order in POS System, and show the amount of such order to customer;
2. Customer show the Payment QRCode of WeChat Wallet;
3. Merchant scan the QRCode with POS Device, and call *MakeScanOrder* API to create payment order.
4. Omipay server solves the request, meanwhile, merchant waits for the result of order.
5. If Omipay returns the *SUCCESS* code, the flow finish; else if return\_code is FAIL and *error\_code* is not *USERPAYING*, the flow fails; else into step 6);
6. The *return\_code* is *USERPAYING* means customer had to input payment password. （Whether to input the password depends on the security policy of WeChat Payment/Alipay）If customer had to input password, merchant should to hold this order and waiting for the response from Omipay server. When customer finished the payment, Omipay would call *notify\_url* and send the payment information. Merchant could also call *QueryOrder* API to get the status of payment transaction.
7. Make Online Order (Alipay Online/POLi)
* Description:

Online order is suits the scene of pc(web eg. Chrome/IE)/mobile

(wap eg. Safari) browser payment.

* API address：

<https://www.omipay.com.cn/omipay/api/v2/MakeOnlineOrder>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_name | String | TEST\_ORDER | The simple order information. |
| currency | String | AUD | Currency code of the order. |
| amount | int | 100 | Order amount in the base unit of the currency of order. Refers to section 1.2.e.g. if currency = AUD, then 100 means A$1.00(The poli payment amount must be greater than A$1.00) |
| out\_order\_no | String |  | Out order number. The notification data of transaction would include this field. So, it best be unique, in order to identify the order. |
| notify\_url | String |  | Notification URL for transaction success. When this order is pay succeed, will send a notification to such URL. Refers to section 2.6. |
| return\_url | String |  | Not mandatory, when transaction finished, browser will redirect to this URL synchronized. |
| type | String | web/wap | PC browser: webmobile browser: wap |
| o\_number | String | 01 | Not mandatory, number of transaction organization. |
| platform | String | ALIPAYONLINE | ALIPAYONLINEPOLI(Not mandatory) ALIPAYONLINE is selected by default |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS：Pay succeed  |
| order\_no | String |  | Omipay order number, 32 characters length. |
| pay\_url | String |  | URL for payment. Merchant should lead user to access this URL.  |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | Server error | Description of error |

* Error Code list:

|  |  |
| --- | --- |
| Error Code | Description |
| WEBSTORE\_NOT\_EXISTS | The merchant did not register an online store.The error will occur if the merchant does not register with Omipay. |
| AMOUNT\_INVALID | The amount is incorrect.This error will be prompted when the amount is greater than the maximum amount of a single transaction allowed by the payment platform, or the amount converted by the exchange rate is less than the minimum amount paid by the payment platform |
| CURRENCY\_ERROR | The currency type error may be the unavailability of the exchange rate of the two currency types or the input error of the currency type |

1. Make APP Order
* Description:

App order is suits the scene of iOS/Android App payment.

* API address：

<https://www.omipay.com.cn/omipay/api/v2/MakeAppOrder>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_name | String | TEST\_ORDER | The simple order information. |
| currency | String | AUD | Currency code of the order. |
| amount | int | 100 | Order amount in the base unit of the currency of order. Refers to section 1.2.e.g. if currency = AUD, then 100 means A$1.00(The poli payment amount must be greater than A$1.00) |
| notify\_url | String |  | Notification URL for transaction success. When this order is pay succeed, will send a notification to such URL. Refers to section 2.6. |
| out\_order\_no | String |  | Out order number. The notification data of transaction would include this field. So, it best be unique, in order to identify the order. |
| platform | String | ALIPAYONLINE | ALIPAYONLINE/WECHATPAY/POLI |
| o\_number | String | 01 | Not mandatory, number of transaction organization. |
| app\_id | String | wxc571a1u199z0qc240 | In some payment platform, such as WechatPay, it identifies the app. |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS：Pay succeed  |
| order\_no | String |  | Omipay order number, 32 characters length. |
| order\_string | String |  | Payment string for Alipay APP calling. |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | Server error | Description of error |

