

In-Store API - QR Code Picture

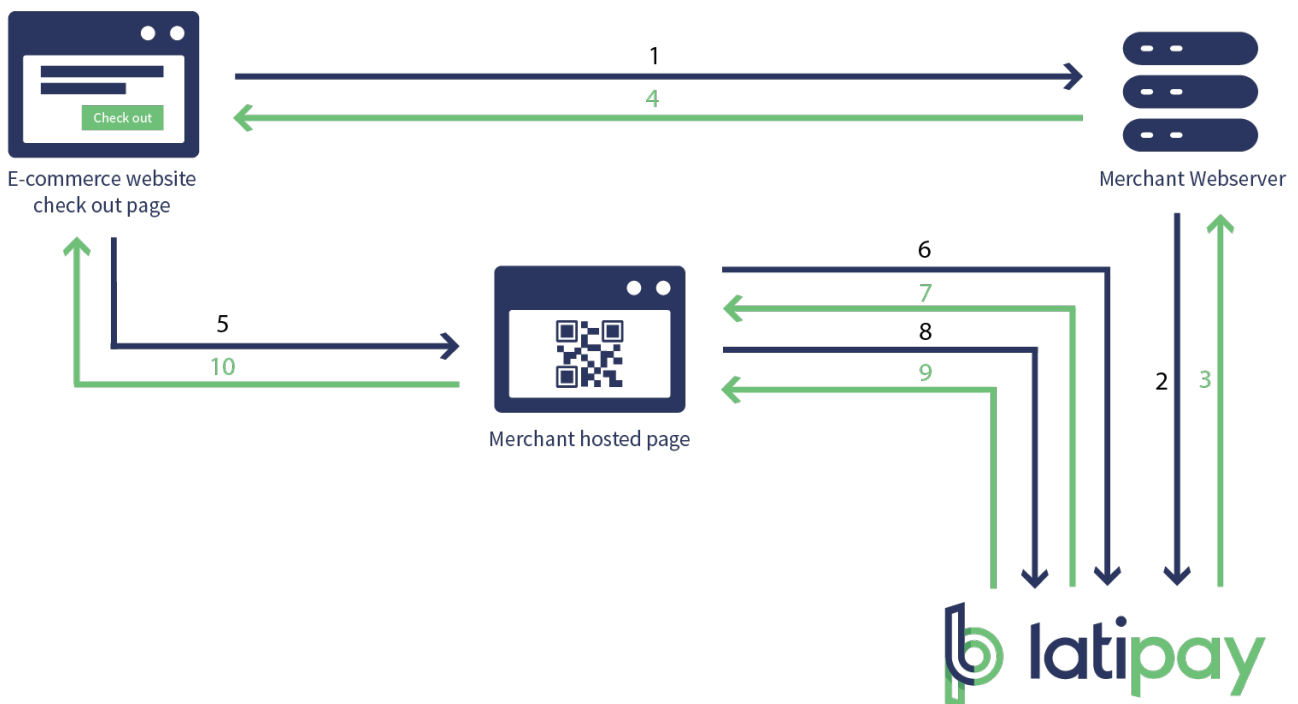
This api will return `QR Code picture` and order information in [#Payment Interface](#Payment-Interface), perfect for EFTPOS. This document is only suitable for the following `in-store` situation.

- Customer load Merchant's E-commerce website on `mobile browser` and pay through `Alipay app`.
- Customer load Merchant's E-commerce website inside of `Alipay app's browser` and pay through `Alipay app`.
- Customer scans `Alipay payment QR code` through `Alipay app`.
- Customer scans `Wechat payment QR code` through `Wechat app`.

How it works?

Here are two scenarios based on the payment methods.

For Alipay and Wechat Pay

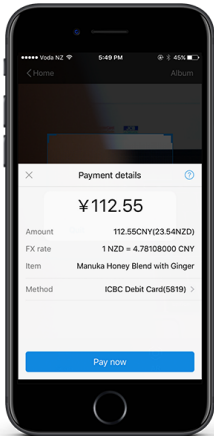


- the **Merchant** means Latipay partnership merchant.

