

DTC Wallet OpenAPI Documentation

Digital Treasures Center @ 2025

Update History

Date	Author	Comment
03 Nov 2021	Henry	
19 Jan 2022	Henry	Signature related texts update
25 Jan 2022	Kevin	Update OTC docs
25 Jan 2022	Kevin	Add MainNet Enum
25 Jan 2022	Henry	Default timezone
13 Jul 2022	Kevin	Add receipientAddressId request param
11 Aug 2022	Kevin	Add 'D-SubAccount-ID' request header
25 Aug 2022	Andy	Add Get Deposit Address API
22 Jan 2024	Kevin	Add getOtcInfoByReferenceNo API
28 Jan 2024	Kevin	Add Enum List: Available currency, Currency Precision 2. Update ErrorMsg & ErrorCode
29 Feb 2024	Kevin	Card Issuing v1
02 Apr 2024	Kevin	Add Webhook service
19 Apr 2024	Kevin	Add Fiat Deposit Account info API
24 Apr 2024	Kevin	Card Issuing v2
08 Oct 2024	Tim	Remit Info add fields: beneficiaryType, beneficiaryFirstName, beneficiaryLastName
27 Nov 2024	Kevin	Add FiatTransactionType: INVOICE_PAYMENT
28 Nov 2024	Kevin	Add Relationship Enum
29 Nov 2024	Kevin	Add Get Crypto Transaction fee v2
01 Jul 2025	Evan	Add Swap Order API

Date	Author	Comment
17 Sep 2025	Evan	Swap Order API Added support for SGQR QR code content query and payment
28 Sep 2025	Eric	Updated exception codes documentation, added detailed explanation on signatures, and included additional notes
03 Oct 2025	Eric	Structural and formatting updates across all Crypto API endpoints - Standardized section layout: each endpoint now includes .Request / .Response / .Error Response / .Request Body Sample / .Response Sample for consistency. - Added Error Response tables with detailed "Scenario" and "Message" descriptions. - Unified all parameter/response definitions into structured tables with columns: Type, Required, and Remark. Endpoint request parameters changes: □ Get Whitelist Address - Added new fields: level (trust level) and type (wallet address type). □ Get Crypto Transaction History - Added new filter parameters: state, type, senderAddressId, recipientAddressId, currency, and mainNet. □ Get Crypto Transaction Fee V2 - Reworked request parameter table with clearer field definitions (amount, currency) and updated examples. □ Crypto Withdraw - Added new constraint: referenceNo must be unique if provided.
10 Oct 2025	Summer	Enhanced explanations for OTC and Fiat module API docs
14 Oct 2025	Summer	Enhanced explanations for Basic info and OTC and Fiat module API docs
27 Oct 2025	Daniel	Add Wallet Account Transfer
29 Oct 2025	Zhi Yan	Added Search Wallet Balance Transaction History API
09 Nov 2025	Jensen	Amount change to String, webhook format
10 Nov 2025	Jensen	Add state field for crypto history search api, update crypto deposit sequence diagram

Date	Author	Comment
18 Nov 2025	Jensen	Update Crypto deposit and withdraw sequence diagram
25 Nov 2025	Zhi Yan	Update Get OTC Info
21 Jan 2026	Kang Sheng	Update open Currencies Api
26 Jan 2026	Kang Sheng	Update crypto Transaction Type, Activity Type, Fiat Transaction Type
05 Feb 2026	Kang Wei	Add institution limit error code and crypto webhook riskWithheldReason
09 Mar 2026	Kang Wei	Add senderAddress to cryptoTransaction webhook
09 Mar 2026	Kang Wei	Add clientId into /openapi/v1/otc/request
24 Mar 2026	Kang Wei	Add timestamp and signature fields to webhook payload examples
03 Apr 2026	Zhi Yan	Update Relationship enum
10 Apr 2026	Kang Wei	Amend BankType enum (add FPS, HK_FPS)
20 Apr 2026	Kang Wei	Add TransferPurpose enum
12 May 2026	Kang Wei	Sync update history
12 May 2026	Kang Wei	Fix 05 Feb 2026 entry position
26 May 2026	Kang Wei	Document fiatTransaction.recipientAmount on Fiat Withdraw (sender bears fee)

Table of Contents

1. Introduction.....	7
1.1. Overview.....	7
1.2. Specification Titles	7
1.3. Security Standards	7
2. Quick Start	9
2.1. Disclaimer.....	9
2.2. Before Integration	9
2.3. OAuth 2.0 Authentication	9
2.3.1. Usage	9
2.3.2. Fetch Access Token	9
2.3.3. Fetch Access Token by Refresh Token	10
2.4. Request Signature.....	11
2.4.1. Example: GET Request (Java).....	12
2.4.2. Example: POST Request (Java).....	13
3. Business Process.....	15
3.1. Otc	15
3.2. Fiat Deposit	15
3.3. Fiat Withdraw	17
3.4. Crypto Deposit.....	19
3.5. Crypto Withdraw	19
3.6. Swap Order	20
4. API	23
4.1. Basic Info	23
4.1.1. Get User Info	23
4.1.2. Get KYC Info	24
4.2. Wallet.....	26
4.2.1. Get Balance.....	26
4.2.2. Get Wallet Account Balance	27
4.2.3. Wallet Account Transfer	29
4.2.4. Search Balance History	31
4.3. OTC Trade.....	34
4.3.1. Get OTC Info	34
4.3.2. Get OTC History.....	36
4.3.3. Get OTC Info by Reference Number	38
4.3.4. Get OTC Rate.....	40
4.3.5. Request OTC	41
4.4. Crypto	44
4.4.1. Get Whitelist Address.....	44

4.4.2. Get Crypto Transaction History	46
4.4.3. Get Deposit Address	48
4.4.4. Get Crypto Transaction Fee V2	50
4.4.5. Crypto Withdraw	51
4.4.6. Get Crypto Transaction	53
4.4.7. Get Crypto Transaction By ReferenceNo	54
4.5. Fiat	56
4.5.1. Add Remit Info	56
4.5.2. Update Remit Info	59
4.5.3. Delete Remit Info	61
4.5.4. Get Fiat Deposit Account Info	62
4.5.5. Get Remit Info	64
4.5.6. Get Fiat Transaction	66
4.5.7. Get Fiat Transaction History	68
4.5.8. Get Upload Token	71
4.5.9. Upload File by Token	72
4.5.10. Get Fiat Transaction Fee	73
4.5.11. Fiat Withdraw	73
4.5.12. Fiat Deposit	76
4.5.13. Get Fiat Transaction by Reference Number	78
4.5.14. Open Currencies for Wallet Account	80
4.6. Webhook Service	80
4.7. Swap Order	84
4.7.1. Create Swap Order	84
4.7.2. Query Quote	87
4.7.3. Cancel Swap Order	89
4.7.4. Get Swap Order Details	90
4.7.5. QR Code Query	92
Appendix A: Data Structure	95
Appendix B: Error Structure	99
Appendix C: Enum List	101

Chapter 1. Introduction

1.1. Overview

The DTC Wallet OpenAPI, a JSON restful web service API, for third-party to integrate DTC Wallet solutions with their own business website or services.