* Error Code list:

|  |  |
| --- | --- |
| Error Code | Description |
| WEBSTORE\_NOT\_EXISTS | The merchant did not register an online store.The error will occur if the merchant does not register with Omipay. |
| AMOUNT\_INVALID | The amount is incorrect.This error will be prompted when the amount is greater than the maximum amount of a single transaction allowed by the payment platform, or the amount converted by the exchange rate is less than the minimum amount paid by the payment platform |
| CURRENCY\_ERROR | The currency type error may be the unavailability of the exchange rate of the two currency types or the input error of the currency type |

* Mobile App develop reference:

iOS:

|  |  |
| --- | --- |
| WECHAT: | https://pay.weixin.qq.com/wiki/doc/api/app/app\_jw.php?chapter=11\_1 |
| ALIPAY: | https://global.alipay.com/service/app\_split/6 |

Android:

|  |  |
| --- | --- |
| WECHAT: | https://pay.weixin.qq.com/wiki/doc/api/app/app\_jw.php?chapter=11\_1 |
| ALIPAY: | https://global.alipay.com/service/app\_split/7 |

1. Make WeChat MiniProgram Order (WeChat)
* Description:

If merchant have a WeChat MiniProgram, then you should call this API to finish the payment.

 \* The introduction of WeChat MiniProgram, you should see here:

<https://mp.weixin.qq.com/cgi-bin/wx>

* Sequence Diagram:

* API Address:

<https://www.omipay.com.cn/omipay/api/v2/MakeAppletOrder>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_name | String | TEST\_ORDER | The simple order information. |
| currency | String | AUD | Currency code of the order. |
| amount | int | 100 | Order amount in the base unit of the currency of order. Refers to section 1.2.e.g. if currency = AUD, then 100 means A$1.00 |
| notify\_url | String |  | Notification URL for transaction success. When this order is pay succeed, will send a notification to such URL. Refers to section 2.6.  |
| out\_order\_no | String |  | Out order number. The notification data of transaction would include this field. So, it best be unique, in order to identify the order. |
| o\_number | String | 01 | Not mandatory, number of transaction organization. |
| app\_id | String | wx20oq811f9q | The AppId of WeChat MiniProgram |
| customer\_id | String | o89hJw3wUb | The customer OpenId obtained from WeChat. |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS |
| order\_no | String |  | Omipay order number, 32 characters length.  |
| timeStamp | String |  | Necessary field to call WeChat Payment in frontend. |
| nonceStr | String |  | Necessary field to call WeChat Payment in frontend. |
| Package | String |  | Necessary field to call WeChat Payment in frontend. |
| SignType | String |  | Necessary field to call WeChat Payment in frontend. |
| paySign | int |  | Necessary field to call WeChat Payment in frontend. |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | system error | Description of error |

* Error code list:

|  |  |
| --- | --- |
| Error Code | Description |
| PARAMETER\_INVALID | Invalid parameters. |
| SIGN\_TIMEOUT | Time stamp expires. |
| MERCHANTNO\_INVALID | Merchant number is invalid. |
| SIGN\_ERROR | Signature check failed.  |
| SYSTEM\_ERROR | System unhandled error. |
| PERMISSION\_ERROR | Permission error, Occurs when the order number is not belongs to current merchant. |

* Request example:

<https://www.omipay.com.au/omipay/api/v2/MakeAppletOrder?amount=1&order_name=aaaa&out_order_no=SEORD000012&nonce_str=uXWTAw0g3q5HOHjdZW4opDbfp1a7dBOI&m_number=000034&platform=WECHATPAY&notify_url=baidu.com&timestamp=1513242617000&sign=1266C828F67A1C5BD0DC0203405D4480&currency=AUD&app_id=wx20u82a9&customer_id=o98c8sq9xz0>

* Additional:

The flow of integrating Omipay in you WeChat MiniProgram:

1) Merchant should apply a MiniProgram in WeChat Public Platform(<https://mp.weixin.qq.com/>). The owner must be the same to the institution of merchant.

2) Finish the WeChat certification and obtain the AppId of your MiniProgram.

3) Give the AppId to Omipay.

4) Start your technical integration flow.

* The example code of call Payment in MiniProgram front-end:

wx.requestPayment(
{
'timeStamp': 'data.timeStamp',
'nonceStr': 'data.nonceStr',
'package': 'data.package',
'signType': 'data.signType',
'paySign': 'data.paySign',
'success':function(res){},
'fail':function(res){},
'complete':function(res){}
})

1. Query Order Status
* Description:

Merchant could call this API to get the status of order.