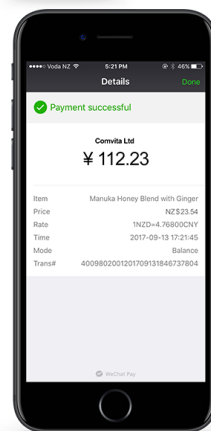
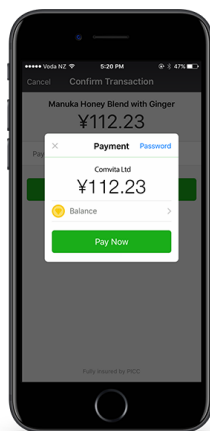
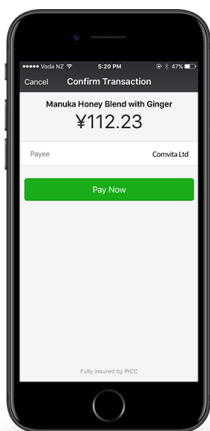
- **E-commerce website** can be any sub-merchant's website.
1. To process a transaction, customers select Latipay Wechat pay or Alipay as payment method, then the **E-commerce website** sends a request to **Merchant**.
 2. **Merchant** sends a request to **Latipay** [#Transaction-Interface](#) with the authentication details.
 3. **Latipay** responds to **Merchant** with a `host_url` and `nonce`
 4. **Merchant** responds to **E-commerce website** with a unique URI for an SSL secure payments page
 5. The shopping cart uses the returned URI to redirect the customer to the secure **Merchant** hosted payments page
 6. **Merchant** use the `host_url` and `nonce` to request a QR code. [#Payment-Interface](#)
 7. **Latipay** responds a QR code Url, then the QR code will display in **Merchant** hosted page. The customer will be prompted to scan the QR code and complete the payment.
 8. **Merchant** query the payment status from **Latipay** [#Query-Interface](#)
 9. **Latipay** updates the payment status
 10. The result is displayed and automatically redirected back to the **E-commerce website**.

Payment Scenarios

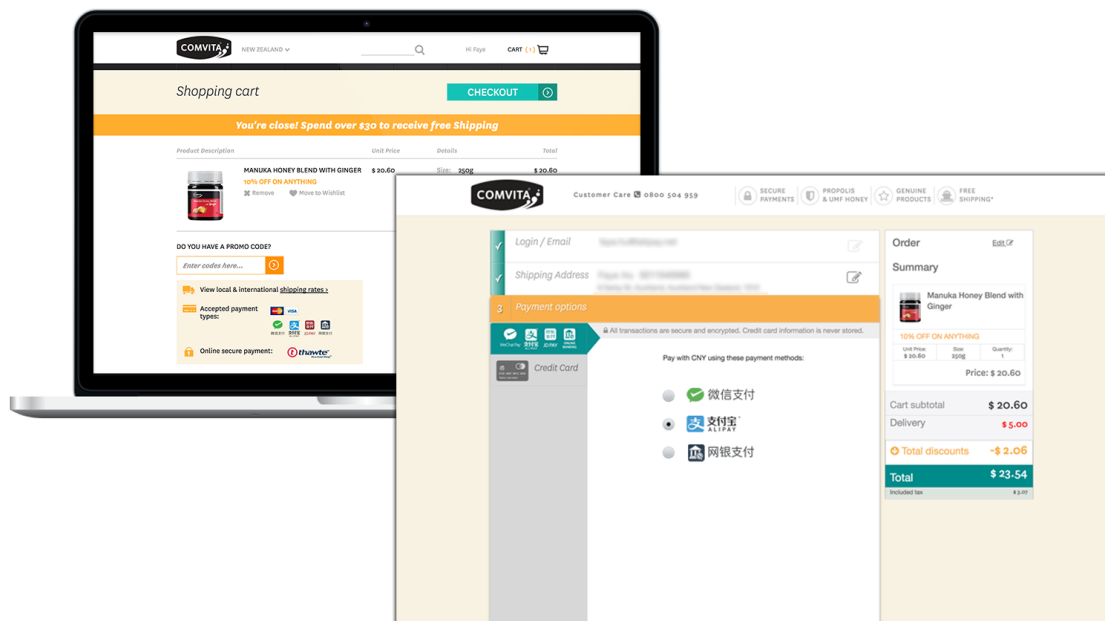
Alipay



WeChat Pay



Website check-out page



API List

Preparation

Before using bellowing api, please make sure you have `user_id` , `wallet_id` and `api_key` on hand. If you don't have them, we would like to help you. [Contact Us](#).

