Integration API v.10.2

Introduction

The Crypto Payment Gateway service allows you to improve your business by enabling crypto payments. Service carries all of the effort required to maintain the technical staff for implementing and supporting the blockchain infrastructure. Service generates crypto currency addresses on your behalf (you get your seed phrase in the initial registration) and monitors the blockchain in real time mode for all transactions executed.

Use Scenarios

The Crypto Payment Gateway practices several approaches to the acceptance of payments. At the integration stage, you will need to choose the one you like best for your business. The following options may affect your integration design.

Payment Page. For your convenience, Service implements the payment page on its side. You receive a URL when calling **Create Payment API** in the field (PAYMENT_URL) and direct your customer to that page, but you also have the ability to implement the payment page on your side with all information for such page provided in the payment on **Create Payment API**.

Payment Type. Service supports two types of payment: fixed amount and open amount. The appropriate payment type will be set by the PAYMENT_TYPE parameter in the call **Create Payment API** and can take two values (FIXED_AMOUNT or OPEN_AMOUNT).
The default option is FIXED_AMOUNT.

- Fixed amount payment (FIXED_AMOUNT). The payment will only be successful when the payer pays an amount equal to or exceeding the one ordered within the time allowed for payment, otherwise the payment will be considered failed. The payer may make several payments within the time allowed for payment. The service will wait for payment of the entire amount ordered until the time allowed for payment expires.
- Open amount payment (OPEN_AMOUNT). The payment will be successful upon receipt of any amount within the time allowed for such payment. The payment will only fail when the payment lifetime expires and no funds are received at the address within that time.

Amount in any currency. You may request creation of a payment from Crypto Payment Gateway indicating the desired amount in a fiat or crypto currency.

- Fiat amount. This type of payment is designed for when you need your payer to pay in a crypto currency while your price is in a fiat currency, and you do not want to bother with any exchange rates.
  For instance, if you need your customer to pay 100 USD in the BTC equivalent, you need to set the following parameters when calling **Create Payment API**:
  AMOUNT – fiat amount (e.g. 100)
  ASSET_ID – fiat currency (e.g. USD)
  PAYMENT_ASSET_ID – crypto currency (e.g., BTC).
  Crypto Payment Gateway will translate 100 USD into BTC at the going rate, and then will ask the customer to pay an amount in BTC.
- Crypto currency amount. This type of payment is designed for when your price is stated in either a crypto or a fiat currency but you want it converted into a crypto currency on your side. For instance, you want your customer to pay 100 USD in BTC, so for an API call you will need to translate 100 USD into BTC yourself, e.g., into 0,037 BTC. The parameters required when calling **Create Payment API**:
  AMOUNT – amount in a currency (e.g. 0,037)
  ASSET_ID – currency (e.g. BTC)
  PAYMENT_ASSET_ID – crypto currency (e.g. BTC).
  Crypto Payment Gateway will ask the customer to pay the amount in BTC.

Markup. To prevent crypto currency volatility from affecting your revenue, you may set a markup for each crypto currency. To set the markup you need to either indicate the absolute amount in crypto currency or a percentage in the merchant's interface. The Store: Settings -> Assets -> The crypto, fields Markup amount and/or Markup percentage. For instance, when calling **Create Payment API** you indicate AMOUNT=100, ASSET_ID = USD, PAYMENT_ASSET_ID = BTC, and in that case, the service will convert the amount of 100 USD into BTC as 0,037 BTC, and will add a minimum markup or a Markup amount, or a Markup percentage from 0,037. If you have stated both, the maximum will be the default selection.

Underpayment threshold. You can also compliment the customer on the crypto currency volatility by providing some tolerance when making payment, through setting certain values in the merchant's interface. The Store: Settings -> Payments, fields Underpayment threshold and/or Underpayment threshold percentage. The option is only available for fixed amount payments.

For instance, when calling **Create Payment API** you indicate AMOUNT=100, ASSET_ID = USD, PAYMENT_ASSET_ID = BTC, and in that case, the service will convert the amount of 100 USD into BTC as 0,037 BTC. To complete the payment, the customer must pay 0,037 BTC less a certain threshold amount equal to the minimum of either the underpayment threshold or the percentage of the underpayment threshold percentage of 0,037.

Amount paid. Crypto Payment Gateway will call your system upon completion or cancelation of payment, and for any incoming crypto currency transaction. The call back will contain the payment, and you may use the fields therein or request information from **Get Payment**, **Get Payment Transactions API**.

The field Payment Status (OPEN) shows that the payment is being processed and the customer's transactions are being waited for, status indications (COMPLETE, CANCEL) mean that the payment has been completed. The fields (AMOUNT, ASSET_ID, PAYMENT_ASSET_ID) show what you s-indicated in the call **Create Payment API**. The fields (PAYMENT_AMOUNT, RATE, RATE_DATE) are the exchange rate and the sums converted. The fields (IP_AMOUNT, IP_ASSET_ID) are what the customer has paid in crypto currency until now, while the fields (PAID_AMOUNT, ASSET_ID) show what has been paid in a fiat currency converted at the exchange rate (RATE, RATE_DATE) effective on the date when the payment was created. To reflect the fiat payment amount in your system, your system will either take the fiat amount (PAID_AMOUNT, ASSET_ID) from the payment, or it will convert the crypto currency amount (IP_AMOUNT, IP_ASSET_ID) on your side at any going rate, e.g., on the date when the crypto currency transaction is published in the blockchain or the payment is completed.

Time allowed for payment. After calling Create Payment API the Crypto Payment Gateway will create a payment and will wait for incoming customer transactions in the blockchain. The timeout is about 60-90 minutes and is indicated in the service, and you may also set your own timeout for the call **Create Payment API** in the field (TIMEOUT).

Dedicated address. The Crypto Payment Gateway allows you to dedicate or assign a particular crypto address for a certain client, so that all incoming crypto on that address will be translated into payment for the client. To activate such option, you will be requested to call **Dedicate Address API**, and as a result, system will dedicate a particular address for the client and return (PAYMENT_URL) that you can provide to the client for payments. You can enhance PAYMENT_URL for user convenience with additional parameters: amount and asset_id like (amount=200&asset_id=USDT.TRC20). You can also implement the payment page on your side, as well. In order to stop that behaviour you need to call **Release Dedicate Address API.** Address can be allocated from pool or generated as new by specifying parameter **address_alloc**. You can also reserve address for particular payments. To achieve that it needs to pass parameter **dedicate_type** as **ASSIGN** on dedicate creation and then pass the **dedicate_id** to payment creation, so that the address be used for the payment.