* API address:

<https://www.omipay.com.cn/omipay/api/v2/QueryOrder>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_no | String |  | Omipay Order number  |
| language | String | EN |  |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS  |
| result\_code | String | PAID | order status |
| out\_order\_no | String |  | The out order no value which is sent by merchant when calls pay api.  |
| currency | String | AUD | order currency code, which is defined in ISO 4217 standard.Currently, it always be AUD. |
| amount | int | 100 | amount in order currency type.(AUD) |
| order\_time | String |  | Order create time, in format of *yyyyMMddHHmmss* |
| pay\_time | String |  | Order pay time, in format of *yyyyMMddHHmmss*;If order is not payed, the field value is empty string.  |
| exchange\_rate | int |  | Exchange rete of order currency type(AUD) to CNY; 10^8 times of real exchange rate value. |
| pay\_currency | String | CNY | The currency no which customer actually pay. |
| cny\_amount | int |  | amount in *pay\_currency*. |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | system error | Description of error |

* Error code list:

|  |  |
| --- | --- |
| Error Code | Description |
| PARAMETER\_INVALID | Invalid parameters. |
| SIGN\_TIMEOUT | Time stamp expires. |
| MERCHANTNO\_INVALID | Merchant number is invalid. |
| SIGN\_ERROR | Signature check failed.  |
| SYSTEM\_ERROR | System unhandled error. |
| PERMISSION\_ERROR | Permission error, Occurs when the order number is not belongs to current merchant. |

* Request example:

<https://www.omipay.com.cn/omipay/api/v2/QueryOrder>? m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&order\_no=bc112874260946a2af2b7107825e6ce2&language=EN

* Success returns example:

{

 "return\_code": "SUCCESS",

 "result\_code": "PAYED",

 "currency": "AUD",

 "total\_amount": 100,

 "order\_time": "20161227105530",

 "pay\_time": "20161227105602",

 "exchange\_rate": 650000000,

 "cny\_amount": 650

}

* Fail returns example:

{

 "return\_code": "FAIL",

 "error\_code": "SIGN\_TIMEOUOT",

 "error\_msg": "Sign timeout."

}

* Order status list:

|  |  |
| --- | --- |
| Status | Description |
| READY | Order is created, but not payed. |
| PAYING | Customer is now paying.  |
| PAID | Customer is already payed. |
| SETTLED | Order is settled. (Settled by Omipay) |
| CANCELLED | Order is cancelled. |
| FAILED | Customer pay failed. |

1. Payment Success Notification
* Description:

If the order is created with *notify\_url* parameter, Omipay will send a request to this URL when payment success. This request would retries in 3 times unless received correct response.

* Request Rules:

|  |  |
| --- | --- |
| Text Encoding | UTF8 |
| Http Header | Content-Type=application/json |
| Data Sending Format | Data is transferred in request body, in JSON format. |
| Returns Data Type | application/json |

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| return\_code | String | SUCCESS | return SUCCESS when pay succeed. |
| nonce\_str | String |  | Random nonce string. |
| timestamp | Long |  | Time stamp |
| sign | String |  | Signature string. |
| order\_no | String |  | Omipay order no. |
| out\_order\_no | String |  | Out order no. |
| currency | String | AUD | order currency code, which is defined in ISO 4217 standard.Currently, it always be AUD. |
| amount | int | 100 | amount in order currency type.(AUD) |
| order\_time | String |  | Order create time, in format of *yyyyMMddHHmmss* |
| pay\_time | String |  | Order pay time, in format of *yyyyMMddHHmmss*;If order is not payed, the field value is empty string.  |
| exchange\_rate | int |  | Exchange rete of order currency type(AUD) to CNY; 10^8 times of real exchange rate value. |
| pay\_currency | String | CNY | The currency no which customer actually pay. |
| cny\_amount | int |  | amount in *pay\_currency*. |

Notice: Merchant has to verify the sign information when receives the notification to prevent of the third-party attack.

* Merchant returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | Must be SUCCESS. |

1. Refund
* Description:

Merchant could call this method to refund.