Transaction Interface

POST <https://api.latipay.net/v2/transaction>

Parameters:

Name	Type	Description	Nullable
signature	String	The SHA-256 HMAC API signature.	No
wallet_id	String	The ID of the wallet you want to use.	No
amount	String	A decimal amount.	No
user_id	String	The user account you want to use to process the transaction.	No
merchant_reference	String	A field for identifying your transaction.	No
currency	String	The currency code of the transaction.	No
return_url	String	The URL of the landing page where Latipay will return the customer after payment.	No
callback_url	String	The URL of the callback address the transaction notification is sent after payment.	No
ip	String	The IP address of the customer.	No
version	String	The latest version of the platform.	No
product_name	String	The name of the product or service being sold.	No
payment_method	String	Payment method options are alipay and wechat.	No
host_type	String	must be "1", payment interface can return QR code picture in base64	No

Example Parameters

```
{
  "signature": "14d5b06a2a5a2ec509a148277ed4cbeb3c43301b239f080a3467ff0aba4070e3",
  "wallet_id": "W000000001",
  "amount": "120.00",
  "user_id": "U000334333",
  "merchant_reference": "dsi39ej430sks03",
  "currency": "AUD",
  "return_url": "https://merchantsite.com/checkout",
  "callback_url": "https://merchantsite.com/confirm",
  "ip": "122.122.122.1",
  "version": "2.0",
  "product_name": "Pinot Noir, Otago",
  "payment_method": "alipay",
  "host_type" : "1"
}
```

SHA-256 HMAC Signature

```
message: user_id + wallet_id + amount + payment_method + return_url + callback_url
secret: api_key
```

Example Signature

```
message: U0000000001W0000000010.01alipayhttp://merchant.com/returnhttp://merchant.com/callback
secret: 111222333

signature: 2367bcd9e9a2f9a547c85d7545d1217702a574b8084bbb7ae33b45a03a89983
```

Response

```
{
  "signature": "14d5b06a2a5a2ec509a148277ed4cbeb3c43301b239f080a3467ff0aba4070e3",
  "host_url": "https://api.latipay.net/merchanthosted/gatewaydata",
  "nonce": "7d5a88119354301ad3fc250404493bd27abf4467283a061d1ed11860a46e1bf3"
}
```

Name	Type	Description
signature	String	The SHA-256 HMAC API signature.
host_url	String	The host URL to send your customer. The nonce should be appended to the end.
nonce	String	The transaction nonce must be appended to the pay URL.

Signature in Response

For security reasons, we highly recommend you verify the signature in the response.

```
message: nonce + host_url
secret: api_key
```

Payment Interface

```
POST {host_url}/{nonce}
```

Example

```
https://api.latipay.net/merchanthosted/gatewaydata/7d5a88119354301ad3fc250404493bd
27abf4467283a061d1ed11860a46e1bf3
```

Response

Name	Type	Description
code	Integer	The response code of payment, 0 or error code.
message	String	The response message of payment interface.
data	Object	The response of payment interface.

the `data` object

Name	Type	Description
order_id	String	A unique transaction identifier generated by Latipay.
nonce	String	A unique transaction nonce generated by Latipay.
payment_method	String	Payment method options are alipay and wechat.
amount	String	A decimal amount.
amount_cny	String	A decimal amount.
currency	String	The currency code of the transaction.
product_name	String	The name of the product or service being sold.
organisation_id	Integer	The merchant id which is used to process the transaction.
org_name	String	The merchant's name
user_id	String	The user account which is used to process the transaction.
user_name	String	The user account's name
wallet_id	String	The user wallet account which is used to process the transaction.
wallet_name	String	The wallet account's name
qr_code	String	The QR code picture
qr_codeurl	String	The QR code text version
currency_rate	String	The rate which is used to the transaction.
merchant_reference	String	A field for identifying your transaction.
signature	String	The SHA-256 HMAC API signature.