Fund withdrawals. The Crypto Payment Gateway service allows customer withdrawals through an API call so that you could be able to implement the withdrawal logic on your side, while service will take care of all technical aspects in the blockchain. For withdrawals through the API, the following risks must be considered:

- Security. When you withdraw crypto currency in the Crypto Payment Gateway user interface, such withdrawals may be protected with a two-factor authentication, but if you call the withdrawal API security shall be your responsibility. The API call is protected with an API key/API secret therefore compromising such key/secret may result in losing your money.

- Fee amount. When you withdraw crypto currency via the Crypto Payment Gateway user interface, you are able to manage the fee amount and control the number of transactions required to make a withdrawal, while you are unable to do so in an API call, and to avoid additional transactions/fees you should keep the balance in a single address sufficient for fund withdrawals.

Test Store

The best way to implement and test your integration is the creation of a test store. With a test store you may test your integration by emulating payments implemented in the Crypto Payment Gateway for a test store, so that you do not need to make real transactions in the blockchain. To do this, you need to go through the following steps:

1. Create a test store
2. Set a store key /secret in Settings -> API: API Key/API Secret, as well as Webhook URL/Secret in Settings -> Webhook: Webhook URL/Webhook Secret
3. Set the environment on your side for using the API Key/test store secret and URL/ Webhook secret.
4. Integration test.

Acceptance of Payments

To start accepting crypto currency payments with the Crypto Payment Gateway, you need to go through the following steps:

1. Create an account in the service.
2. Set your store or at least API Key/store secret: The Store -> Settings -> API -> API Key, API Secret.
3. Integrate your system with the Crypto Payment Gateway implementing API calls and accepting webhooks.
4. Call **Create Payment API** on your side and then either re-direct the customer to the payment page or implement that page on your side.
5. Wait for the payer's transaction on the blockchain.
6. Process the webhook from the service and record it in your system.

API Documentation

Introduction

API the Crypto Payment Gateway is built on REST principles; it operates with real objects and has predictable behaviour. With the API you can send payment request, obtain payment information, information on crypto currency transactions in the blockchain and much more. API uses HTTP as the base protocol, meaning that it is suitable for development in any programming language that is able to operate with HTTP libraries (cURL and others).

The base URL for all API calls is https://api.[service_name]/int. All requests must bear a title ("Content-Type" as "application/json").

Authentication

The Crypto Payment Gateway uses the base HTTP access authentication. Requests must be authenticated with HTTP Basic Auth. In the request title, you will need to transmit your store identifier Settings -> API fields API Key and API Secret as username and password.

The secret key is responsible for your data security. You should keep it in a safe place and not publish it on third party resources (for instance, together with code examples).

Your duty is to change the API Secret on schedule, e.g., once a month. You can change or re-generate your API Secret in the use interface: The Store -> Settings -> API, field API Secret.

Here is an example of an authenticated request:

cURL https://api.[service_name]/int/payment/{payment_id} \

-u <API Key>:<API Secret>

Processing of Requests

The Crypto Payment Gateway will promptly process any request it receives and will return the processing result ("successful" or "failed"). The response will contain the response HTTP code, standard titles and, if desired, the response body in the JSON format. If no accurate response can be given within 30 seconds, for instance because of problems on the side of the blockchain provider, service will return the response code HTTP 500, and for requests related to payments, it will also try to cancel the operation.

HTTP 500 does not show whether your operation was successful or failed. Therefore upon receiving a HTTP 500, you must first find the result of your request processing and only than make any decisions on the transaction.

Responses to Errors

All validations (business errors) will give out HTTP code 422, as well as the body message in the code and message fields, as in below:

```
{
  "code": "ERR_NO_PAYMENT",
  "message": "Payment <e5792528-d265-4813-8b59-055cdfef31b1> not found."
}
```

Let us now view the methods.

Get assets

GET/assets

Obtain the list of assets or crypto currencies available for accepting payments in the Crypto Payment Gateway service.
No parameters required.
The response will be the array of assets.

Asset_id string	Asset ID, to be used for creating payment
asset_name string	Asset name
currency string	Currency (Ticker)
currency_name string	Currency name
asset_type string	Type of asset, FIAT or CRYPTO
asset_type_name string	Name of asset and link to the type
symbol string	Symbol of currency
sort_order integer	Order of sorting in the array
min_size float64	Minimum amount in the currency
precision float64	Currency precision
timeout_to_trx float64	Time allowed for receipt of payment or deposit
timeout_to_wdr float64	Time allowed for completion of withdrawal
crypto_network string	Crypto network, e.g. BITCOIN or ETHEREUM
crypto_network_name string	Crypto network name
crypto_address_link string	Template for creating an address link
crypto_transaction_link string	Template for creating a link to the crypto network transaction
min_withdrawal_amount float64	Minimum withdrawal amount
max_withdrawal_amount float64	Maximum withdrawal amount

Example:


```
[
  {
    "asset_id": "BTC",
    "asset_name": "Bitcoin",
    "currency": "BTC",
    "currency_name": "Bitcoin",
    "language": "en",
    "asset_type": "CRYPTO",
    "asset_type_name": "Crypto Currenies",
    "symbol": "Ƀ",
    "sort_order": 20,
    "min_size": 1e-08,
    "precision": 8,
    "timeout_to_trx": 90,
    "timeout_to_wdr": 1440,
    "crypto_network": "BITCOIN",
    "crypto_network_name": "Bitcoin",
    "crypto_network_confirmations": 6,
    "crypto_address_link": "https://live.blockcypher.com/btc/address/{{address}}",
    "crypto_transaction_link": "https://live.blockcypher.com/btc/tx/{{txId}}",
    "min_withdrawal_amount": 0.0001,
    "max_withdrawal_amount": 2400
  },
...
]
```

Get accounts

GET/accounts

Obtaining the list of crypto currency balances.
No parameters required.
The response will be the array of balances.