This specification defines a set of interfaces and standard of security standards.

NOTE

The time-related values are based on the Singapore Time (SGT) Timezone by default.

1.2. Specification Titles

- **Name:** This column specifies the name of JSON field in request or response.
- **Max Length:** This column specifies the content and maximum length of value of JSON field.
 - **a** - Alpha
 - **n** - Numeric
 - **s** - Special printable character
- **Type:** This column specifies the type of JSON field.
 - **String** - the value is in string type;
 - **Nested JSON** - the value is a JSON object.
- **Required:** This column specifies the requirement of JSON field
 - **M** - Mandatory. DTC system will validate the value of this field in request
 - **C** - Conditional. In some conditions, this field is Mandatory. The condition will be specified in the description.
 - **O** - Optional. This field is optional, which will not affect the result of the transaction.
- **Description:** This column describes the usage, conditions or values contained of this JSON field.

1.3. Security Standards

The OpenAPI provides different level of security standards.

HTTPS: First level of security which is compulsory for all connections to DTC Wallet OpenAPI. TLS 1.2.

Whitelist: IP whitelist is optional, you can set it up in the API Key Management Page.

OAuth 2.0: Request session management.

Signature: For security checking.

Chapter 2. Quick Start

2.1. Disclaimer

Use of these APIs is subject to dtcpay's review and approval. dtcpay reserves the right, at its sole discretion, to approve, restrict, suspend, or revoke access at any time.

2.2. Before Integration

1. Request for account registration in DTC Wallet platform
2. Login with username and password issued by DTC team, go to the API Key Management Page.
3. Click + Create to open the dialog.
4. Key in the Name and IP Whitelist then submit.
5. The API Key, API Secret and Sign Key will be generated and pop-up in dialog. Be noted, These values are sensitive and present **one-time**, please keep them safely **before** close this dialog.

2.3. OAuth 2.0 Authentication

2.3.1. Usage

To use OAuth 2.0, please use Fetch Access Token first.

Add Authorization: Bearer {access_token} in the other API requests.

If the access token expired, you can re-fetch it or use the Fetch Access Token by Refresh Token API. Or you can get a new access token before it expired.

NOTE

Only one access token can be exists in the same time per client-id, we suggest cache the access token in your caching system, for example, Redis.

2.3.2. Fetch Access Token

Request sample

```
curl -X "POST" "https://{host}/openapi/auth/oauth2/token" \  
  -H 'Content-Type: multipart/form-data; charset=utf-8;  
boundary=__X_PAW_BOUNDARY__' \ ①  
  -F "client_id=xxx" \ ②  
  -F "client_secret=xxx" \ ③  
  -F "grant_type=client_credentials" ④
```

① The Content-Type should be multipart/form-data.

② The API Key value fetched from Before Integration.

③ The API Secret value fetched from Before Integration.

④ Fixed value client_credentials.

NOTE

The client_id and client_secret can be obtained via the web portal at <https://business.dtcpay.com/login>. Navigate to the **API** menu and create a **Swap API** to generate the credentials.

Response sample (formatted)

```
{
  "access_token": "CkppLkyiqEUKITGdtCZRUB1", ①
  "expires_in": 21600, ②
  "refresh_token": "CpCAhcCRtLU4kPjs54omWW3A", ③
  "rt_expires_in": 43200, ④
  "token_type": "bearer" ⑤
}
```

① For API calls, keep safely in your system. Any leakage will cause potential security risk. Please contact DTC if you feel it's compromised.

② The Access Token expiry time, in seconds.

③ To fetch a new access token before these two tokens are expired.

④ The Refresh Token will expiry, in seconds.

⑤ Fixed value bearer.

NOTE

When Access Denied (errCode:00006) occurs, it may be caused by a client_id/client_secret mismatch. It can also happen if the IP address configured during API key creation does not match the requesting IP address (the source IP of the client calling the API).

2.3.3. Fetch Access Token by Refresh Token

Request sample

```
curl -X "POST" "https://{host}/openapi/auth/oauth2/token" \
  -H 'Content-Type: multipart/form-data; charset=utf-8;
  boundary=__X_PAW_BOUNDARY__' \ ①
  -F "refresh_token=CpCAhcCRtLU4kPjs54omWW3A" \ ②
  -F "grant_type=refresh_token" ③
```

① The Content-Type should be multipart/form-data.

② The refresh_token value fetched from the previous step.

③ Fixed value refresh_token.

Response sample (formatted)

```
{
  "access_token": "CE2ZgMjcgcf0vzY0jcS7csYe", ①
  "expires_in": 21600,
```



```

    "refresh_token": "CdugTgyDDa05uURhK6n6GYiW", ②
    "rt_expires_in": 43200,
    "token_type": "bearer"
}

```

- ① A new Access Token will be generated.
- ② A new Refresh Token will be generated.

2.4. Request Signature

For key function API requests, the signature are mandatory in the HTTP Header for security reason. You can apply to all the API requests as well.

1. Components of signature before hash:

- HTTP Method: In uppercase.
- Timestamp: Timestamp number in millis, **Singapore timezone**. The request will be failed if the timestamp is before or after **2 minutes**.
- URL: The request URL without scheme and hostname.
- Querystring: The in-URL parameters without the prefix ?, un-encoded.
- Request Body: For string-based body, empty if not set. Must **exactly same** as the actually request body, including the invisible characters.

2. You need to combine these values by sequence, for example:

```

GET1636360576641/openapi/test?test=234
POST1636360661729/openapi/test{"a":"124"}

```

3. Generate the signature string

After you get the string to sign in the 2nd step, then you can generate the signature string by using **HMAC-SHA512** method and **Base64** encoding with the Sign Key from Before Integration.

For example:

- Sign Key: aR2822Y6XbehWMcInB0Y2NJK
- String to sign: POST1636360661729/openapi/test{"a":"124"}

Then the signature string should be:
MxrYnCm9Q7J0Av0rISf8+T2kuTW1d/w0at8aaPaoiX08VWfun3XPokVlIx1TkHXdcitls09wzfUGtXQZq23xdg==

4. Append signature to HTTP headers

Beside the Authorization header every request needed, the following headers are also need to add when perform a request:

- D-TIMESTAMP: The timestamp value should be same as Timestamp component in signature.

- D-SIGNATURE: The signature string generated.
- D-SUB-ACCOUNT-ID : The id of sub account, registered under the master account
- DTC-MFA-Data : Verification codes or security data used for validation

NOTE

The body used for signature calculation is read by the server directly from the raw HTTP request (HttpServletRequest, via IOUtils.toByteArray(request.getReader(), UTF-8)). Therefore, the body in the signature must be **exactly the same** as the request body actually sent by the client, including spaces, line breaks, tabs, and any invisible characters. Even the slightest difference (e.g., an extra space) will cause signature verification to fail. The correct practice is to use the exact same raw request body string for both sending the request and generating the signature.

