

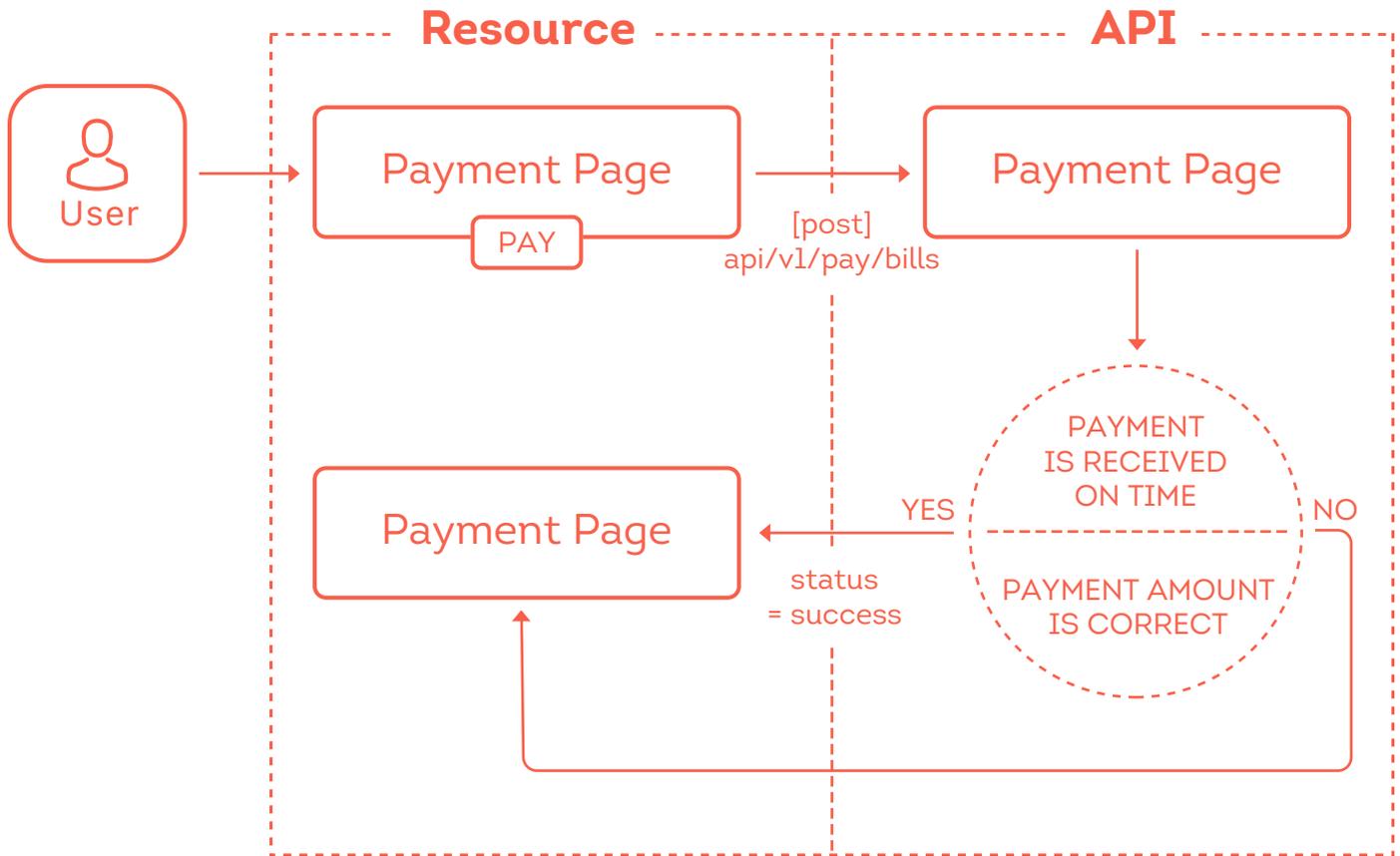
B2BinPAY Payment System Integration Guide

(Cryptocurrency Payments)

Table of contents

1. Scheme: Interaction with the payment system
2. Information about payment system
 - 2.1. [getaway] URL
 - 2.2. [currency node] URL
3. Obtaining a temporary token
4. Creating a payment order
5. Data returned by callback
6. Request for quotation
 - 6.1. Request for quotes: deposit
 - 6.2. Request for quotes: withdraw
7. Automatic withdrawal
8. Payment status list
9. Examples of use
 - 9.1. Examples of use
 - 9.2. Creating a payment order

1. Scheme: Interaction with the payment system



2. Information about payment system

2.1. [getaway] URL

Production	Sandbox
https://gw.b2binpay.com	https://gw-test.b2binpay.com/

2.2. [currency node] URL

Валюта	Production	Sandbox	Payment Order Validation
BTC	https://btc.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
LTC	https://ltc.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
ETH	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
XMR	https://xmr.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
BCH	https://bch.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
DASH	https://dash.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
NEO	https://neo.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
ADA	https://ada.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
DOGE	https://doge.b2binpay.com	https://cr-test.b2binpay.com	Recipient's unique address
B2BX	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
USDT	https://omni.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
USDT (Ethereum)	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
EURT (Ethereum)	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
R	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
OMG	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
IOST	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
VIU	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
XRP	https://xrp.b2binpay.com	https://cr-test.b2binpay.com	Payment's unique message
NEM	https://nem.b2binpay.com	https://cr-test.b2binpay.com	Payment's unique message
PAX	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
TUSD	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
GUSD	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
USDC	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address
BNB	https://eth.b2binpay.com	https://cr-test.b2binpay.com	Sender's address

3. Obtaining a temporary token

OAuth 2.0 is used to access to Application Programming Interface (API) services.

