



FXBit Crypto Payment Solution

API Integration Guide

v1.0.8



Copyright Information

This document and the software described by this document are copyright of © 2020 FXBit. All rights reserved.

Use of the software described herein may only be done in accordance with the License Agreement provided with the software. This document may not be reproduced in full or partial form except for the purpose of using the software described herein in accordance with the License Agreement provided with the software. Information in this document is subject to change without notice. Companies, names and data used in the examples herein are fictitious unless otherwise noted.

Confidentiality and Usage

The information contained in this document (and any attachments) is confidential information provided by FXBit. This document is intended only for use by merchants approved by FXBit. Any copying, distribution or dissemination of this document by any unauthorized parties is prohibited.

Revision History

Revision	Date	Revision Description
1.0.0	29 Mar, 2019	First issue
1.0.1	20 Apr, 2019	Updated data format of 'get exchange rate'
1.0.2	26 Sep, 2019	Added callback signature to 'Create a trade' function
1.0.3	29 Nov, 2019	Updated 'appendix B - currency' with description
1.0.4	22 Jan, 2020	Added rate to 'get wallet address' response
1.0.5	17 Feb, 2020	Updated 'appendix B - currency' and some descriptions
1.0.6	02 Mar, 2020	Added marked_currency, marked_price and billed_currency to callback response of 'Create a trade' function
1.0.7	10 Apr, 2020	Added expire time to 'get wallet address' response
1.0.8	23 Jul, 2020	Added H5 payment page

Table of Contents

1 Specification

2 Function of Merchant API

- 2.1 Get exchange rate
- 2.2 Get transaction details
- 2.3 Get wallet address
- 2.4 Create a trade
- 2.5 Polling the transaction status
- 2.6 H5 payment page

3 Appendix A – Response Code Specification

4 Appendix B - Currency

1. Specification

1.1. Communication Protocol

HTTP/HTTPS

1.2. Request Method

Support Post or Get method only.

1.3. Encoding Mode

All encoding formats use UTF-8, it includes the following terms:

- HTTP protocol communication
- BASE64 encoding of the message

1.4. Message Specification

The message specification is divided into request message and response message.

- The request message is initiated by the initiator.
- The response message is responded by the receiver.

2. Function of Merchant API

2.1. Get exchange rate

Function description:

Get the latest exchange rate

Request URL:

<https://merchant-api.fxbit.io/v1/api/getExchangeRate>

Request Mode:

GET

Request parameter format:

N/A

Response parameter format:

application/json

Request parameters:

N/A

Response parameters:

Parameter	Type	Remark
from_currency_code	Object	Currency before exchange
to_currency_code	Object	Currency after exchange
buyRate	String	Buying rate
sellRate	String	Selling rate
updated_timestamp	String	Exchange rate update time(timestamp)

Sample request:

```
curl -X GET "https://merchant-api.fxbit.io/v1/api/getExchangeRate" -H "accept: application/json" -H "x-api-key: HIX00hYqStkQ2G2NID4JoNR1YLB LPF0DMYqt7uka"
```

Successful response sample:

```
{
  "BTC": {
    "HKD": {
      "updated_timestamp": "2019-04-24T06:33:01.000Z",
      "buyRate": "48337.3",
      "sellRate": "41745.85"
    }
  }
}
```

```
},
"JPY": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "689032.3",
  "sellRate": "595073.35"
},
"CNY": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "41445.8",
  "sellRate": "35794.1"
},
"USD": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "5744.47925",
  "sellRate": "5464.26075"
},
"ETH": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "36.23674186",
  "sellRate": "31.29536797"
},
"USDT": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "6129.15807658",
  "sellRate": "5293.36379341"
},
"USDC": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "6170.76180864",
  "sellRate": "5329.29428928"
},
"TUSD": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "6164.19165032",
  "sellRate": "5323.62006164"
},
"PAX": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "6172.14678487",
  "sellRate": "5330.490405115"
},
"GUSD": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "6168.33959511",
```

```
"sellRate": "5327.202377595"
}
},
"ETH": {
  "HKD": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "1466.784",
    "sellRate": "1266.768"
  },
  "JPY": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "20914.355",
    "sellRate": "18062.3975"
  },
  "CNY": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "1257.674",
    "sellRate": "1086.173"
  },
  "USD": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "176.02245",
    "sellRate": "164.11755"
  },
  "USDT": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "186.05909916",
    "sellRate": "160.68740382"
  },
  "USDC": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "187.32236303",
    "sellRate": "161.778404435"
  },
  "TUSD": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "187.12288364",
    "sellRate": "161.60612678"
  },
  "PAX": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "187.32651014",
    "sellRate": "161.78198603"
  },
}
```



```
"GUSD": {
  "updated_timestamp": "2019-04-24T06:33:01.000Z",
  "buyRate": "187.21651586",
  "sellRate": "161.68699097"
},
"USDT": {
  "HKD": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "8.668",
    "sellRate": "7.486"
  },
  "JPY": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "123.651",
    "sellRate": "106.7895"
  },
  "CNY": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "7.436",
    "sellRate": "6.422"
  },
  "USD": {
    "updated_timestamp": "2019-04-24T06:33:01.000Z",
    "buyRate": "1.02515",
    "sellRate": "0.99485"
  }
}
```