2.4.1. Example: GET Request (Java)

```
import com.google.common.hash.Hashing;

import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class SignatureTestGuava {

    public static void main(String[] args) {
        // Shared secret key (Sign Key) provided during integration
        String signKey = "4mt0MUec0tOEDI42WjN4BmM0";

        // Example request details
        String method = "GET";
        String uri = "/openapi/v1/user";
        String query = "";
        String body = "";
        String timestamp = String.valueOf(System.currentTimeMillis());

        // Step 1: Build the stringToSign (method + timestamp + uri + query + body)
        String stringToSign = method + timestamp + uri;
        if (!query.isEmpty()) {
            stringToSign += "?" + query;
        }
        stringToSign += body;

        // Step 2: Generate HMAC-SHA512 signature with Sign Key, then Base64 encode it
        String signature = Base64.getEncoder().encodeToString(
            Hashing.hmacSha512(signKey.getBytes(StandardCharsets.UTF_8))
                .hashBytes(stringToSign.getBytes(StandardCharsets.UTF_8))
                .asBytes()
        );

        System.out.println("Timestamp    : " + timestamp);
        System.out.println("StringToSign: " + stringToSign);
    }
}
```

```

        System.out.println("Signature    : " + signature);
    }
}

```

2.4.2. Example: POST Request (Java)

```

import com.google.common.hash.Hashing;

import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class SignatureTestGuavaPOST {
    public static void main(String[] args) {
        // Shared secret key (Sign Key)
        String signKey = "4mt0MUec0tOEDI42WjN4BmM0";

        // Example request details
        String method = "POST";
        String uri = "/openapi/v1/otc/get-otc-rate";
        String query = "";
        String body = "{\"query\":{\"buyCurrency\":\"USD\",\"sellCurrency\":\"USDT\"}}";

        // Current timestamp in milliseconds
        String timestamp = String.valueOf(System.currentTimeMillis());

        // Step 1: Build the stringToSign (method + timestamp + uri + query + body)
        String stringToSign = method + timestamp + uri;
        if (!query.isEmpty()) {
            stringToSign += "?" + query;
        }
        stringToSign += body;

        // Step 2: Generate HMAC-SHA512 signature with Sign Key, then Base64 encode it
        String signature = Base64.getEncoder().encodeToString(
            Hashing.hmacSha512(signKey.getBytes(StandardCharsets.UTF_8))
                .hashBytes(stringToSign.getBytes(StandardCharsets.UTF_8))
                .asBytes()
        );

        System.out.println("Timestamp    : " + timestamp); // Value for D-TIMESTAMP
        System.out.println("StringToSign : " + stringToSign); // Raw string used for
        signing
        System.out.println("Signature    : " + signature); // Value for D-SIGNATURE

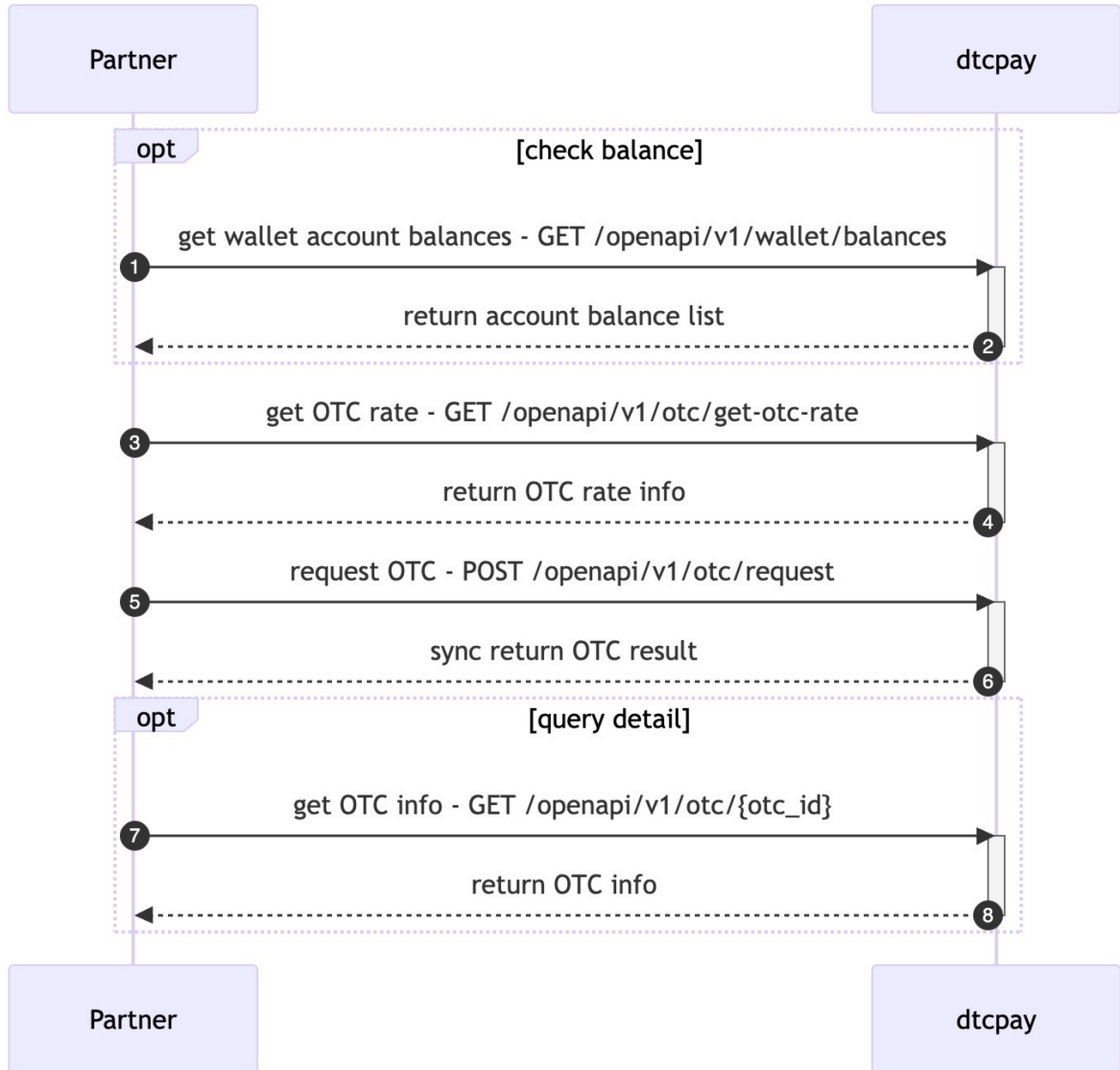
    }
}