Balance float	Aggregate balance in the store currency	
available float	Aggregate amount available for withdrawal	
on_hold float	Aggregate amount on hold	
asset_id string	Store currency	
accounts array	**account_id** string	ID of the coin balance
	store_id string	Store ID
	account_num integer	Array sort order
	crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
	Crypto_network_name string	Crypto network name
	balance string	Balance as a string
	available string	Available amount; available = balance – on hold
	on_hold string	Amount on hold
	asset_id string	Asset ID
	currency string	Currency
	base_balance string	Balance in the store currency (in fiat)
	base_available string	Available amount in the store currency (in fiat)
	base_on_hold string	Amount on hold in the store currency (in fiat)
	base_asset_id string	ID of the store currency asset
	base_currency string	Currency
	rate float64	Exchange rate used when converting from fiat into crypto currency at the rate date
	rate_date string	Date/time of the exchange rate

Example:


```
{
  “balance”: “281.34”,
  “available”: “281.34”,
  “on_hold”: “0”,
  “asset_id”: “USD”,
  “accounts”: [
    {
      “account_id”: “dadd1e36-33e1-4e6d-8e97-2d00ec9446bb”,
      “store_id”: “b6282b7f-f790-4acc-ab76-26aca5cf1f99”,
      “account_num”: 4,
      “crypto_network”: “BITCOIN”,
      “asset_id”: “BTC”,
      “balance”: “0.00772465”,
      “available”: “0.00772465”,
      “on_hold”: “0”,
      “currency”: “BTC”,
      “base_asset_id”: “USD”,
      “base_balance”: “201.20”,
      “base_available”: “201.20”,
      “base_on_hold”: “0”,
      “base_currency”: “USD”,
```

```
      "rate": 26046.990234375,
      "rate_date": "2023-08-22T14:03:52Z",
    },
    {
      "account_id": "943d119b-a65d-4412-b651-f4cc8d1f1c5f",
      "store_id": "b6282b7f-f790-4acc-ab76-26aca5cf1f99",
      "account_num": 5,
      "crypto_network": "ETHEREUM",
      "asset_id": "ETH",
      "balance": "0.009821214514536778",
      "available": "0.009821214514536778",
      "on_hold": "0",
      "currency": "ETH",
      "base_asset_id": "USD",
      "base_balance": "16.29",
      "base_available": "16.29",
      "base_on_hold": "0",
      "base_currency": "USD",
      "rate": 1658.7900390625,
      "rate_date": "2023-08-22T14:03:52Z",
    },
  ]
}
```

Convert amount

POST/convert

Converting an amount from one currency into another

Request:

from_amount float64	Amount to be converted
from_asset_id string	Asset ID. From which currency to be converted
asset_id string	Asset ID. Into which currency to be converted

Example:


```
{
  "from_amount": 100,
  "from_asset_id": "USD",
  "asset_id": "BTC"
}
```


Response:

from_amount float64	Amount to be converted
from_asset_id string	Asset ID. From which currency to be converted
amount float64	Amount resulting from the conversion
asset_id string	Asset ID. Into which currency to be converted
rate float64	Exchange rate used in the conversion
rate_date string	Exchange rate date/time

Example:


```
{
   "from_amount": 100,
   "from_amount_str": "100",
   "from_asset_id": "USD",
   "amount": 0.00369555,
   "amount_str": "0.00369555",
   "asset_id": "BTC",
   "rate": 3.695550182251254e-05,
   "rate_date": "2023-06-02T13:13:30Z"
}
```

Create payment

POST/payment

Creates payment in the Crypto Payment Gateway service and returns the payment.

Request:

payment_type string	Payment type. Values FIXED_AMOUNT or OPEN_AMOUNT
amount float64	Amount payable
asset_id string	Asset ID. Amount currency
payment_asset_id string	Asset ID. Crypto currency of the payment
payer_email string	Payer's Email
payer_name string	Payer's name
payer_lang string	Payer's interface language. Language two-letter codes ar Arab en English es Spanish fa Farsi fr French ja Japanese pt Portuguese ru Russian tr Turkish zh Chinese
description string	Payment description
custom string	String for storing any value transmittable in webhook, and you may add any necessary payment information. Can also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
dedicate_id string	Dedicate ID to use reserved address
webhook_url string	URL of the webhook. If not shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page
cancel_url string	URL for redirecting the payer to the canceled payment page
timeout integer	Payment waiting time (in minutes). Not a mandatory value (between 0 and 1440 minutes)

Example:

```
{
  "payment_type": "FIXED_AMOUNT",
  "amount": 100,
  "asset_id": "USD",
  "payment_asset_id": "BTC",
  "payer_email": "name@email.com",
  "payer_name": "Customer name",
  "payer_lang": "en",
  "description": "The payment for item in store",
```

```
  "custom": "{
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "field1": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
  }",
  " custom1 ": " some value 1",
  " custom2 ": "some value 2",
  "webhook_url": "https://yourserver.com/payment/[service_name]/webhook",
  "success_url": "https://yoursite.com/payment/[service_name]/success",
  "cancel_url": "https://yoursite.com/payment/[service_name]/fail",
  "timeout": 60
}
```


Response:

payment_id string	Payment ID in the Crypto Payment Gateway
payment_type string	Payment type. Values FIXED_AMOUNT or OPEN_AMOUNT
payment_type_name string	Name of payment type in service
payment_num string	Payment number in the service
store_id string	Store ID in the service
store_name string	Store name
account_id string	ID of the coin account in the service
is_test boolean	Test payment (payment to a test store)
amount string	Amount payable
asset_id string	Asset ID. Amount currency
payment_amount string	Amount payable in crypto currency
payment_asset_id string	Asset ID актива. Payment crypto currency
rate float64	Exchange rate used when converting from fiat to crypto currency
rate_date string	Date/time of the exchange rate
ip_amount string	Amount in process. Amount in crypto currency already paid (until now)
ip_asset_id string	Asset ID. Currency ip amount
paid_amount string	Amount in fiat currency already paid (until now)
remain_amount string	Remaining amount payable in crypto currency
remain_asset_id string	Asset ID. Currency of the remaining amount
status string	Payment status. Values OPEN (open, in process), COMPLETE (completed successfully), CANCEL (cancelled), CLOSING (in the process of closing)
status_name string	Payment status name
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
Crypto_network_name string	Crypto network name
crypto_address_link string	Link to address in the blockchain
crypto_transaction_link string	Link to transaction in the blockchain
address_id string	Address ID in the service for accepting payments
address string	Crypto currency address in the blockchain
confirms integer	Required number of confirmations (blocks) to confirm the transaction in the blockchain
start_date string	Payment creation time
end_date string	Payment expiry time
complete_date string	Payment completion time
payment_url string	URL for redirecting the payer to the payment page
qrcode_url string	URL to the png picture with the payment QR code
payer_email string	Payer's Email

payer_name string	Payer's name
payer_lang string	Payer's interface language
description string	Payment description
custom string	String for storing any value transmittable in webhook you can add any payment information required. May also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
dedicate_id string	Dedicate ID to use reserved address
webhook_url string	URL of the webhook. If no value shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page
cancel_url string	URL for redirecting the payer to the cancelled payment page
timeout integer	Payment waiting time (in minutes). Not a mandatory value

