

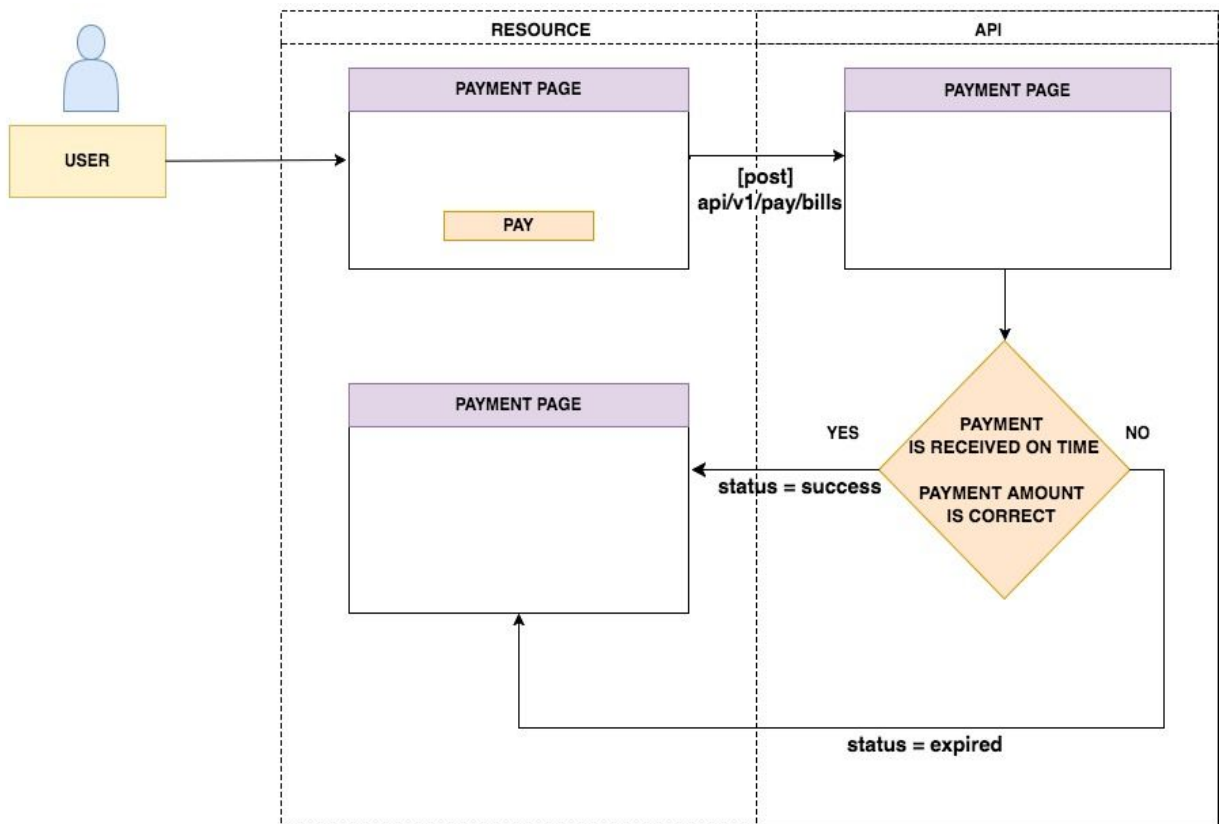
**B2BinPAY Payment System Integration
GUIDE
(Cryptocurrency Payments)**



Table of contents

Scheme: Interaction with the payment system	2
2. Information about payment system	3
2.1. [getaway] URL	3
2.2. [currency node] URL	3
3. Obtaining a temporary token	5
4. Creating a payment order	6
5. Data returned by callback	8
6. Request for quotation	9
6.1. Request for quotes: deposit	9
6.2. Request for quotes: withdrawal	10
7. Automatic withdrawal	11
8. Payment status list	14
Error on payment execution	14
9. Examples of use	15
9.1. Obtaining a temporary token	16
9.2. Creating a payment order	17