```


Chapter 3. Business Process

3.1. Otc

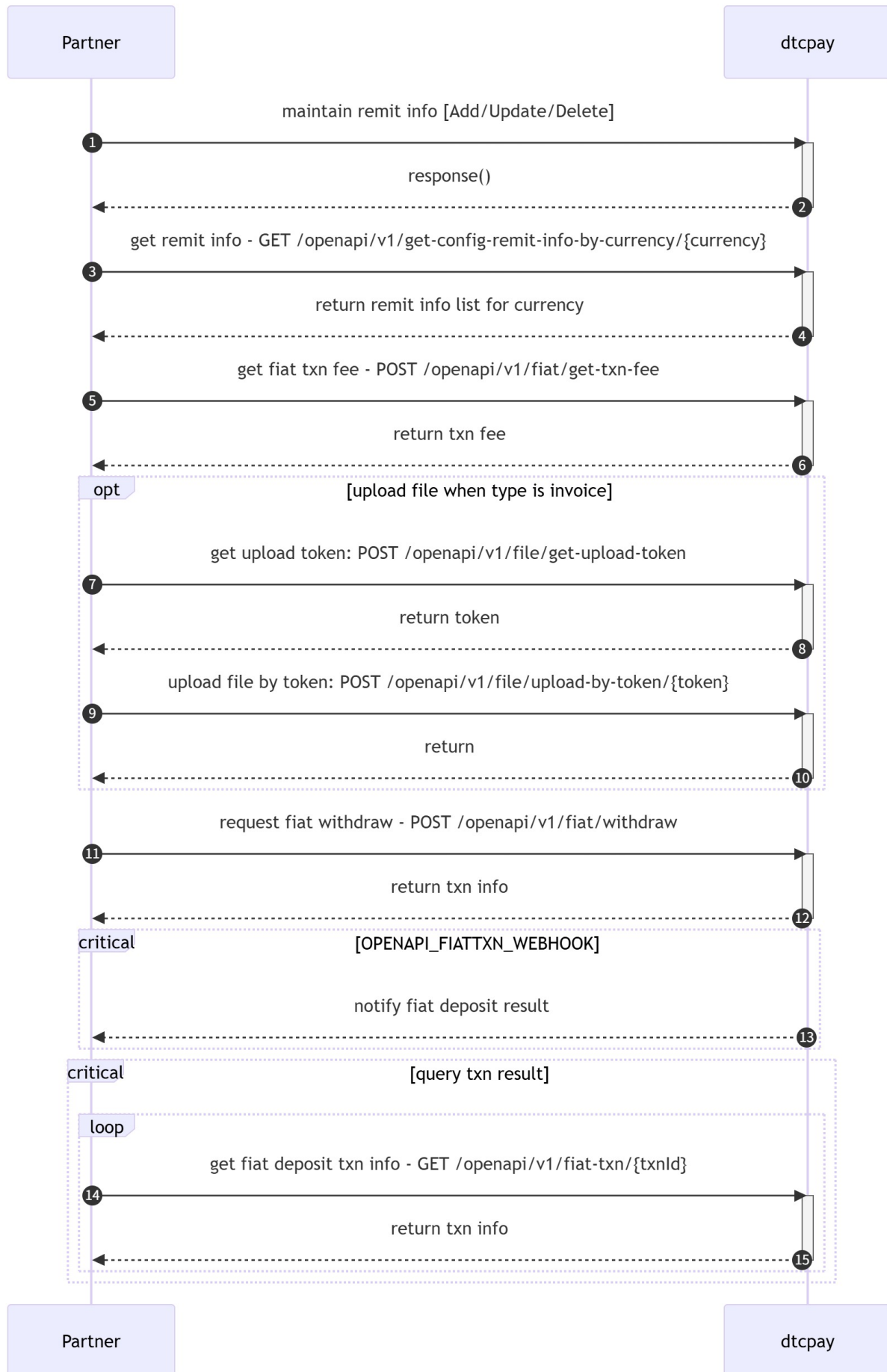
This use case describes the process of initiating OTC trading



3.2. Fiat Deposit

This use case describes the process of recharging a Fiat account.

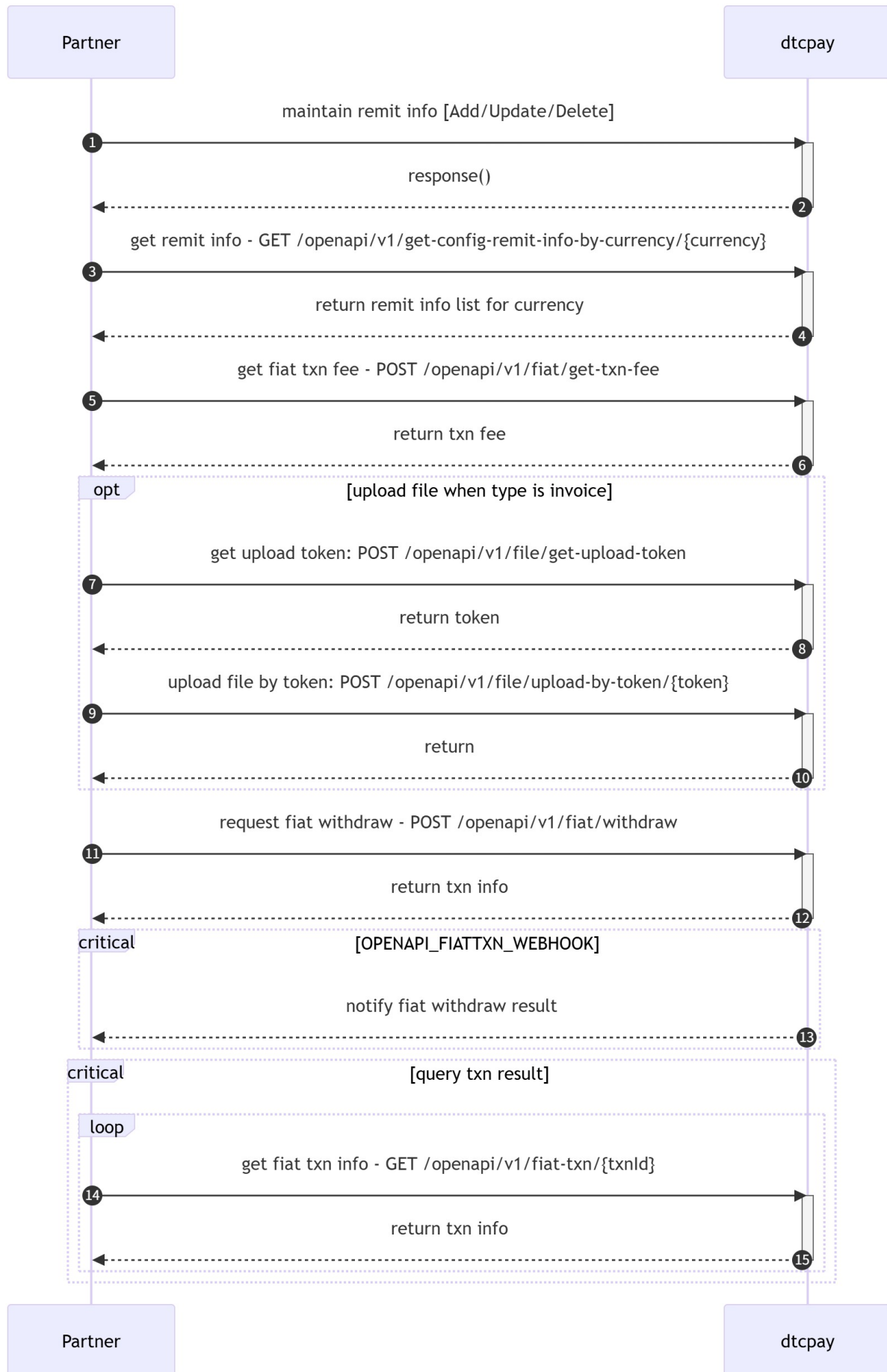
The partner obtains the corresponding currency's dtcpay receiving account information through the interface and submits a recharge form. The partner's financial personnel complete the transfer of the corresponding amount offline. After checking and confirming that the receiving account has been credited, the dtcpay finance will complete the recharge of the partner's corresponding wallet account.