Example:


```
{
  "payment_id": "f2a681c0-34e6-4668-8cb4-f76bac6b81c3",
  "store_id": "b6282b7f-f790-4acc-ab76-26aca5cf1f99",
  "account_id": "dadd1e36-33e1-4e6d-8e97-2d00ec9446bb",
  "payment_num": "164",
  "payment_type": "FIXED_AMOUNT",
  "payment_type_name": "Fixed Amount",
  "store_name": "Main",
  "is_test": false,
  "amount": "100",
  "asset_id": "USD",
  "paid_amount": "0",
  "payment_amount": "0.00370885",
  "payment_asset_id": "BTC",
  "rate": 3.708847905797581e-05,
  "rate_date": "2023-06-02T14:03:33Z",
  "ip_amount": "0",
  "ip_asset_id": "BTC",
  "remain_amount": "0.00370885",
  "remain_asset_id": "BTC",
  "status": "OPEN",
  "status_name": "Open",
  "crypto_network": "BITCOIN",
  "crypto_network_name": "Bitcoin",
  "address_id": "df08d662-7abd-40b3-9a96-c7eada9c334e",
  "address": "bc1qv2pqpdf3qq5q96djfgnlehn92a7e7n67wrcnqe",
  "confirms": 6,
  "start_date": "2023-06-02T14:06:25Z",
  "end_date": "2023-06-02T15:06:25Z",
  "complete_date": "2023-09-12T10:45:33Z",
  "payment_url": "https://payer. [service_name].com/payment/f2a681c0-34e6-4668-8cb4-f76bac6b81c3?code=e16d90d2-13a0-4ae1-bf70-b6e7106d2966",
  "qrcode_url": "https://api.[service_name].com/int/payment/f2a681c0-34e6-4668-8cb4-f76bac6b81c3/qrcode?code=e16d90d2-13a0-4ae1-bf70-b6e7106d2966&oi=daee91ff-d4b7-4bb1-8c4a-855f57f9af6b&l=0",
```

```
    “payer_email”: “name@email.com”,
    “payer_name”: “Customer name”,
    “payer_lang”: “en”,
    “description”: “The payment for item in store”,
    “custom”: “{
      “client_id”: “9a33c8b0-151c-481d-aef9-da15d883dc42”,
      “org_id”: 3,
      “idempotence”: “341ba962-6ebf-4f3a-aef9-41c835a1dc26”
    }”,
    “ custom1 “: “ some value 1”,
    “ custom2 “: “some value 2”,
  “webhook_url”: “https://yourserver.com/[service_name]/result”,
    “success_url”: “https://yoursite.com/payment/success”,
    “cancel_url”: “https://yoursite.com/payment/fail”,
    “timeout”: 0
}
```

Create fiat payment

POST/payment

Creates fiat payment in the Crypto Payment Gateway service and returns the payment.

Request:

payment_type string	Payment type. Values FIXED_AMOUNT or OPEN_AMOUNT
amount float64	Amount payable
asset_id string	Asset ID. Amount currency
payment_asset_id string	Asset ID. Crypto currency of the payment
payer_email string	Payer's Email
payer_name string	Payer's name
payer_lang string	Payer's interface language. Language two-letter codes ar Arab en English es Spanish fa Farsi fr French ja Japanese pt Portuguese ru Russian tr Turkish zh Chinese
description string	Payment description
custom string	String for storing any value transmittable in webhook, and you may add any necessary payment information. Can also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
dedicate_id string	Dedicate ID to use reserved address
webhook_url string	URL of the webhook. If not shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page
cancel_url string	URL for redirecting the payer to the canceled payment page
timeout integer	Payment waiting time (in minutes). Not a mandatory value (between 0 and 1440 minutes)
fiat_provider string	Fiat Provider. Available providers (sguardarian, cmoonpay, cbanxa, cwert, cransak, ckado)
fiat_asset_id string	Asset ID. Fiat currency of the payment (USD, EUR, JPY, KRW, TRY)

Example:

```
{
  "payment_type": "FIXED_AMOUNT",
  "amount": 100,
  "asset_id": "USD",
  "payment_asset_id": "BTC",
  "payer_email": "name@email.com",
```

```
  "payer_name": "Customer name",
  "payer_lang": "en",
  "description": "The payment for item in store",
  "custom": "{
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "field1": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
  }",
  " custom1 ": " some value 1",
  " custom2 ": "some value 2",
  "webhook_url": "https://yourserver.com/payment/[service_name]/webhook",
  "success_url": "https://yoursite.com/payment/[service_name]/success",
  "cancel_url": "https://yoursite.com/payment/[service_name]/fail",
  "timeout": 60,
  "fiat_provider": "sguardian",
  "fiat_asset_id": "USD"
}
```


Result is the same as for Create Payment.

Get Payments

GET/payments

Returns array the payment. It accepts parameters in URL as following: /payments?from=2021-01-31&to=2025-01-31&status=COMPLETE&asset_id=USDT.TRC20&dedicate_id=0c1c57ac-3f88-4a5f-b0ad-5ec94c10eae2

Parameters:

from string	**required**	Date/time of the start date. Format YYYY-MM-DD
to string	**required**	Date/time of the end date. Format YYYY-MM-DD
status string	**optional**	Payment status. Values OPEN (open, in process), COMPLETE (completed successfully), CANCEL (cancelled)
asset_id string	**optional**	Asset ID. Values of BTC, ETH, TRX, USDT.ERC20 or USDT.TRC20
dedicate_id string	**optional**	Dedicate Address ID in the service