Example


```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "order_id": "20180112000003",
    "nonce": "fd270820180112103849a19380c7300c425898780b3d0b1fbc",
    "payment_method": "wechat",
    "amount": 0.01,
    "amount_cny": 0.05,
    "currency": "AUD",
    "product_name": "产品中文",
    "organisation_id": 695,
    "organisation_name": "126Emai",
    "user_id": "U000000266",
    "user_name": "04_oliver",
    "wallet_id": "W000000329",
    "wallet_name": "AUD_01",
    "qr_code": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAMgAAADICAYAAACT
WK6eAAAEzU1EQVR42u3dS27cQAwE0Ln/pZMTGDBgDVlFPQLZaWBjZadFpT+ff0qpH+vjFSgFiFKAKAWIUo
AoBYhS3UA+n0/Mv18/0C9/2/hsf3nev7yDrd8mvHtAAAEAAAAQAAAAABBJBoICPJwR9e9MSAPPlSE8+7
dS+NfQUIIIAAAggggAACCCCAAFIN50lm27ru6SZPSrEm7iUpaQAAAAQAAAAABBBBAAAEkAMiV6SJJCv
06akAAAAQAAAAABBBBAAAEFECKUqzGaSAT0za2kiIAAEAAAAQAAAAABBBBAvgjkbS9w4u+mY7UeBBBA
AAAEAAAAQAAAAABJABI0lQO1925ztaJgLG0EEBcBwggrgMEENcB8uIzCtPXyEW821bKVtUngAACCCCAAA
IIIIIAAAgggbUAmliRcnlay1ZTpiWTChwQQAABBBBAAAEAAAAASQaSOMRBhO/3RrM9CMWJhK6b94LIIAA
AggggAACCCCAAJINJckZCsJ3JWdFbeuSz2yARBAAEAAAAQAAAAABpA5IelpzZcdEzfv8cwAACCCCAAA
IIIIIAAAggggNQaedsB91tp0uUUq71fAAAEAAAAQAAAAABBBBAYoBsJTNJCdMEhulk5hquqJgXEEAAQQA
QAABBBBAAAFk8n/SJxKIK03euDviFq7UzTsAAQAAAAABBBBAAAEFEEDqUqytShrg9CZKnXoy8YEGBBBA
AAAEAAAAQAAAAAC5stlB+1l8SSnblatTsTMwLCCCAAAIIIIIAAAggggADSkjC17/jXCH0rGdxCAwgggAAC
CCCAAAIIIIIAAAkANIut9OzxAQxaT5M01eRMigUIIIIAAAggggAACCCCAAJLY0Fs79F1pyq0mb0/UAAEEEE
AAQAAAAABBBBAXgVkk4naevlXlmUkJXTfvD9AAAEAAAAQAAAAABBJAYIJeTisvJVuN7OXM+CCCAAAII
IIIAAAggggAACyCSQLUjPzTa9mcB24yc070TyBggggAACCCCAAAIIIIIAAEgMkaeAaz/tLeo705K0yxQIEEE
AAQAAAAABBBBAAJkEkv7yG481SGqE9F0jE5I8QAABBBBAAAEAAAAQSQGCBbaU1SqtOY/mjy554NEEAA
AQAAAAABBBBAAAEkBsJw1Iatw+ev7CCYvv6laVoOIIIAAAggggAACCCCAAJIDJD0Bkw/B7FxLUn6zooTfQ
oIIIIIAAAggggAACCCCAABIJ5PIajK2kDLiuMQIEEGMECCDGCBBAAAEFEEDqUqytShrg9CZKnXoy8YEGBBBA
AAAEAAAAQAAAAQACpBrI1wG/bxKBx3U16yvby2AACCCCAAAIIIIIAAAggggCQDSUqntqYxSMDyU6yn7g8QQA
ABBBBAAAEAAAAQOSOOZgt98qZNNnv7RSf+IAQIIIIIAAAggggAACCCCAAFIEZGKaReOUmfrjF9bOKAQE
EEAAQAAAAABBBBAAHkrkMYzBSfSs63pIlIsQAABBBBAAAEAAAAQSQACATDdiYuDQmOEnnTQICCCCAAA
IIIIIAAAggggAAynEpc3lBhKxncSveSurGn7gUQQAABBBBAAAEAAAAASQGiFJvL0CUAkQpQJQCRC1AlAJE
KUCUqq7/Vhpz81YibhcAAAAASUVORK5CYII=",
    "qr_code_url": "weixin://wxpay/bizpayurl?pr=Z0bbdsW",
    "currency_rate": "5.13710",
    "merchant_reference": "",
    "signature": "cf56a6bbelaa42e1c273c4d7fe6233f70a6b10ba39e88938b13469f3315e
4c41"
  }
}
```