3.3. Fiat Withdraw

This use case describes the process of withdrawing funds from fiat account.

It is necessary to first maintain the Remit payee account information, and then initiate a Fiat withdrawal request. The dtcpay operator will process the payment, usually 1-3 work days it can be completed.



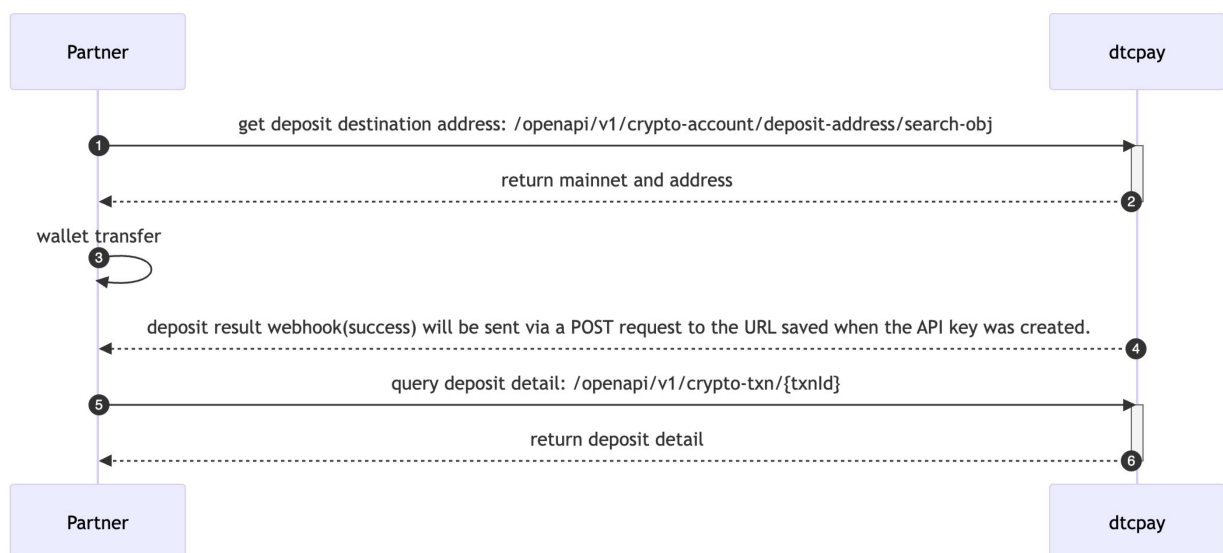
3.4. Crypto Deposit

Crypto's business includes whitelisting, deposit, and withdrawal.

We usually operate on web pages to add or disable whitelists. If API access is required, we will provide a separate API for reference after regulatory approval.

When recharging with Crypto, it involves senderAddress and destinationAddress, and users need to add senderAddress to the whitelist and enable it. The destinationAddress is automatically assigned to the user by DTC as the recharge destination address (calling the [POST]/openapi/v1/crypto-account/deposit-address/search-obj api will return the deposit recipient address, please refer to the API documentation for details).

If the user does not add the senderAddress to the whitelist, the transaction will be set as a risky transaction (status=102 (Risk Withheld)). At this point, the user can continue to add to the whitelist and enable it. After add whitelist successful, the transaction will be automatically changed to success.

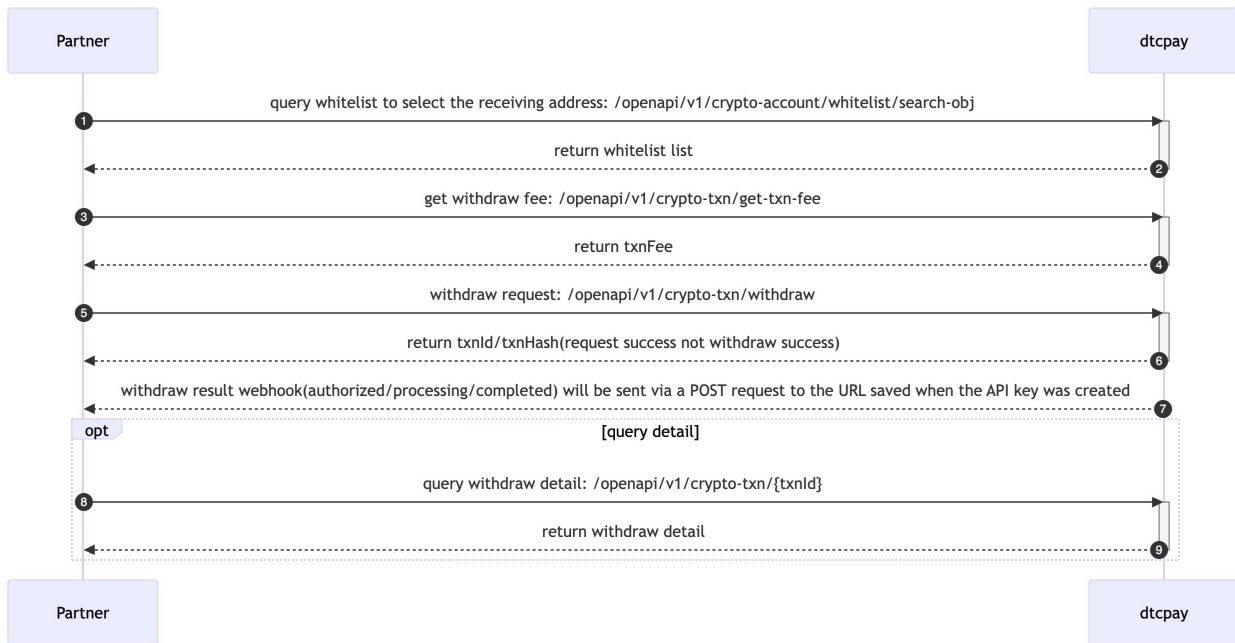


3.5. Crypto Withdraw

At present, users need to select an address from the whitelist to receive payments.