1. Scheme: Interaction with the payment system



2. Information about payment system

2.1. [getaway] URL

Production	Sandbox
https://gw.b2binpay.com	https://paysystemtest.b2broker.info/

2.2. [currency node] URL

Currency	Production	Sandbox	Payment Order Validation
BTC	https://btc.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address
LTC	https://ltc.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address
ETH	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address
XMR	https://xmr.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address
BCH	https://bch.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address
DASH	https://dash.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address
NEO	https://neo.b2binpay.com	https://paysystemtest.b2broker.info/	Recipient's unique address

B2BX	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
USDT	https://omni.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
USDT (Ethereum)	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
EURT (Ethereum)	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
R	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
OMG	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
IOST	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
VIU	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
EOS	https://eth.b2binpay.com	https://paysystemtest.b2broker.info/	Sender's address
XRP	https://xrp.b2binpay.com	https://paysystemtest.b2broker.info/	Payment's unique message
NEM	https://nem.b2binpay.com	https://paysystemtest.b2broker.info/	Payment's unique message

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.

Payment and transaction statuses returned are located in the chapter “Payment status list”.

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
<i>authorization</i>	Bearer access_token Where access_token is - temporary token received at the authorization stage

with the following parameters:

Required parameters	Description
<i>amount</i>	Transaction amount (real number transmitted into the system; number should be positive real one)
<i>wallet</i>	Identification of currency (data transmitted by the system)
<i>address</i>	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
<i>tracking_id</i>	Identifier of the payment order in your system (could be a number or a string)
<i>lifetime</i>	Lifetime of created payment order in seconds; If <i>lifetime</i> is set to 0 (zero), created payment order will never expire.
<i>pow</i>	Power of the parameter <i>amount</i> ($amount * 10^{pow}$); In order to maintain the precision of small sums during transmission, capability to raise <i>amount</i> to the power is provided.
<i>callback_url</i>	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

Data is sent via form-data in POST requests

Parameter	Description
<i>id</i>	Identifier of the payment order in the payment system
<i>url</i>	Link to the payment order page
<i>address</i>	Recipient's unique address in the blockchain
<i>created</i>	Time of payment order creation
<i>expired</i>	Expiry time of payment order
<i>status</i>	Payment order status (see table: "Payment status list" below)
<i>tracking_id</i>	Identifier of the payment order specified at the account opening stage
<i>amount</i>	Payment amount specified at the account opening stage (expected payment amount)
<i>actual_amount</i>	Amount of funds actually received on the account
<i>pow</i>	Power of the parameters: <i>amount</i> and <i>actual_amount</i> ($\text{amount} / \text{actual_amount} * 10^{\text{pow}}$)
<i>transactions</i>	List of payment transactions in blockchain; Interest should be paid only to <i>transaction</i> 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://paysystemtest.b2broker.info/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://paysystemtest.b2broker.info/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
<i>authorization</i>	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
<i>authorization</i>	Bearer access_token Where access_token is - temporary token received at the authorization stage

with the following parameters:

Required parameters	Description
<i>amount</i>	Transaction amount (real number transmitted into the system; number should be positive real one)
<i>virtual_wallet_id</i>	Identification of virtual wallet (data transmitted by the system)
<i>address</i>	Wallet address of the recipient
<i>currency</i>	Alpha-code of withdrawal currency
Optional parameters	Description
<i>tracking_id</i>	Identifier of the withdrawal in your system (could be a number or a string)
<i>pow</i>	Power of the parameter <i>amount</i> ($amount * 10^{pow}$);

	In order to maintain the precision of small sums during transmission, capability to raise amount to the power is provided.
with_fee	A way to work with blockchain commission. By default, the withdrawal amount does not contain blockchain commission. To include commission it is necessary to specify 1 in a request.
callback_url	Address for notification on change of withdrawal status.