SHA-256 HMAC Signature

Rearrange parameters in the `data` set alphabetically (except `signature` and other parameters with value of null or empty string) And connect rearranged parameters with `&`:

```
message: amount=1.00&amount_cny=5.00&currency=NZD&currency_rate=5.29930&merchant_r
eference=dsi39ej430sks03&nonce=7d5a88119354301ad3fc250404493bd27abf4467283a061d1ed
11860a46e1bf3&order_id=20170829-alipay-3990527237343&organisation_id=18&org_name=L
atipay&payment_method=alipay&product_name=test&qr_code=https://qr.alipay.com/bax03
286h4vlfpxldgwq4035&type=Online&user_id=U000000051&wallet_id=W000000037&wallet_nam
e=aud01
secret: api_key
```

Payment Result Asynchronous Notification

There is an asynchronous and back-end payment result notification sent by Latipay to merchant after the payment is done successfully. There is a re-try mechanism with the notification to ensure the notification could be delivered to the merchant.

```
POST merchant's callback_url
```

Parameters:

Name	Type	Description
signature	String	The SHA-256 HMAC API signature.
transaction_id	String	f66f-4f1b-9ef5-43cb917ff8a6 (string) - A unique transaction identifier generated by Latipay.
merchant_reference	String	A field for identifying your transaction.
currency	String	The currency code of the transaction.
amount	String	A decimal amount.
payment_method	String	The payment method used. Possible values are alipay and wechat.
status	String	The status of the transaction. Possible values are: pending, paid, or failed.
pay_time	String	Show the payment time of the transaction order.

Example input

```
{
  "signature": "14d5b06a2a5a2ec509a148277ed4cbeb3c43301b239f080a3467ff0aba4070e3",
  "transaction_id": "20170707-wechat-3473511594933",
  "merchant_reference": "dsi39ej430sks03",
  "currency": "AUD",
  "amount": "120.00",
  "payment_method": "wechat",
  "status": "paid",
  "pay_time": "2017-07-07 10:53:50"
}
```

SHA-256 HMAC Signature

Merchant backend need to validate the signature for protecting against malicious requests.

```
message: merchant_reference + payment_method + status + currency + amount
secret: api_key
```

Expected the text "sent" in Response's body

```
sent
```

Query Interface

All customers can send request to search their transaction order by merchant order id(that should be unique id for the merchant) as merchant_reference by HTTP GET request.

```
GET https://api.latipay.net/v2/transaction/{merchant_reference}
```

Parameters

Name	Type	Description
signature	String	The SHA-256 HMAC API signature.
user_id	String	The user account you want to use to process the transaction.

Example Parameters

```
{
  "signature": "14d5b06a2a5a2ec509a148277ed4cbeb3c43301b239f080a3467ff0aba4070e3",
  "user_id": "U000334333"
}
```

SHA-256 HMAC Signature

```
message: merchant_reference + user_id
secret: api_key
```

Response

Name	Type	Description
signature	String	The SHA-256 HMAC API signature.
transaction_id	String	A unique transaction identifier generated by Latipay.
merchant_reference	String	A field for identifying your transaction.
currency	String	The currency code of the transaction.
amount	String	A decimal amount.
payment_method	String	The payment method used. Possible values are alipay and wechat.
status	String	The status of the transaction. Possible values are: pending, paid, or failed.
pay_time	String	Show the payment time of the transaction order.

Example Response

```
{
  "signature": "14d5b06a2a5a2ec509a148277ed4cbeb3c43301b239f080a3467ff0aba4070e3",
  "transaction_id": "20170707-wechat-3473511594933",
  "merchant_reference": "dsi39ej430sks03",
  "currency": "AUD",
  "amount": "120.00",
  "payment_method": "wechat",
  "status": "paid",
  "pay_time": "2017-07-07 10:53:50"
}
```

Signature in Response

For security reasons, we highly recommend you verify the signature in the response.

```
message: merchant_reference + payment_method + status + currency + amount
secret: api_key
```