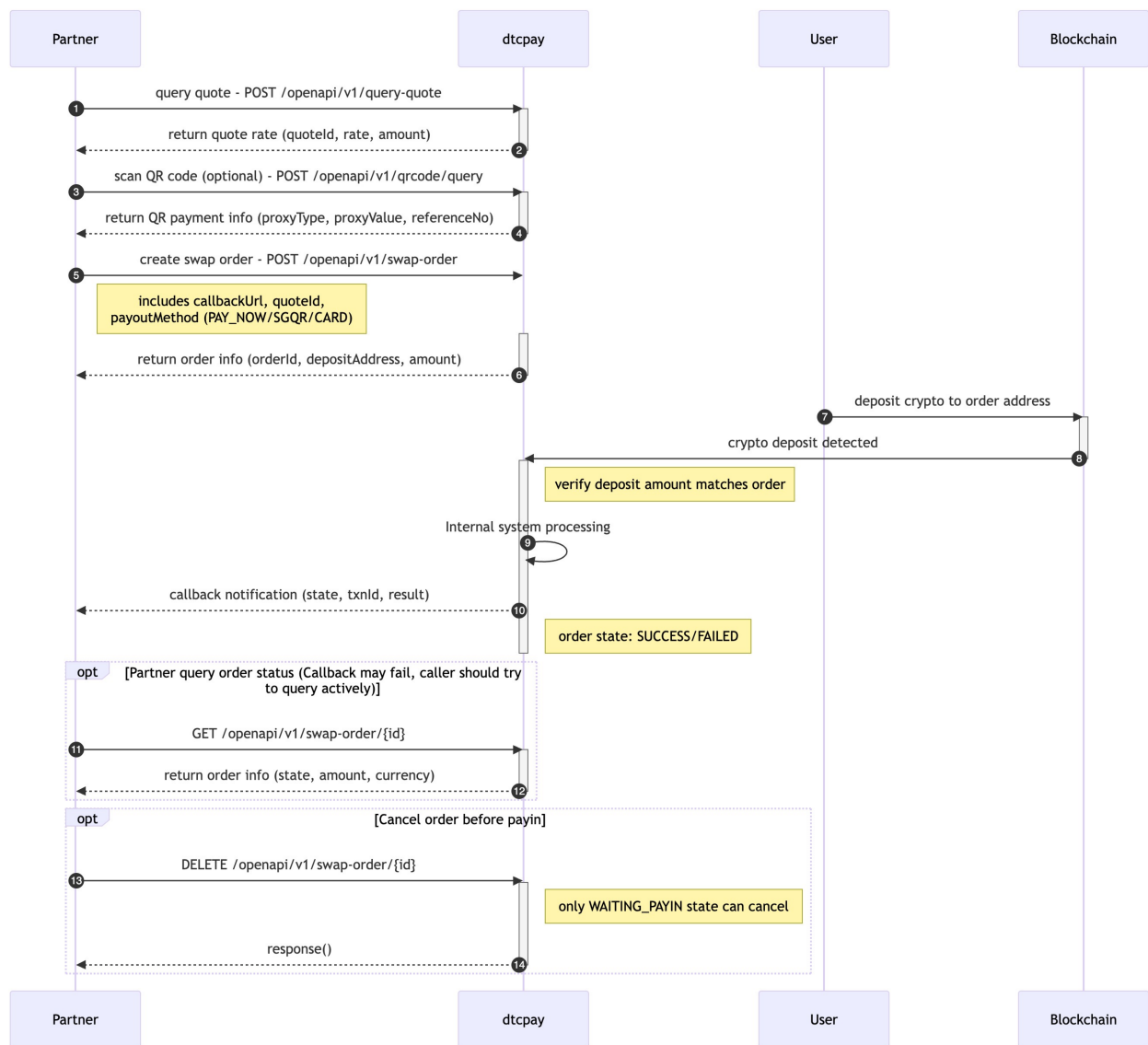
When making a withdrawal request, it only represents that the withdrawal request is successful.

After the actual processing on the chain is completed, dtc will conduct a webhook, and users can also choose to query the results based on the returned ID or referenceNo.



3.6. Swap Order

This use case describes the process of using the swap order interface in a scan code payment scenario.



Chapter 4. API

4.1. Basic Info

4.1.1. Get User Info

Endpoint

[GET] /openapi/v1/user

Description

Retrieve the authenticated partner's own company (non-individual) KYB information.

This endpoint returns company information for the authenticated client if it is registered as a non-individual (corporate) partner. Individual clients will receive a “fetch data failed” response.

Response

Name	Length	Type	Required	Description	Remark
nonIndividual		Object	M	Company information object	
nonIndividual.id	-	Integer	M	Company ID	
nonIndividual.fullName	63	String	M	Full company name	
nonIndividual.shortName	31	String	O	Short company name	
nonIndividual.address	255	String	O	Company address	
nonIndividual.city	255	String	O	City	
nonIndividual.state	255	String	O	State/Province	
nonIndividual.country	3	String	O	Country code	ISO 3166-1 alpha-3 code
nonIndividual.postalCode	255	String	O	Postal code	
nonIndividual.mccId	-	Integer	O	Merchant category code	
nonIndividual.status	-	Integer	O	Company status	refer to Enum List [ClientStatus]
nonIndividual.type	-	Integer	O	Company type	
nonIndividual.createdAt	19	String	O	Record creation timestamp	Format: yyyy-MM-dd HH:mm:ss

Name	Length	Type	Required	Description	Remark
nonIndividual.lastUpdatedDate	19	String	O	Last update timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00001	Failed to fetch data

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "nonIndividual": {
      "address": "7 Straits View, Marina One East Tower",
      "city": "Singapore",
      "country": "SGP",
      "createdDate": "2020-11-10 11:54:17",
      "fullName": "Some Company Pte Ltd. (Test)",
      "id": 8520110007,
      "lastUpdatedDate": "2021-07-02 15:31:16",
      "mccId": 791,
      "postalCode": "018936",
      "shortName": "SomeCom",
      "state": "Central Singapore",
      "status": 99
    }
  }
}
```

4.1.2. Get KYC Info

Endpoint