Response is array of:

payment_id string	Payment ID in the service
payment_type string	Payment type. Values FIXED_AMOUNT or OPEN_AMOUNT
payment_type_name string	Name of payment type in the service
payment_num string	Payment number in the service
store_id string	Store ID in the service
store_name string	Store name
account_id string	ID of the coin account in the service
is_test boolean	Test payment (payment to a test store)
amount string	Amount payable
asset_id string	Asset ID. Amount currency
payment_amount string	Amount payable in crypto currency
payment_asset_id string	Asset ID актива. Payment crypto currency
rate float64	Exchange rate used when converting from fiat to crypto currency
rate_date string	Date/time of the exchange rate
ip_amount string	Amount in process. Amount in crypto currency already paid (until now)
ip_asset_id string	Asset ID. Currency ip amount
paid_amount string	Amount in fiat currency already paid (until now)
remain_amount string	Remaining amount payable in crypto currency
remain_asset_id string	Asset ID. Currency of the remaining amount
status_type string	Payment status. Values OPEN (open, in process), COMPLETE (completed successfully), CANCEL (cancelled), CLOSING (in the process of closing)
status_type_name string	Payment status name
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
crypto_network_name string	Crypto network name
crypto_address_link string	Link to address in the blockchain
crypto_transaction_link string	Link to transaction in the blockchain
address_id string	Address ID in the service for accepting payments
address string	Crypto currency address in the blockchain
confirms integer	Required number of confirmations (blocks) to confirm the transaction in the blockchain
start_date string	Payment creation time
end_date string	Payment expiry time
complete_date string	Payment completion time

payment_url string	URL for redirecting the payer to the payment page
qrcode_url string	URL to the png picture with the payment QR code
payer_email string	Payer's Email
payer_name string	Payer's name
payer_lang string	Payer's interface language
description string	Payment description
custom string	String for storing any value transmittable in webhook, you can add any payment information required. May also be a json object as string
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
dedicate_id string	Dedicate ID to use reserved address
webhook_url string	URL of the webhook. If no value shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page
cancel_url string	URL for redirecting the payer to the cancelled payment page
timeout integer	Payment waiting time (in minutes). Not a mandatory value

Example:


```
[
    {
      "payment_id": "f2a681c0-34e6-4668-8cb4-f76bac6b81c3",
      "store_id": "b6282b7f-f790-4acc-ab76-26aca5cf1f99",
      "account_id": "dadd1e36-33e1-4e6d-8e97-2d00ec9446bb",
      "payment_num": "164",
      "payment_type": "FIXED_AMOUNT",
      "payment_type_name": "Fixed Amount",
      "store_name": "Main",
      "is_test": false,
      "amount": "100",
      "asset_id": "USD",
      "paid_amount": "0",
      "payment_amount": "0.00370885",
      "payment_asset_id": "BTC",
      "rate": 3.708847905797581e-05,
      "rate_date": "2023-06-02T14:03:33Z",
      "ip_amount": "0",
      "ip_asset_id": "BTC",
      "remain_amount": "0.00370885",
      "remain_asset_id": "BTC",
      "status": "OPEN",
      "status_name": "Open",
      "crypto_network": "BITCOIN",
      "crypto_network_name": "Bitcoin",
      "address_id": "df08d662-7abd-40b3-9a96-c7eada9c334e",
      "address": "bc1qv2pqpdf3qq5q96djfgnlehn92a7e7n67wrcnqe",
      "confirms": 6,
      "start_date": "2023-06-02T14:06:25Z",
      "end_date": "2023-06-02T15:06:25Z",
```

```
    "complete_date": "2023-09-12T10:45:33Z",
    "payment_url": "https://payer.[service_name].com/payment/f2a681c0-34e6-4668-8cb4-f76bac6b81c3?code=e16d90d2-13a0-4ae1-bf70-b6e7106d2966",
    "qrcode_url": "https://api.[service_name].com/int/payment/f2a681c0-34e6-4668-8cb4-f76bac6b81c3/qrcode?code=e16d90d2-13a0-4ae1-bf70-b6e7106d2966&oi=daee91ff-d4b7-4bb1-8c4a-855f57f9af6b&l=0",
    "payer_email": "name@email.com",
    "payer_name": "Customer name",
    "payer_lang": "en",
    "description": "The payment for item in store",
    "custom": "{
      "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
      "org_id": 3,
      "idempotence": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
    }",
    " custom1 ": " some value 1",
    " custom2 ": "some value 2",
"webhook_url": "https://yourserver.com/[service_name]/result",
    "success_url": "https://yoursite.com/payment/success",
    "cancel_url": "https://yoursite.com/payment/fail",
    "timeout": 0
  },
  {
  .....
  }
]
```

Receipt of Payment (Get Payment)

GET/payment/{payment_id}

Returns the payment. To create a URL you need to use the field payment_id returned **Create Payment API**.
Returns the payment, see in more detail in **Create Payment API**.

Receipt of Payment Transactions (Get Payment Transactions)

GET/payment/{payment_id}/trxs

Returns the payment transactions array in the blockchain. To create a URL you need to use the field payment_id returned **Create Payment API**.

Response:

status_type string	Transaction status
status_type_name string	Transaction status name
trx_date string	Time when the transaction was completed in the service
trx_received string	Time when the transaction was published in the blockchain block
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM и т.д.
crypto_network_name string	Crypto network name
address string	Address of the crypto currency in the blockchain
trx_hash string	Transaction hash in the blockchain
crypto_address_link string	Link to the address in the blockchain
crypto_transaction_link string	Link to the transaction in the blockchain
amount string	Transaction amount in the blockchain
asset_id string	Asset ID. Amount currency
currency string	Amount currency

Example:


```
[
  {
    "status": "COMPLETE",
    "status_name": "Complete",
    "trx_date": "2022-12-13T10:17:46Z",
    "crypto_network": "BITCOIN",
    "crypto_network_name": "Bitcoin",
    "address": "1L2es2Z9oZmjmUSXkeCQ4itVzfUPYG4Fif",
    "trx_hash": "cba4ae07643a93316062aaba156d1329c2dbab9af05a6b70d771ccc988ff2bf5",
    "trx_received": "2022-12-13T10:17:46Z",
    "crypto_address_link": "https://live.blockcypher.com/btc/address/1L2es2Z9oZmjmUSXkeCQ4itVzfUPYG4Fif",
    "crypto_transaction_link": "https://live.blockcypher.com/btc/tx/cba4ae07643a93316062aaba156d1329c2dbab9af05a6b70d771ccc988ff2bf5",
    "amount": "0.0008",
    "currency": "BTC",
    "asset_id": "BTC"
  }
]
```

Create withdrawal

POST/withdraw

Creates a withdrawal of funds in the Crypto Payment Gateway service and returns the withdrawal.