To authorize and obtain the temporary token on the URL `currency_domain/api/login`, GET HTTP-request is sent with the following header:

Authorization: Basic base64_encode(key:secret)

In the case of successful authorization, the following HTTP-response (JSON) comes back:

```
Status: 200
Body: {
  "token_type": "Bearer",
  "access_token": TOKEN_WILL_BE_HERE,
  "lifetime": TOKEN_LIFETIME,
}
```

In the case of error detection, the system returns the following information (JSON):

```
Status: 404|401
Body: {
  "code": ERROR_CODE,
  "error": ERROR_NAME
}
```

This temporary token received is used when accessing API services.

4. Creating a payment order

The POST method is used to create a payment order.

URL address: [currency node]/api/v1/pay/bills.

Whereas authorization is required to make the POST request, it is essential to provide temporary token in the HTTP-request header.

To create a payment order, the HTTP-request should be sent with the following header:

Parameter	Description
authorization	Bearer access_token Where access_token is - temporary token received at the authorization stage

with the following parameters:

Required parameters	Description
amount	Transaction amount (real number transmitted into the system; number should be positive real one)
wallet	Identification of currency (data transmitted by the system)
address	Wallet address of the sender needed for payment systems in order to validate payments (see Section 2: Information about payment system) Note: This parameter is not used in relation to other systems.

Optional parameters	Description
tracking_id	Identifier of the payment order in your system (could be a number or a string)
lifetime	Lifetime of created payment order in seconds; If lifetime is set to 0 (zero), created payment order will never expire.
pow	Power of the parameter amount (amount*10^pow); In order to maintain the precision of small sums during transmission, capability to raise amount to the power is provided.
callback_url	Address for notification on change of payment status. If parameter is absent, primary address specified in connection settings is used.

If the payment order is successfully created, the following HTTP-response comes back:

Status: 200

Body : {

 "data": {

 "id": ID,

 "url": URL_TO_BILL_PAGE,

 "address": BLOCKCHAIN_ADDRESS,

 "created": TIME,

 "expired": TIME|NULL,

 "status": BILL_STATUS,

 "tracking_id": TRACKING_ID,

 "amount": AMOUNT_MULTIPLIED_BY_TEN_IN_POW,

 "actual_amount":

ALREADY_PAID_AMOUNT_MULTIPLIED_BY_TEN_IN_POW,

 "pow": POW,

 "message": MESSAGE,

 },

}

5. Data returned by callback

Parameter	Description
id	Identifier of the payment order in the payment system
url	Link to the payment order page
address	Recipient's unique address in the blockchain
created	Time of payment order creation
expired	Expiry time of payment order
status	Payment order status (see table: "Payment status list" below)
tracking_id	Identifier of the payment order specified at the account opening stage
amount	Payment amount specified at the account opening stage (expected payment amount)
actual_amount	Amount of funds actually received on the account
pow	Power of the parameters: amount and actual_amount (amount / actual_amount *10^pow)
transactions	List of payment transactions in blockchain; Interest should be paid only to transaction field, which allows to request transaction in blockchain.
message	Additional message which is required to send inside a transaction (if blockchain requires)

When the data is successfully processed, the following response should be returned:

Status: 200

Body : OK

Note: The answer confirms transaction in the system and allows to withdraw funds subsequently (the system will repeat the request until it receives this answer or reaches the limits of requests). If necessary, it is possible to call the request manually via Administration Panel.

6. Request for quotation

6.1. Request for quotes: deposit

The GET-method is used to request quotes.

URL: <https://gw.b2binpay.com/api/v1/rates/deposit/>
 (sandbox: <https://gw-test.b2binpay.com/api/v1/rates/deposit/>).

You can also add the base currency alpha code at the end of the link
 (For example, <https://gw.b2binpay.com/api/v1/rates/deposit/eur/>).
 The default base currency is the dollar (USD).

Whereas authorization is required to make the request,
 it is essential to provide temporary token in the HTTP-request header.

To create a payment order, the HTTP-request should be sent with the following header:

Parameter	Description
authorization	Bearer access_token Where access_token is - temporary token received at the authorization stage

If the request for quotes is successful, the following HTTP-response comes back:

```
Status: 200
Body : {
  "data": [
    {
      "from": {
        "alpha": BASE_CURRENCY_ALPHA,
        "iso": BASE_CURRENCY_ISO
      },
      "to": {
        "alpha": TARGET_CURRENCY_ALPHA,
        "iso": TARGET_CURRENCY_ISO
      },
      "rate": RATE_MULTIPLIED_BY_TEN_IN_POW,
      "pow": POW,
      "expire": RATE_EXPIRE_TIMESTAMP
    },
    ...
  ],
}
```

6.2. Request for quotes: withdrawal

The GET-method is used to request quotes.