```
[GET] /openapi/v1/profile/kyc-info
```

Description

Retrieve the authenticated partner's KYB (Know Your Business) verification details, including company KYC records and registered address information. This endpoint returns KYC information only for non-individual (corporate) clients. Individual clients will receive a "fetch data failed" response.

Response

Name	Length	Type	Required	Description	Remark
geoAddress		Object	M	Geographic address information	
geoAddress.id	-	Integer	M	Address ID	
geoAddress.address1	255	String	O	Address line 1	
geoAddress.address2	255	String	O	Address line 2	
geoAddress.city	63	String	O	City	
geoAddress.state	63	String	O	State/Province	
geoAddress.country	3	String	O	Country code	ISO 3166-1 alpha-3 code
geoAddress.postal	15	String	O	Postal code	
geoAddress.type		String	O	Address type	Refer to Enum List - GeoAddressType
kycNonIndividual		Object	M	KYC information for non-individual entity	
kycNonIndividual.id	-	Integer	M	KYC record ID	
kycNonIndividual.registerName	127	String	M	Registered company name	
kycNonIndividual.registerNumber	127	String	M	Company registration number	
kycNonIndividual.addressId	-	Integer	M	Reference to geoAddress.id	
kycNonIndividual.contactPerson	127	String	O	Contact person name	
kycNonIndividual.contactNumber	31	String	O	Contact phone number	
kycNonIndividual.email	255	String	O	Contact email	
kycNonIndividual.businessDescription	255	String	O	Business description	
kycNonIndividual.sourceOfFund	127	String	O	Source of funds	

Name	Length	Type	Required	Description	Remark
kycNonIndividual.purpose	255	String	O	Business purpose	
kycNonIndividual.lastUpdatedDate	19	String	O	Last update timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00001	Failed to fetch data

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "geoAddress": {
      "address1": "7 Straits View, Marina One East Tower TEST",
      "address2": "05-01 Suite S5012",
      "city": "Singapore",
      "country": "SGP",
      "id": 4,
      "postal": "018936",
      "state": "Central Singapore"
    },
    "kycNonIndividual": {
      "addressId": 4,
      "businessDescription": "consultant",
      "contactNumber": "+65 88888888",
      "contactPerson": "John Doe",
      "email": "john.doe@some.com",
      "id": 8520110007,
      "lastUpdatedDate": "2021-10-12 16:51:41",
      "registerName": "Some Company Pte Ltd. (Test)",
      "registerNumber": "123143e1",
      "sourceOfFund": "Business"
    }
  }
}
```

4.2. Wallet

4.2.1. Get Balance

Endpoint

[GET] /openapi/v1/wallet/balance/{currency}

Description

Retrieve the balance for a specific currency wallet.

Path Variables

No	Name	Type	Required	Description
1	currency	String	M	Wallet balance currency (e.g., USD, SGD, BTC, ETH, USDT). Refer to Enum List - Available Currency

Response

Name	Length	Type	Required	Description	Remark
balance	-	String	M	Current wallet balance for the specified currency	

Possible Error Codes

Code	Description
10001	Wallet account not exist
00006	Access denied

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "balance": "10"
  }
}
```

4.2.2. Get Wallet Account Balance

Endpoint

[GET] /openapi/v1/wallet/balances

Description

Retrieve all wallet account balances for the authenticated client across all currencies.

Response (Array of wallet accounts in resultList)

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Wallet account ID	
clientId	-	Integer	M	Client ID	
status	-	Integer	M	Wallet account status	Refer to Enum List - WalletAccountStatus
currency	7	String	M	Wallet currency	e.g., USD, SGD, BTC, ETH, USDT
balance	-	String	M	Current balance	
createdDate	19	String	M	Wallet creation timestamp	Format: yyyy-MM-dd HH:mm:ss
lastUpdatedDate	19	String	M	Last update timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
11999	Get wallet balance failed

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "resultList": [
    {
      "id": 988,
      "clientId": 8520111007,
      "status": 1,
      "currency": "HKD",
      "balance": "70526.27",
      "createdDate": "2022-12-07 16:27:52",
      "lastUpdatedDate": "2024-01-12 00:08:00"
    },
    {
      "id": 987,
      "clientId": 8520111007,
      "status": 1,
      "currency": "EUR",
      "balance": "544.39",
      "createdDate": "2022-12-07 16:27:52",
      "lastUpdatedDate": "2024-01-12 00:08:00"
    }
  ]
}
```

```
}
```

4.2.3. Wallet Account Transfer

Endpoint

```
[POST] /openapi/v1/wallet/transfer
```

Description

Used when users transfer money to another dtcpay internal customers, include P2P,P2B,B2B,B2P.

Request

Name	Length	Type	Required	Description	Remark
currency	-	String	M	Transfer currency	
amount	-	String	M	Transfer amount	Amount after deduct fee
recipientClientId	-	Integer	M	Recipient client id	e.g., Subaccount clientId
note	35	String	O	Message to beneficiary	e.g., Test txn

Request Sample

- Fiat transfer request sample

```
{
  "query": {
    "currency": "SGD",
    "amount": "1",
    "recipientClientId": 85250630150001070,
    "note": "test1"
  }
}
```

- Stable coin transfer request sample

```
{
  "query": {
    "currency": "USDT",
    "amount": "1",
    "recipientClientId": 85250630150001070,
    "note": "test1"
  }
}
```

Response

Name	Length	Type	Required	Description	Remark
fiatTransaction	-	Object	O	Returned when transferring fiat currency	Refer to FiatTransaction
cryptoTransaction	-	Object	O	Returned when transferring Stable Coin currency	Refer to CryptoTransaction

Possible Error Codes

Code	Description
00011	Balance is insufficient.
20999	Account abnormality.
21999	Wallet transfer request failed.

Response Sample (formatted)

- Fiat transfer response

```
{
  "header": {
    "success": true
  },
  "result": {
    "fiatTransaction": {
      "id": 2510202055340218000,
      "type": 6,
      "state": 200,
      "clientId": 250507120005941,
      "senderAccountId": 250507120005941,
      "recipientAccountId": 85250630150001070,
      "recipientName": "KngCompany",
      "currency": "SGD",
      "recipientCurrency": "SGD",
      "feeCurrency": "SGD",
      "amount": "1",
      "recipientAmount": "1",
      "transactionFee": "0",
      "operator": "[250507120005941] test1",
      "approver": "[250507120005941] test1",
      "note": "test",
      "createdTime": "2025-10-20 20:55:35",
      "lastUpdatedDate": "2025-10-20 20:55:35"
    }
  }
}
```

- Stable coin transfer response

```

{
  "header": {
    "success": true
  },
  "result": {
    "cryptoTransaction": {
      "id": 2510202054090305000,
      "type": 7,
      "state": 200,
      "clientId": 250507120005941,
      "amount": "1",
      "currency": "USDT",
      "gasFee": "0",
      "transactionFee": "0",
      "recipientAddressId": 85250630150001070,
      "senderAddressId": 250507120005941,
      "remark": "test",
      "operator": "[250507120005941] test1",
      "approver": "[250507120005941] test1",
      "requestTimestamp": "2025-10-20 20:54:10",
      "lastUpdatedDate": "2025-10-20 20:54:10"
    }
  }
}

```

4.2.4. Search Balance History

Endpoint

[GET] /openapi/v1/wallet/balance/history/search

Description

Retrieve the client's wallet transaction history.