Request:

amount float64	Amount to be withdrawn
asset_id string	Asset ID. Amount currency
withdraw_asset_id string	Asset ID. Withdrawal crypto currency
address string	Address to be withdrawn to
payer_email string	Payer's Email
payer_name string	Payer's name
description string	Description
custom string	String for storing any value transmittable in webhook, you can add any payment information required. May also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
totp_code string	TOTP Code to confirm withdraw
webhook_url string	URL of the webhook. If no value shown, will be taken from the store settings
reallocate boolean	true/false value. It makes sense for Contract assets only, like USDT. If value is true, the system will try to reallocate fee assets like ETH, TRX in such a way it's enough to execute transaction.
consolidate boolean	true/false value. If value is true, it creates withdraw to the destination address within one transaction. In order to do so, it accumulates requested amount on one address, so that final transfer, can be done by single transaction.
Timeout integer	Waiting time (in minutes). Not a mandatory value (between 0 and 1440 minutes)

Example:


```
{
  "amount": 100,
  "asset_id": "USD",
  "withdraw_asset_id": "BTC",
  "payer_email": "name@email.com",
  "payer_name": "Customer name",
  "description": "The payment for item in store",
  "custom":" {
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "field1": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
          }",
  " custom1 ": " some value 1",
  " custom2 ": "some value 2",
 "webhook_url": "https://yourserver.com/payment/[service_name]/webhook",
  "auto_realloc": "N",
  "timeout": 60
}
```

Response:

withdraw_id string	Withdrawal ID in the service
store_id string	Store ID in the service
store_name string	Store name
account_id string	ID of the coin's account in the service
is_test boolean	Test withdrawal
amount string	Amount to be withdrawn
asset_id string	Asset ID. Amount currency
fee string	Fee amount. Amount in crypto currency payable for the transaction ю
fee_asset_id string	Asset ID. Fee amount currency
status_type string	Status. Values OPEN (open, in process), COMPLETE (completed successfully), CANCEL (cancelled), CLOSING (in the process of closing)
status_type_name string	Status name
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
Crypto_network_name string	Crypto network name
address string	Address of the crypto currency in the block chain
confirms integer	Required number of blocks (confirmations) to confirm the transaction in the block chain
start_date string	Withdrawal creation time
end_date string	Withdrawal expiry time
complete_date string	Payment completion time
payer_email string	User's Email
payer_name string	User's name
description string	Description
custom string	String for storing any value transmittable in webhook, you can add any withdrawal information required. May also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
webhook_url string	URL of the webhook. If no value shown, will be taken from the store settings
timeout integer	Waiting time (in minutes). Not a mandatory value

Example:


```
{
  "withdraw_id": "6e50075a-333e-43ba-ab1c-74f4fbaa783f",
  "store_id": "fbe7d39e-6370-49fe-8d18-dc78930021a2",
  "account_id": "a97bb84e-df28-4bbd-8de6-37ad9176ace2",
  "withdraw_num": "214",
  "store_name": "Test",
  "is_test": true,
  "amount": "0.00333202",
  "asset_id": "BTC",
  "fee": "0.0001",
  "fee_asset_id": "BTC",
  "crypto_network": "BITCOIN",
  "crypto_network_name": "Bitcoin",
```

```
  "status": "COMPLETE",
  "status_name": "Completed",
  "address": "bc1qec0djsvjhh0dg8mvsxc4u0tvk49t9jh7vd7gzp",
  "confirms": 0,
  "start_date": "2023-06-23T08:48:05Z",
  "end_date": "2023-06-23T09:48:05Z",
  "complete_date": "2023-06-23T08:48:05Z",
  "description": "The payment for item in store",
  "payer_email": "name@email.com",
  "payer_name": "Customer name",
  "custom": "{
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "idempotence": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
  }",
  " custom1 ": " some value 1",
      " custom2 ": "some value 2",
      "webhook_url": "https://yourserver.com/[service_name]/result",
  "timeout": 0
}
```


Get Withdrawal

GET/withdraw/{withdraw_id}

Returns the withdrawal. See in more detail in **Create Withdrawal API**.

Create dedicate address

POST /dedicate

Creates dedicated address in the service and returns the dedication object.

Request:

asset_id string	Asset ID. Address currency
payer_email string	Payer's Email
payer_name string	Payer's name
payer_lang string	Payer's interface language. Language two-letter codes ar Arab en English es Spanish fa Farsi fr French ja Japanese pt Portuguese ru Russian tr Turkish zh Chinese
description string	Payment description
address_alloc string	Address allocation. Allowed values (NEW, POOL), by default (NEW). NEW-create new address, POOL- allocate address from pool.
Dedicate_type string	Address dedication type. Allowed values (USES, ASSIGN), by default (USES). USES-dedicate address in use for all time, ASSIGN-reserve address only, while use it for particular payment only.
Custom string	String for storing any value transmittable in webhook, and you may add any necessary payment information. Can also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
webhook_url string	URL of the webhook. If not shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page

Example:


```
{
  "asset_id": "BTC",
  "payer_email": "name@email.com",
  "payer_name": "Customer name",
  "payer_lang": "en",
  "description": "The payment for item in store",
  "address_alloc": "NEW",
  "dedicate_type": "USES",
  "custom": "{
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "field1": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
```

```
  }",
  " custom1 ": " some value 1",
  " custom2 ": "some value 2",
 "webhook_url": "https://yourserver.com/payment/[service_name]/webhook",
  "success_url": "https://yoursite.com/payment/[service_name]/success"
}
```

Response:

dedicate_id string	Dedicate ID in the service
dedicate_num string	Dedicate number in the system
dedicate_type string	Address dedication type. Allowed values (USES, ASSIGN), by default (USES). USES-dedicate address in use for all time, ASSIGN- reserve address only, while use it for particular payment only.
Store_id string	Store ID in the service
store_name string	Store name
account_id string	ID of the coin account in the service
is_test boolean	Test payment (payment to a test store)
asset_id string	Asset ID. Amount currency
status_type string	Dedicate status. Values OPEN (open, in process), COMPLETE (completed successfully)
status_type_name string	Dedicate status name
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
Crypto_network_name string	Crypto network name
crypto_address_link string	Link to address in the blockchain
address_id string	Address ID in the service for accepting payments
address string	Crypto currency address in the blockchain
address_alloc string	Address allocation, NEW-create new address, POOL- allocate address from pool.
Start_date string	Dedicate creation time
end_date string	Dedicate expiry time
payment_url string	URL for redirecting the payer to the payment page
qrcode_url string	URL to the png picture with the payment QR code
payer_email string	Payer's Email
payer_name string	Payer's name
payer_lang string	Payer's interface language
description string	Payment description
custom string	String for storing any value transmittable in webhook you can add any payment information required. May also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
webhook_url string	URL of the webhook. If no value shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page

Release dedicate address

POST /dedicate/{dedicate_id}/release

Releases dedicated address in the service and returns the dedication object.