Failed response sample:

```
{
  "error": {
    "code": "API_KEY_MISSING",
    "message": "No api_key was supplied. Get one at https://demo-api.fxbit.io"
  }
}
```

2.2. Get transaction details

Function description:

Get transaction details

Request URL:

<https://merchant-api.fxbit.io/v1/api/getTransactionDetail>

Request Mode:

GET

Request parameter format:

N/A

Response parameter format:

application/json

Request parameters:

Parameters	Required field?	Type	Remark
receipt_token	Y	String	Transaction code

Response parameters:

Parameters	Type	Remark
error	Number	0 - There is no error
receipt_token	String	Transaction code (Corresponding to the Payment ID)
shop_name	String	Name of shop
marked_currency	String	Settlement currency with merchant
marked_price	Number	Settlement amount with merchant
billed_currency	String	Payment currency from the customer
billed_price	Number	Payment amount from the customer
received_amount	Number	Amount received (in payment currency)
payment_status	Number	Transaction status: 0: new; 1: confirming; 2: confirmed; 3: failed; 4: insufficient; 5: expired; 6: Incorrect_amount
net	Number	Merchant net received amount
fee	Number	Handling fee

Sample request:

```
curl -X GET "https://merchant-api.fxbit.io/v1/api/getTransactionDetail?receipt_token=1552455439430" -H "accept: application/json" -H "x-api-key: HIX00hYqStkQ2G2NID4JoNR1YLB LPF0DMYqt7uka"
```

Successful response sample:

```
{  
  "error": 0,  
  "receipt_token": "1552455439430",  
  "shop_name": "Yuen_BitPay",  
  "marked_currency": "ETH",  
  "marked_price": 0.05,  
  "billed_currency": "ETH",  
  "billed_price": 0.05,  
  "received_amount": 0.05,  
  "payment_status": 2,  
  "net":0.04,  
  "fee":0.01  
}
```

Failed response sample:

```
{  
  "error": 1  
}
```

2.3. Get wallet address

Function description:

In response to the transaction, get the wallet address and payment amount.

Request URL:

<https://merchant-api.fxbit.io/v1/api/getWalletAddress>

Request Mode:

POST

Request parameter format:

application/json

Response parameter format:

application/json

Request parameters:

Parameters	Required field?	Type	Remark
receipt_token	Y	String	Transaction code
billed_currency	Y	String	Payment currency from the customer

Response parameters:

Parameters	Type	Remark
public_address	String	Payment address for cryptocurrency
billed_price	Number	Payment amount from the customer
index	Number	Unique address index ID
rate	Number	Transaction exchange rate
expire	Number	Rate expire time (in seconds)

Sample request:

```
curl -X POST "https://merchant-api.fxbit.io/v1/api/getWalletAddress" -H "accept: application/json" -H "x-api-key: HIX00hYqStkQ2G2NID4JoNR1YLBLPF0DMYqt7uka" -H "Content-Type: application/json" -d "{\"receipt_token\": \"1552483134660\", \"billed_currency\": \"BTC\"}"
```

Successful response sample:

```
{  
  
  "public_address": "1CjoVrwpNBLMCdvYAAAYEU4keahJLsiY1C7",  
  "billed_price": "0.02671",  
  "index": 99,  
}
```

```
"rate": "6168.33959511",  
"expire": 1800
```

```
}
```

Failed response sample:

```
{  
  "error": 2,  
  "error_message": "Shop or receipt does not exist."  
}
```

2.4. Create a trade

Function description:

Create a new trading (No currency has been chosen)

Request URL:

<https://merchant-api.fxbit.io/v1/api/getReceiptToken>

Request Mode:

POST

Request parameter format:

application/json

Response parameter format:

application/json

Request parameters:

Parameters	Required Field?	Type	Remark
marked_currency	Y	String	Settlement currency with merchant
marked_price	Y	Number	Settlement amount with merchant
remark	Y	String	The remark
callback_url	Y	String	The callback address for notification

Response parameters:

Parameters	Type	Remark
error	Number	0 - There is no error
shop_identifier	String	The identifier of shop
receipt_token	String	Transaction code (Corresponding to the Payment ID)

Sample request:

```
curl -X POST "https://merchant-api.fxbit.io/v1/api/getReceiptToken" -H "accept: application/json" -H "x-api-key: HIX00hYqStkQ2G2NID4JoNR1YLBLPF0DMYqt7uka" -H "Content-Type: application/json" -d "{\"marked_currency\": \"USD\", \"marked_price\": \"100\", \"remark\": \"remark\", \"callback_url\": \"https://example.com/callback\"}"
```

