

[Flex Cryptopay Payment Flow](#)

1. Initiate a crypto currency payment request through the Flex Cryptopay system.
2. Upon submission, you will receive a URL for the payment page, which should be utilized for redirection.
3. Access the provided URL to render the crypto currency payment interface for the user.
4. An API endpoint is provided to retrieve the payment status using the payment ID generated during the initial request.

Additionally, it is required to include both a successURL and a cancelURL within the payment request API call.

The payment page will subsequently redirect to the appropriate URL based on the payment outcome:

successURL: Directs the user upon successful detection of the payment on the block chain.

cancelURL: Directs the user if no payment is made and the payment session expires.

[Flex Crypto pay API Integration](#)

[Authentication and Authorization](#)

Two params need to be send as HTTP custom headers in each API request.

1. CLIENTID : Client unique Id (should be stored in app)
2. APIKEY :

APIKEY needs to be send in MD5 hash format (including current date) and calculated as mentioned below. This way the value of APIKEY shall change every day.

md5(SECRET + date("m-d-Y"))

Where **SECRET** is also unique for each client and should also be stored in app and date should be current date (in **CST (America/Chicago)** Timezone)

For eg let's say **SECRET** = "DI3hK8921M" and current date="04-01-2021", then the actual APIKEY will be computed as **MD5(DI3hK8921M04-01-2021)**

The values of CLIENTID and SECRET will be shared separately in file.

[1.0 Crypto Payment API](#)

This API shall be used to initiate the Crypto payment API for the BTC.

API URL

{BASE_URL}/ extras/rest-api/transactions/refill

REQUEST

Request Method: POST

Request Headers: Please refer Authentication and Authorization_section to include headers for auth

HTTP Payload:

```
{
    "phone": "8159848908",          -- mobile number of customer
    "carrierId": 333423,             -- hardcode this field
    "planId": "888881",              -- hardcode this field
    "amount": 30,                    -- Amount to be deposited
    "vp_username": "testuser",       -- username of customer
    "successURL": "",                -- URL for sending success
    "cancelURL": "",                 -- URL for sending failure
    "reference_id": 1234              -- Unique identifier (integer) set by
client
}
```

RESPONSE

Response Body

```
{
    "Result": "Success", -- HTTP API response status
    "TransResult": "ok", -- Status of the transaction (expected values are
"ok", "Error", "pending").
    "CCTransactionId": "", -- Transaction ID of the crypto payment
    "bitcoinRate": "91201.2083", -- Crypto Exchange rate
    "bitcoinValue": "0.00021930", -- Crypto value of the amount
    "hosted_url": ""         -- URL to redirect after the crypto payment
}
```

Once the user is redirected to the hosted_url, hosted_url page would retry to fetch the status of the crypto payment after every 60 sec.

Once the payment is successful, it shall re-direct to the successURL page provided in the refill API. If the user doesn't make any payment on the BTC address shown on the hosted_url page, the page would time out and redirect to the cancel URL.

ERC-20 USDT

For the ERC-20 USDT support, we need to send the following request body.

```
{
    "phone": "8159848908",          -- mobile number of customer
    "carrierId": 333424,             -- hardcode this field
    "planId": "erc20usdt",          -- hardcode this field
    "amount": 30,                    -- Amount to be deposited
    "vp_username": "testuser",       -- username of customer
    "successURL": "",                -- URL for sending success
    "cancelURL": "",                 -- URL for sending failure
    "reference_id": 1234              -- Unique identifier (integer) set by
client
}
```

Change is only in the carrierId and the planId fields.

TRC-20 USDT

For the TRC-20 USDT support, we need to send the following request body.

```
{
    "phone": "8159848908",          -- mobile number of customer
    "carrierId": 333425,             -- hardcode this field
    "planId": "trc20usdt",          -- hardcode this field
    "amount": 30,                   -- Amount to be deposited
    "vp_username": "testuser",      -- username of customer
    "successURL": "",               -- URL for sending success
    "cancelURL": "",                -- URL for sending failure
    "reference_id": 1234             -- Unique identifier (integer) set by
client
}
```

Change is only in the carrierId and the planId fields.

BTC lightning

For the BTC lightning, we need to send the following request body.

```
{
    "phone": "8159848908",          -- mobile number of customer
    "carrierId": 333426,             -- hardcode this field
    "planId": "btclite",            -- hardcode this field
    "amount": 30,                   -- Amount to be deposited
    "vp_username": "testuser",      -- username of customer
    "successURL": "",               -- URL for sending success
    "cancelURL": "",                -- URL for sending failure
    "reference_id": 1234             -- Unique identifier (integer) set by
client
}
```

Change is only in the carrierId and the planId fields.

Response for ERC, TRC and BTC lightning

For the response, the keys has been changed to Rate and Value (Rate: refers to the crypto exchange rate and Value: refers to the Crypto amount value)

```
{
    "Result": "Success", -- HTTP API response status
    "TransResult": "ok", -- Status of the transaction (expected values are
"ok", "Error", "pending").
    "CCTransactionId": "", -- Transaction ID of the crypto payment
    "Rate" : "0.99",       -- Crypto Exchange rate
    "Value" : "9.90",      -- Crypto value of the amount
    "hosted_url": ""       -- URL to redirect after the crypto payment
}
```

The response body shall be similar for all the Crypto currency.

2. GET Payment Status

This API shall be used to GET the Crypto payment API. (Status for the payment for BTC, ERC-20 USDT, TRC-20 USDT and BTC lightning)

API URL

{BASE_URL}/ extras/rest-api/transactions/ btcTxnStatus

REQUEST

Request Method: GET

Request Headers: Please refer Authentication and Authorization_section to include headers for auth

GET Request

{BASE_URL}/ extras/rest-api/transactions/ btcTxnStatus/{address} (CCTransactionId – this would have the crypto address whose status needs to be fetched)

Current state machine of the transaction

1. first state is "open" - when transaction is not initiated
2. when user makes payment status will change to "pending"
3. when amount is received status changes to "complete"

RESPONSE

Response Body

```
{
  "status": "open", -- Status of the crypto transaction (expected values are
"open", "pending", "completed", "failed").
}
```

When the status of the transaction, would change to pending or completed, the output would be

```
{
  "status": "pending",
  "receivedAmt": 20
}
```

```
{
```

```
    "status": "completed",  
    "receivedAmt": 20  
}
```