Response

dedicate_id string	Dedicate ID in the service
dedicate_num string	Dedicate number in the system
dedicate_type string	Address dedication type. Allowed values (USES, ASSIGN), by default (USES). USES-dedicate address in use for all time, ASSIGN- reserve address only, while use it for particular payment only.
store_id string	Store ID in the service
store_name string	Store name
account_id string	ID of the coin account in the service
is_test boolean	Test payment (payment to a test store)
asset_id string	Asset ID. Amount currency
status_type string	Dedicate status. Values OPEN (open, in process), COMPLETE (completed successfully)
status_type_name string	Dedicate status name
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
crypto_network_name string	Crypto network name
crypto_address_link string	Link to address in the blockchain
address_id string	Address ID in the service for accepting payments
address string	Crypto currency address in the blockchain
address_alloc string	Address allocation, NEW-create new address, POOL- allocate address from pool.
start_date string	Dedicate creation time
end_date string	Dedicate expiry time
payment_url string	URL for redirecting the payer to the payment page
qrcode_url string	URL to the png picture with the payment QR code
payer_email string	Payer's Email
payer_name string	Payer's name
payer_lang string	Payer's interface language
description string	Payment description
custom string	String for storing any value transmittable in webhook you can add any payment information required. May also be a json object
custom1 string	String for storing any simple value
custom2 string	String for storing any simple value
custom3 string	String for storing any simple value
custom4 string	String for storing any simple value
custom5 string	String for storing any simple value
custom6 string	String for storing any simple value
custom7 string	String for storing any simple value
custom8 string	String for storing any simple value
custom9 string	String for storing any simple value
webhook_url string	URL of the webhook. If no value shown, will be taken from the store settings
success_url string	URL for redirecting the payer to the successful payment page

Example:


```
{
  "dedicate_id": "f2a681c0-34e6-4668-8cb4-f76bac6b81c3",
```

```
  "dedicate_type": "USES",
  "store_id": "b6282b7f-f790-4acc-ab76-26aca5cf1f99",
  "account_id": "dadd1e36-33e1-4e6d-8e97-2d00ec9446bb",
  "dedicate_num": "164",
 "store_name": "Main",
  "is_test": false,
  "asset_id": "USD",
  "status": "COMPLETE",
  "status_name": "Complete",
  "crypto_network": "BITCOIN",
  "crypto_network_name": "Bitcoin",
  "address_id": "df08d662-7abd-40b3-9a96-c7eada9c334e",
  "address": "bc1qv2pqpdf3qq5q96djfgnlehn92a7e7n67wrcnqe",
  "address_alloc": "NEW",
  "start_date": "2023-06-02T14:06:25Z",
  "end_date": "2033-06-02T15:06:25Z",
  "payment_url": "https://payer.[service_name].com/dedicate/f2a681c0-34e6-4668-8cb4-f76bac6b81c3,
  "qrcode_url": "https://api.[service_name]/int/dedicate/f2a681c0-34e6-4668-8cb4-f76bac6b81c3/qrcode?oi=daee91ff-d4b7-4bb1-8c4a-855f57f9af6b&l=0",
  "payer_email": "name@email.com",
  "payer_name": "Customer name",
  "payer_lang": "en",
  "description": "The payment for item in store",
  "custom": "{
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "idempotence": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
  }",
  " custom1 ": " some value 1",
  " custom2 ": "some value 2",
 "webhook_url": "https://yourserver.com/[service_name]/result",
  "success_url": "https://yoursite.com/payment/success
}
```

Transfer amount

POST /transfer

Transfer move amount between stores.

Request:

to_store_id string	Store ID. Destination store
amount number	Amount to transfer
asset_id string	Asset ID. Asset to transfer
description string	Transfer description

Example:


```
{
  "to_store_id": "daee91ff-d4b7-4bb1-8c4a-855f57f9af6b",
  "amount": 30,
  "asset_id": "USDT.ERC20",
  "description": "api transfer"
}
```


Response

transfer_id string	Transfer ID in the service
transfer_num string	Transfer number in the system
from_store_id string	From Store ID in the service
from_store_name string	From Store name
from_account_id string	ID of the coin account in the service
to_store_id string	To Store ID in the service
to_store_name string	To Store name
to_account_id string	ID of the coin account in the service
is_test boolean	Test transfer (transfer to a test store)
amount number	Amount to transfer
asset_id string	Asset ID. Amount currency
status_type string	Transfer status. Values OPEN (open, in process), COMPLETE (completed successfully)
status_type_name string	Transfer status name
crypto_network string	Crypto network, e.g. BITCOIN, ETHEREUM….
crypto_network_name string	Crypto network name
start_date string	Transfer creation time
complete_date string	Transfer expiry time
description string	Payment description

Webhooks

Webhooks enable you to monitor the updates to transactions connected with your store. You can use webhooks to update your database entries for successful or cancelled payment. Crypto Payment Gateway will send webhook events upon the creation, completion or cancelation of a payment and upon any receipt from a transaction.

To accept webhooks, you need to subscribe your store in The Store -> Settings -> Webhooks in the field Webhook URL. You will be offered to enter the endpoint URL (https only) where you want to accept webhooks. You can choose whether to receive notices of all events or only those of interest to you. Prior to sending the webhook data Crypto Payment Gateway will check the security of the connection with your service therefore your server must be correctly set for supporting https. Your endpoint must respond with a status code 200 HTTP to confirm the acceptance of a webhook. If there is no receipt confirmation, we will repeat the attempt up to ten times. The maximum interval of such repeat attempts is one hour. Crypto Payment Gateway will sign each webhook event sent by it to your endpoints. The signature will be included in the title X-Webhook-HMAC-SHA256. Such title will contain a SHA256 HMAC signature of the un-processed request payload, computed with the use of your webhook secret as a key. Where the Webhook Secret was not filled in, no signature will be sent in the title.