Successful response sample:

```
{
  "error": 0,
  "receipt_token": 1552483134660,
```

```
"shop_identifier": "39cf4fb7d461d2fd73a4baf6585b7c21"
}
```

Failed response sample:

```
{
  "err": "Invalid Currency"
}
```

Callback response parameters:

Parameters	Type	Remark
receipt_token	String	Transaction code (Corresponding to the Payment ID)
marked_currency	String	Settlement currency with merchant
marked_price	Number	Settlement amount with merchant
billed_currency	String	Payment currency from the customer
billed_price	Number	Payment amount from the customer
received_amount	Number	Amount received (in payment currency)
status	Number	Transaction status: 0: new; 1: confirming; 2: confirmed; 3: failed; 4: insufficient; 5: expired; 6: Incorrect_amount
signature	String	

Callback response sample:

```
{
  "receipt_token": "1552455439430",
  "marked_currency": "ETH",
  "marked_price": 0.05,
  "billed_currency": "ETH",
  "billed_price": 0.05,
  "received_amount": 0.05,
  "status": 2,
  "signature ":
  "765a554fc41cb3058f1143e4f416e0bc5c8bbe6fa4549e6b98d21d8fa5ac596a"
}
```

Tips: Please receive callback data in json format

Signature:

Signature mechanism

In order to ensure the authenticity and completeness of the data transmission process, the callback is returned with a signature, and the merchants verify after receiving the callback.

The signature is generated by combining the values of the ascending order according to the key parameter letters, and add the merchant API Key at the end to get the raw string. After that, use SHA-256 encryption method to get the signature result.

String to be signed

1. Except for the “signature” string, all parameters are handled in ascending order of key parameter letters.
2. In the string to be signed, the raw values are used in the field values.

Example

Using the above callback response as example, combine the values of the ascending order according to the json data, and the merchant API Key is added at the end to get the following result:

```
billed_price = 0.05 & receipt_token = 1552455439430 & received_amount = 0.05 & status = 2 & your_api_key
```

Then encrypt the above string to be signed by SHA-256 algorithm and obtain the data from the signature.

And get the following NodeJS signing code:

```
signature = crypto.createHash('sha256').update('billed_price = 0.05 & receipt_token = 1552455439430 & received_amount = 0.05 & status = 2 & your_api_key').digest('hex');
```

Verify signature when receiving callback response

When the callback response is received, the merchant needs to process the parameters with the above signature algorithm and obtain a 64-bit string. The signature from the callback response should be identical as this 64-bit string to proof the signature is validated.

2.5. Polling the transaction status

Function description:

Polling the transaction status

Request URL:

<https://merchant-api.fxbit.io/v1/api/pollPaymentStatus>

Request Mode:

POST

Request parameter format:

application/json

Response parameter format:

application/json

Request parameters:

Parameters	Required field?	Type	Remark
address	Y	String	Transaction address

Response parameters:

Parameters	Type	Remark
status	Number	Transaction status: 0: There are no entries 1: Monitoring money into this address

Sample request:

```
curl -X POST "https://merchant-api.fxbit.io/v1/api/pollPaymentStatus" -H "accept: application/json" -H "x-api-key: HIX00hYqStkQ2G2NID4JoNR1YLBLPF0DMYqt7uka" -H "Content-Type: application/json" -d "{\"address\": \"1CjoVrwpNBLMCdvYAAAYEU4keahJLsiY1C7\"}"
```

Successful response sample:

```
{  
  "status": 1  
}
```

Failed response sample:

```
{  
  "status": 0  
}
```

2.6. H5 payment page

Function description:

Create wallet address QR code page

Request URL:

<https://merchant-api.fxbit.io/v2>

Request Mode:

GET

Request parameter format:

Query String Parameters

Response parameter format:

Redirect to QR code page

Request parameters:

Parameters	Required field?	Type	Remark
receipt_token	Y	String	Transaction code
wallet_addr	Y	String	Wallet address

Request description:

Call “2.4 getReceiptToken” to get the transaction code (receipt_token), and get the public_address from “2.3 getWalletAddress”. Use the receipt_token and public_address to redirect to H5 QR code page.

Sample request:

https://merchant-api.fxbit.io/v2?receipt_token=1594693369670&wallet_addr=0x70fb3ff36f1c357f08a39b6590b220eb9361a2a8

Note:

The QR code page can only access for the transaction created within 7 days. For expired or completed transactions, the wallet address will not be shown in the page again.

3. Appendix A - Response Code Specification

Response Code	Remark
200	Success
301	Parse message error
302	Invalid call credentials
303	Incorrect parameter
500	Internal system error

4. Appendix B – Currency

Currency	Remark
CNY	China Yuan
HKD	Hong Kong Dollar
JPY	Japanese Yen
USD	American Dollar
BTC	Bitcoin
ETH	Ethereum
USDT	Tether USD (OMNI)
USDT_ERC20	Tether USD (ERC20)
USDC	USDC
TUSD	TUSD
PAX	PAX