If the withdrawal order is successfully created, the following HTTP-response comes back (the same response will be sent to the `callback_url` specified in the parameters. Response is encoded to form-data of the POST request):

```
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

Status	Description
-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

9. Examples of use

Test data used in the examples:

Parameter	Description
<i>domain</i>	https://paysystemtest.b2broker.info/
<i>key</i>	0fba80bf40fd931
<i>secret</i>	f203ef7d0f76a98
<i>base64_encode</i> <i>(key:secret)</i>	MWMzZWQ4MzQzODoyZmNhODk4MjVkJzlyZmQ=
<i>token</i>	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJDcnlwdG9QYXkiLCJzdzWliOiIwZmJhODBiZjQwZmQ5MzEiLCJpYXQiOiJlMTE5MDY3MjI1IiwiaWF0IjoiMTUyMDU0NjcyMn0.-AaTOAnhne-u8ioWMJrTozph_25mQhSTQGS2cx3tx6w
<i>wallet</i>	2

9.1. Obtaining a temporary token

cURL:

```
curl --request GET \  
  --url https://paysystemtest.b2broker.info/api/login \  
  --header 'authorization: Basic \  
MWMzZWQ4MzQzODoyZmNhODk4MjVkJzlyZmQ='
```

jQuery:

```
var settings = {
```

```

    "async": true,
    "crossDomain": true,
    "url": "https://paysystemtest.b2broker.info/api/login",
    "method": "GET",
    "headers": {
      "authorization": "Basic MWMzZWQ4MzQzODoyZmNhODk4MjVkJVZmQ="
    }
  }
}

$.ajax(settings).done(function (response) {
  console.log(response);
});

```

PHP:

```

<?php

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => "https://paysystemtest.b2broker.info/api/login",
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "GET",
  CURLOPT_HTTPHEADER => array(
    "Authorization: Basic MWMzZWQ4MzQzODoyZmNhODk4MjVkJVZmQ="
  ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
  echo "cURL Error #:" . $err;
} else {
  echo $response;
}

```

```
}
```

9.2. Creating a payment order

cURL:

```
curl --request POST \  
  --url https://paysystemtest.b2broker.info/api/v1/pay/bills \  
  --header 'authorization: Bearer  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJDcnlwdG9QYXkiLCJz  
dWIiOiIwZmJhODBiZjQwZmQ5MzEiLCJpYXQiOiJlMTE5MDY3MjIsImV4cCI6MTUyMDU  
0NjcyMn0.-AaTOAnhne-u8ioWMJrTozph_25mQhSTQGS2cx3tx6w' \  
  --header 'content-type: application/x-www-form-urlencoded' \  
  --data 'wallet=2&amount=100000&lifetime=0&pow=8'
```

jQuery:

```
var settings = {  
  "async": true,  
  "crossDomain": true,  
  "url": "https://paysystemtest.b2broker.info/api/v1/pay/bills",  
  "method": "POST",  
  "headers": {  
    "authorization": "Bearer  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJDcnlwdG9QYXkiLCJz  
dWIiOiIwZmJhODBiZjQwZmQ5MzEiLCJpYXQiOiJlMTE5MDY3MjIsImV4cCI6MTUyMDU  
0NjcyMn0.-AaTOAnhne-u8ioWMJrTozph_25mQhSTQGS2cx3tx6w",  
    "content-type": "application/x-www-form-urlencoded"  
  },  
  "data": {  
    "wallet": "2",  
    "amount": "100000",  
    "lifetime": "0",  
    "pow": "8"  
  }  
}  
  
$.ajax(settings).done(function (response) {
```

```
    console.log(response);
});
```

PHP:

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL =>
    "https://paysystemtest.b2broker.info/api/v1/pay/bills",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_POSTFIELDS => "wallet=2&amount=100000&lifetime=0&pow=8",
    CURLOPT_HTTPHEADER => array(
        "authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJKDcnlwdG9QYXkiLCJz
dWUiOiIwZmJhODBiZjZjQwZmQ5MzEiLCJpYXQiOiJlMTE5MDY3MjIsImV4cCI6MTUy
MDU0NjcyMn0.-AaTOAnhne-u8ioWMJrTozph_25mQhSTQGS2cx3tx6w",
        "content-type: application/x-www-form-urlencoded"
    )
));

$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);

if ($error) {
    echo "cURL Error #:" . $error;
} else {
    echo $response;
}
```