Fields

All webhooks contain:

type	Type of webhook
event	Event of webhook: (payment.open, payment.complete, payment.cancel)
payment	Object of webhook. Contain Payment object, for details see **Create Payment API**

Example:


```
{
  “type”: “notification”,
  “event”: “payment.complete”,
  “payment”: {
    “payment_id”: “48bfcfc1-5658-48f9-81ca-d56ac92afd9d”,
    “store_id”: “b6282b7f-f790-4acc-ab76-26aca5cf1f99”,
    “account_id”: “943d119b-a65d-4412-b651-f4cc8d1f1c5f”,
    “store_name”: “Main”,
    “payment_num”: “162”,
    “payment_type”: “FIXED_AMOUNT”,
    “payment_type_name”: “Fixed Amount”,
    “is_test”: false,
    “amount”: “4”,
    “asset_id”: “USD”,
    “paid_amount”: “4”,
    “payment_amount”: “0.002137791379927765”,
    “payment_asset_id”: “ETH”,
    “rate”: 0.0005344478449819411,
    “rate_date”: “2023-05-31T13:38:10Z”,
    “ip_amount”: “0.002137791379927765”,
    “ip_asset_id”: “ETH”,
    “status”: “COMPLETE”,
    “status_name”: “Completed”,
    “address_id”: “db7d2924-dde5-4b63-b507-714b6a83a7c5”,
    “address”: “0x83e7BfD58357436FaF82d1C9fb86F94CFC39D0fD”,
    “confirms”: 10,
    “start_date”: “2023-05-31T13:38:59Z”,
    “end_date”: “2023-05-31T15:08:59Z”,
```

```
    "payment_url": "https://payer.[service_name].com/payment/48bfcfc1-5658-48f9-81ca-d56ac92afd9d?code=689b4a81-2a38-4fce-94c3-87b430d80d74",
    "qrcode_url": "https://api.[service_name].com/int/payment/48bfcfc1-5658-48f9-81ca-d56ac92afd9d/qrcode?code=689b4a81-2a38-4fce-94c3-87b430d80d74&oi=daee91ff-d4b7-4bb1-8c4a-855f57f9af6b&l=0",
    "payer_email": "1321@yopmail.com",
    "payer_name": "Test",
    «description»: «Платеж за сервис консультанта»,
    "custom": "{
      "client_id": "f54e57ab-b43c-480e-8d9d-cbd75764255c",
      "org_id": 3,
      "idempotence": "66ce9aa4-378b-4de1-a9b6-501b1f194883"
    }",
    " custom1 ": " some value 1",
    " custom2 ": "some value 2",
   "webhook_url": "https://yourserver.com/[service_name]/result",
   "success_url": "https://yoursite.com/payment/success",
   "cancel_url": "https://yoursite.com/payment/fail",
    "timeout": 0
  }
}
```


Example:


```
{
  "dedicate_id": "f2a681c0-34e6-4668-8cb4-f76bac6b81c3",
  "dedicate_type": "USES",
  "store_id": "b6282b7f-f790-4acc-ab76-26aca5cf1f99",
  "account_id": "dadd1e36-33e1-4e6d-8e97-2d00ec9446bb",
  "dedicate_num": "164",
 "store_name": "Main",
  "is_test": false,
  "asset_id": "USD",
  "status": "OPEN",
  "status_name": "Open",
  "crypto_network": "BITCOIN",
  "crypto_network_name": "Bitcoin",
  "address_id": "df08d662-7abd-40b3-9a96-c7eada9c334e",
  "address": "bc1qv2pqpdf3qq5q96djfgnlehn92a7e7n67wrcnqe",
  "address_alloc": "NEW",
 "start_date": "2023-06-02T14:06:25Z",
  "end_date": "2033-06-02T15:06:25Z",
  "payment_url": "https://payer.[service_name].com/dedicate/f2a681c0-34e6-4668-8cb4-f76bac6b81c3,
  "qrcode_url": "https://api.[service_name].com/int/dedicate/f2a681c0-34e6-4668-8cb4-f76bac6b81c3/qrcode?oi=daee91ff-d4b7-4bb1-8c4a-855f57f9af6b&l=0",
  "payer_email": "name@email.com",
  "payer_name": "Customer name",
  "payer_lang": "en",
  "description": "The payment for item in store",
  "custom": "{
    "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
    "org_id": 3,
    "idempotence": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
  }",
  " custom1 ": " some value 1",
  " custom2 ": "some value 2",
 "webhook_url": "https://yourserver.com/[service_name]/result",
  "success_url": "https://yoursite.com/payment/success
}
```

Example:


```
{
   "dedicate_id": "f2a681c0-34e6-4668-8cb4-f76bac6b81c3",
  "dedicate_type": "USES",
   "store_id": "b6282b7f-f790-4acc-ab76-26aca5cf1f99",
   "account_id": "dadd1e36-33e1-4e6d-8e97-2d00ec9446bb",
   "dedicate_num": "164",
  "store_name": "Main",
   "is_test": false,
   "asset_id": "USD",
   "status": "COMPLETE",
   "status_name": "Complete",
   "crypto_network": "BITCOIN",
   "crypto_network_name": "Bitcoin",
   "address_id": "df08d662-7abd-40b3-9a96-c7eada9c334e",
   "address": "bc1qv2pqpdf3qq5q96djfgnlehn92a7e7n67wrcnqe",
   "address_alloc": "NEW",
   "start_date": "2023-06-02T14:06:25Z",
   "end_date": "2033-06-02T15:06:25Z",
   "payment_url": "https://payer.[service_name].com/dedicate/f2a681c0-34e6-4668-8cb4-f76bac6b81c3,
   "qrcode_url": "https://api.[service_name].com/int/dedicate/f2a681c0-34e6-4668-8cb4-f76bac6b81c3/qrcode?oi=daee91ff-d4b7-4bb1-8c4a-
855f57f9af6b&l=0",
   "payer_email": "name@email.com",
   "payer_name": "Customer name",
   "payer_lang": "en",
   "description": "The payment for item in store",
   "custom": "{
      "client_id": "9a33c8b0-151c-481d-aef9-da15d883dc42",
      "org_id": 3,
      "idempotence": "341ba962-6ebf-4f3a-aef9-41c835a1dc26"
   }",
   " custom1 ": " some value 1",
   " custom2 ": "some value 2",
  "webhook_url": "https://yourserver.com/[service_name]/result",
   "success_url": "https://yoursite.com/payment/success
}
```