

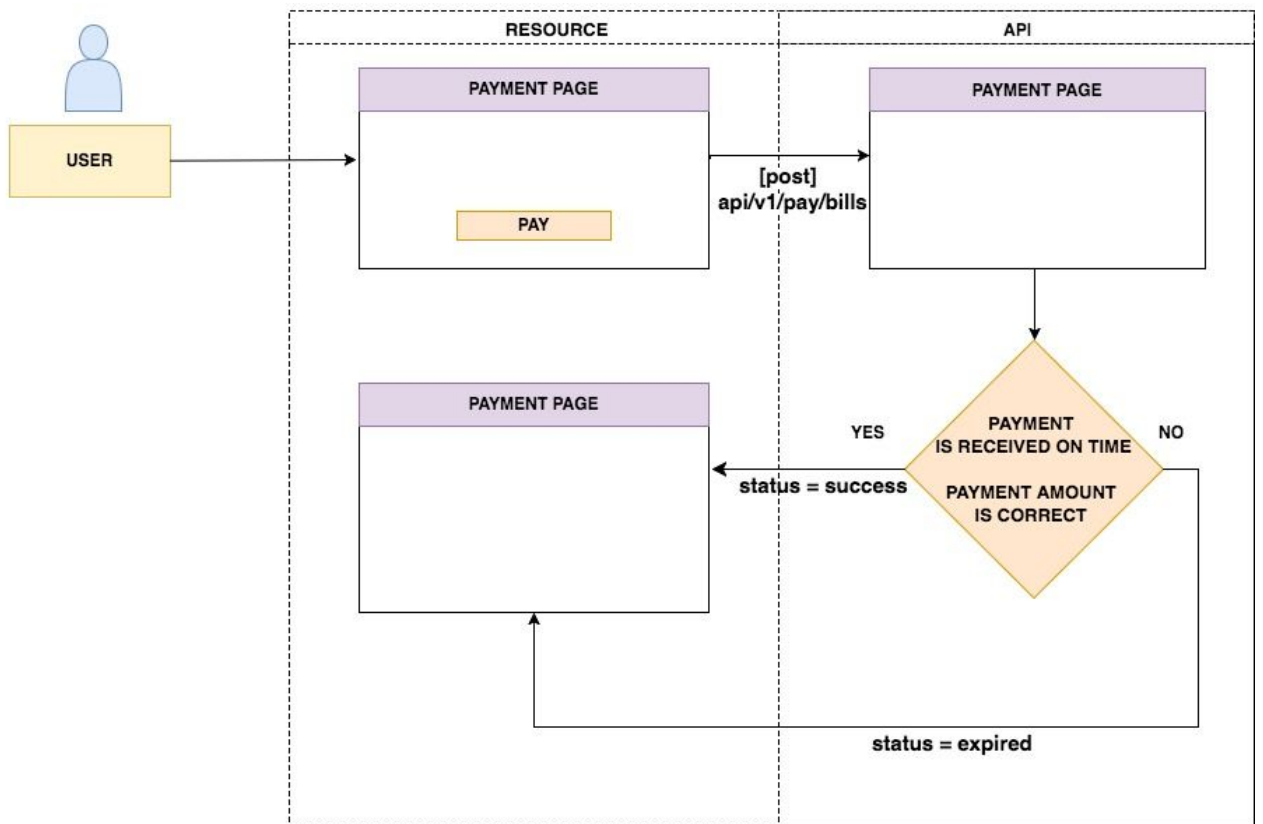
**B2BinPAY Payment System Integration
GUIDE
(Cryptocurrency Payments)**



Contents

1. Scheme: Interaction with the payment system.....	2
2. Information about payment system.....	3
3. Obtaining a temporary token.....	4
4. Creating a payment order.....	5
5. Data returned by callback.....	7
6. Payment status list.....	8
7. Examples of use.....	10
7.1. Obtaining a temporary token.....	10
7.2. Creating a payment order.....	11

1. Scheme: Interaction with the payment system



2. Information about payment system

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
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

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_domain/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
<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
  },
}
```

5. Data returned by callback

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: amount and actual_amount ($\text{amount} / \text{actual_amount} * 10^{\text{pow}}$)
<i>transactions</i>	List of payment transactions in blockchain; Interest should be paid only to transaction field, which allows to request transaction in blockchain.

6. Payment status list

Status	Description
-2	Error on payment execution
-1	Account validity period has expired
1	Payment is pending (located in blockchain, but it does not have sufficient number of confirmations)
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  
    }  
}  
]  
}
```

7. 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 (key:secret)</i>	MWMzZWQ4MzQzODoyZmNhODk4MjVkJzlyZmQ=
<i>token</i>	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJDcnlwdG9QYXkiLCJzdzWliOilwZmJhODBiZjZjQwZmQ5MzEiLCJpYXQiOiJlMTE5MDY3MjlsImV4cCI6MTUyMDU0NjcyMn0.-AaTOAnhne-u8ioWMJrTozph_25mQhSTQGS2cx3tx6w
<i>wallet</i>	2

7.1. Obtaining a temporary token

cURL:

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

jQuery:

```
var settings = {  
  "async": true,
```


jQuery:

```
var settings = {
  "async": true,
  "crossDomain": true,
  "url": "https://paysystemtest.b2broker.info/api/v1/pay/bills",
  "method": "POST",
  "headers": {
    "authorization": "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJDcnlwdG9QYXkiL
CjZdWIiOiIwZmJhODBiZjQwZmQ5MzEiLCJpYXQiOiJlMTE5MDY3MjIsImV4cCI6MTU
yMDU0NjcyMn0.-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
eyJ0eXAI0iJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJCMkJDcnlwdG9QYXkiL
CJzdWIiOiIwZmJhODBiZjQwZmQ5MzEiLCJpYXQiOiE1MTE5MDY3MjIsImV4cCI6MTU
yMDU0NjcyMn0.-AaTOAnhne-u8ioWMJrTozph_25mQhSTQGS2cx3tx6w",
    "content-type: application/x-www-form-urlencoded"
),
));

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

curl_close($curl);

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

```