URL: <https://gw.b2binpay.com/api/v1/rates/withdraw/>
 (sandbox: <https://gw-test.b2binpay.com/api/v1/rates/withdraw/>).

You can also add the base currency alpha code at the end of the link
 (For example, <https://gw.b2binpay.com/api/v1/rates/withdraw/eur/>).
 The default base currency is the dollar (USD).

Whereas authorization is required to make the request,
 it is essential to provide temporary token in the HTTP-request header.

To create a payment order, the HTTP-request should be sent with the following header:

Parameter	Description
authorization	Bearer access_token Where access_token is - temporary token received at the authorization stage

If the request for quotes is successful, the following HTTP-response comes back:

```
Status: 200
Body : {
  "data": [
    {
      "from": {
        "alpha": BASE_CURRENCY_ALPHA,
        "iso": BASE_CURRENCY_ISO
      },
      "to": {
        "alpha": TARGET_CURRENCY_ALPHA,
        "iso": TARGET_CURRENCY_ISO
      },
      "rate": RATE_MULTIPLIED_BY_TEN_IN_POW,
      "pow": POW,
      "expire": RATE_EXPIRE_TIMESTAMP
    },
    ...
  ],
}
```

7. Automatic withdrawal

The POST method is used to create a withdrawal.
 URL address: [gateway]/api/v1/virtualwallets/withdraws.

Whereas authorization is required to make the POST request, it is essential to provide temporary token in the HTTP-request header.

To create a payment order, the HTTP-request should be sent with the following header:

Parameter	Description
authorization	Bearer access_token Where access_token is - temporary token received at the authorization stage

with the following parameters:

Required parameters	Description
amount	Transaction amount (real number transmitted into the system; number should be positive real one)
virtual_wallet_id	Identification of virtual wallet (data transmitted by the system)
address	Wallet address of the recipient
currency	Alpha-code of withdrawal currency
Optional parameters	Description
tracking_id	Identifier of the withdrawal in your system (could be a number or a string)
pow	Power of the parameter amount (amount*10^pow); In order to maintain the precision of small sums during transmission, capability to raise amount to the power is provided.
callback_url	Address for notification on change of withdrawal status.

If the withdrawal order is successfully created, the following HTTP-response comes back (in the same format data comes via POST-request on the callback_url specified in the parameters):

```

Status: 200
Body : {
  "data": {
    "id": ID,
    "virtual_wallet_id": VIRTUAL_WALLET_ID,
    "address": BLOCKCHAIN_ADDRESS,
    "transaction": BLOCKCHAIN_TRANSACTION,
    "created": TIME,
    "callback_url": CALLBACK_URL|NULL,
    "status": BILL_STATUS,
    "tracking_id": TRACKING_ID,
    "amount": AMOUNT_MULTIPLIED_BY_TEN_IN_POW,
    "pow": POW,
    "currency": {
      "alpha": CURRENCY_ALPHA,
      "iso": CURRENCY_ISO
    },
  },
},
}
    
```

Withdrawal status list:

Status	Description
-2	Withdrawal ended up with an error. The funds are returned to the sender's e-Wallet
0	Withdrawal is processed
1	Withdrawal has been sent to the blockchain
2	Withdrawal is confirmed in the blockchain

8. Payment status list

Status	Description
-2	Error on payment execution
-1	Account validity period has expired
1	Waiting for payment
2	Payment is paid
3	Payment is frozen
4	Payment is closed (funds are extracted)

```
{
  "id": ID,
  "url": URL,
  "address": ADDRESS_IN_BLOCKCHAIN,
  "created": TIME,
  "expired": TIME|NULL,
  "status": BILL_STATUS,
  "tracking_id": BILL_IDENTIFIER,
  "amount": PAYMENT_AMOUNT,
  "actual_amount": AMOUNT_OF_FUNDS_RECEIVED,
  "pow": POW,
  "transactions":
    {
      "id": TRANSACTION_ID,
      "bill_id": BILL_ID,
      "created": TIME_OF_PAYMENT_CREATION,
      "amount": AMOUNT_MULTIPLIED_BY_TEN_IN_POW,
      "pow": POW,
      "status": TRANSACTION_STATUS,
      "transaction": TRANSACTION_IN_BLOCKCHAIN,
      "type": TYPE (SEND|RECEIVE),
      "currency": {
        "iso": CURRENCY_ISO,
        "alpha": CURRENCY_ALPHA
      }
    }
  ]
}
```

Transaction status list:

Статус	Описание
-2	Error on transaction execution
0	Transaction is pending (located in blockchain, but it does not have sufficient number of confirmations)
1	Transaction is approved in blockchain, merchant's approval is expected (callback)
2	Transaction is approved