* API address:

<https://www.omipay.com.cn/omipay/api/v2/Refund>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| order\_no | String |  | Omipay Order number which is going to be refunded. |
| out\_refund\_no | String |  | Out refund no. |
| amount | Int |  | Refund amount in order currency. |
| language | String | EN |  |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS |
| refund\_no | String |  | Omipay refund bill No. |
| currency | String | AUD | Currency No of refund. |
| amount | int | 100 | Refund amount |
| refund\_time | String |  | Refund time |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | Server error | Description of error |

* Error code list:

|  |  |
| --- | --- |
| Error Code | Description |
| AMOUNT\_OVER\_LIMIT | The amount exceeds the refund limit |
| PERMISSION\_ERROR | Permission error occurred when the refund order number does not belong to the merchant. |

* Request example:

<https://www.omipay.com.cn/omipay/api/v2/Refund>? m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&order\_no=bc112874260946a2af2b7107825e6ce2&out\_refund\_no=xxxxxxx&amount=100&language=EN

* Success returns:

{

 "return\_code": "SUCCESS",

 "refund\_no ": "ADDFGHJKL",

 "currency": "AUD",

 "amount": 100,

 "refund\_time": "20161227105530"

}

* Fail returns:

{

 "return\_code": "FAIL",

 "error\_code": "SIGN\_TIMEOUOT",

 "error\_msg": "Sign timeout."

}

* Additional comments：
1. Refund is based on transaction, each refund bill has hooked with a specific transaction.
2. An order could be refunded multiple times. The total amount of such refunds shouldn’t be more than order amount.
3. The api returns SUCCESS just means refund request is sent to payment platform, and the platform is accepted the request, it doesn’t means the refund flow finished.
4. Query Refund
* Description：

Merchant should call this api to get the status of specific refund.

* API address：

<https://www.omipay.com.cn/omipay/api/v2/QueryRefund>

* Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| refund\_no | String |  | Refund no of Omipay. |
| language | String | EN |  |

* Success returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Example | Description |
| return\_code | String | SUCCESS | SUCCESS |
| result\_code | String | CLOSED | status of refund. |
| out\_refund\_no | String |  | Out order No. |
| currency | String | AUD | Currency No of refund |
| amount | int | 100 | amount in order currency type. |
| refund\_time | String |  | Refund create date. In formats of yyyyMMddHHmmss |
| success\_time | String |  | Actually refund time. In formats of yyyyMMddHHmmss; If refund status is not CLOSED, then returns empty string. |

* Fail returns:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Example | Description |
| return\_code | String | FAIL | FAIL |
| error\_code | String | SYSTEM\_ERROR | Code of error |
| error\_msg | String | Server error | Description of error |

Error code list:

|  |  |
| --- | --- |
| Error code | Description |
| PARAMETER\_INVALID | Invalid parameters. |
| SIGN\_TIMEOUT | Time stamp expires. |
| MERCHANTNO\_INVALID | Merchant number is invalid. |
| SIGN\_ERROR | Signature check failed.  |
| SYSTEM\_ERROR | System unhandled error. |
| PERMISSION\_ERROR | Permission error, Occurs when the order number is not belongs to current merchant. |
| REFUND\_NO\_ERROR | Refund No is not correct. |

* Request example:

https://www.omipay.com.cn/omipay/api/v2/QueryRefund? m\_number=123456&timestamp=1482812036067&nonce\_str=313644f42ecd4758b5e23b80e86efdc4&sign=8516A3B52F9C8897F52239B19CD8A499&refund\_no=bc112874260946a2af2b7107825e6ce2&language=EN

* Success returns:

{

 "return\_code": "SUCCESS",

 "result\_code": "CLOSED",

 "currency": "AUD",

 "amount": 100,

 "refund\_time": "20161227105530",

 "success\_time": "20161227105602"

}

* Fail returns:

{

 "return\_code": "FAIL",

 "error\_code": "SIGN\_TIMEOUOT",

 "error\_msg": "Sign time out"

}

* Refund status list:

|  |  |
| --- | --- |
| Status | Description |
| Applied | Customer is applied the refund request. |
| MerchantConfirmed | Merchant accepted the refund request, and submitted to payment platform. |
| OrganizationConfirmed | Payment platform is confirmed refund. |
| OrganizationPayback | Payment platform is already refunded. |
| Closed | Refund flow is finished, flow closed. |
| MerchantRejected | Merchant rejected the refund request. |
| TimeoutClosed | The refund request is over the time limit for refunding of this order. |
| OrganizationFailed | Payment platform operation failed. |
| CustomerCancelled | Customer cancelled refund request. |