Request

Name	Type	Length	Required	Description	Remark
currency	Enum/String	7	O	Currency of the transaction	Refer to Enum List – Currency
type	Enum	1	O	Transaction type	Refer to Enum List – ActivityType
yearMonth	String	6	O	Year and month for filtering	Format: yyyyMM
createTimestamp	LocalDateTime	—	O	Start time for filtering	Format: yyyy-MM-dd HH:mm:ss

Name	Type	Length	Required	Description	Remark
createTimeEnd	LocalDateTime	—	O	End time for filtering	Format: yyyy-MM-dd HH:mm:ss
pending	Boolean	—	O	Whether to search for pending transactions	—
pendingTxnIdList	List<Long>	—	O	List of pending transaction IDs to exclude	—
size	4	Long	M	Transactions per page	Maximum 500
current	8	Long	M	Number of current pages	

Response

Name	Length	Type	Required	Description	Remark
activityType	-	Enum	M	Activity type	Refer to Enum List - ActivityType
detail	255	String	O	Description or transaction remark	
mainAmount	-	String	M	Main transaction amount	
additionalAmount	-	String	O	Additional amount (e.g., fee or adjustment)	
mainCurrency	7	String	M	Main currency code	
additionalCurrency	7	String	O	Additional currency code	
totalChargedAmount	-	String	O	Total amount charged in the transaction	
totalChargedCurrency	7	String	M	Currency code of the charged amount	
time	19	String	M	Transaction time	Format: yyyy-MM-dd HH:mm:ss
relatedId	-	Long	M	Related record ID or reference	
paymentMethod	-	String	O	Payment method used to top up wallet or purchase crypto	

Name	Length	Type	Required	Description	Remark
title	-	String	O	Card transaction sub type or wallet balance history type	
productCode	-	String	O	Product code of the dtc card	
otcRate	-	String	O	Otc rate	
rate	-	String	O	Otc rate or rate from third party	
state	-	Integer	M	Different types may have the same value	Refer to Enum List - FiatTransactionState, CryptoTransactionState, PaymentTransactionState, OtcStatus, CommissionStatus, OrderStatus, AdjustmentStatus
network	-	String	O	Main Net of the wallet address	
isAdjustmentIncrease	-	Boolean	O	Whether adjustment is an increase	
brand	-	Enum	O	Associated brand	
size	4	Long	M	Transactions per page	Maximum 500
current	8	Long	M	Number of current pages	
total		Long	M	Total Transactions number	

Possible Error Codes

Code	Description
00004	Unknown API error
00006	Access denied

Request Body Sample (formatted)

```
{
  "query": {
  },
  "page": {
    "current": 1,
    "size": 10
  }
}
```

```
}  
}
```

Response Sample (formatted)

```
{  
  "header": {  
    "success": true  
  },  
  "resultList": [  
    {  
      "relatedId": 2509231015230507618,  
      "detail": "****0001",  
      "maAmount": "2222",  
      "maCurrency": "USDT",  
      "time": "2025-09-23 10:15:24",  
      "activityType": 15,  
      "state": 200,  
      "productCode": "4202",  
      "brand": 1  
    },  
    {  
      "relatedId": 2509231014530507607,  
      "detail": "****0001",  
      "maAmount": "2",  
      "maCurrency": "USDT",  
      "time": "2025-09-23 10:14:56",  
      "activityType": 15,  
      "state": 200,  
      "productCode": "4202",  
      "brand": 1  
    }  
  ],  
  "pagination": {  
    "current": 1,  
    "size": 10,  
    "total": 1980  
  }  
}
```

4.3. OTC Trade

4.3.1. Get OTC Info

Request

```
[GET] /openapi/v1/otc/{otcId}
```


Path Variables

No	Name	Type	Required	Description
1	otcId	Long	Yes	OTC ID (unique ID from DTC)

Response

Name	Length	Type	Required	Description	Remark
id	-	Long	M	OTC ID	
status	-	Integer	M	OTC status	Refer to Enum List - OtcStatus
clientId	-	Long	M	Client ID	
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Currency
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Currency
rate	16	String	M	Exchange rate	
dtcQuoteId	-	Long	M	DTC quote ID	
sellAmount	16	String	M	Amount to sell	
buyAmount	16	String	M	Amount to buy	
operator	63	String	M	Operator	
referenceNo	63	String	M	Reference number	
approver	63	String	M	Approver	
completedTime	18	String	M	Completed time	Format: yyyy-MM-dd HH:mm:ss
lastUpdatedDate	18	String	M	Last updated time	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00999	Swap Info not exist
00006	Access denied

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 2510101421380624067,
    "status": 5,
    "clientId": 8524010424,
    "sellCurrency": "GBP",
    "buyCurrency": "USD",
    "rate": "1.345119262388",
    "dtcQuoteId": 1155389391925733471,
    "sellAmount": "0.75",
```

```

    "buyAmount": "1",
    "operator": "[8524010424] openapi",
    "referenceNo": "173561238951",
    "approver": "[8524010424] openapi",
    "completedTime": "2025-10-10 14:21:39",
    "lastUpdatedDate": "2025-10-10 14:21:39"
  }
}

```

4.3.2. Get OTC History

[POST] /openapi/v1/otc/search

Request

Name	Length	Type	Required	Description	Remark
id	-	Long	O	OTC ID (when provided, all other parameters will be ignored)	
status	-	Integer	O	OTC status	
sellCurrency	3	String	O	Currency to sell	Refer to Enum List - Currency
buyCurrency	3	String	O	Currency to buy	Refer to Enum List - Currency
orderTimeStart	9	String	O	Order start time	Format: yyyy-MM-dd
orderTimeEnd	9	String	O	Order end time	Format: yyyy-MM-dd
page.current	-	Integer	O	Current page number	
page.size	-	Integer	O	Page size	

Response

Name	Length	Type	Required	Description	Remark
id	-	Long	M	OTC ID	
status	-	Integer	M	OTC status	Refer to Enum List - OtcStatus
clientId	-	Long	M	Client ID	
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Currency
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Currency

Name	Length	Type	Required	Description	Remark
rate	16	String	M	Exchange rate	
sellAmount	16	String	M	Amount to sell	
buyAmount	16	String	M	Amount to buy	
operator	63	String	M	Operator	
completedTime	18	String	M	Completed time	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
11999	Invalid OTC Id

Request Body Sample (formatted)

```
{
  "query": {
    "id": 1,
    "status": 5,
    "buyCurrency": "USDT",
    "sellCurrency": "USD",
    "orderTimeStart": "YYYY-MM-DD",
    "orderTimeEnd": "YYYY-MM-DD"
  },
  "page": {
    "current": 1,
    "size": 50
  }
}
```

Response Sample (formatted)

```
{
  "resultList": [
    {
      "id": 2508071556490618367,
      "status": 5,
      "clientId": 8524010424,
      "sellCurrency": "SGD",
      "buyCurrency": "USDT",
      "rate": "0.7702672119830299368",
      "sellAmount": "123123",
      "buyAmount": "94837.6",
      "operator": "downey.wang@dtc.top",
      "completedTime": "2025-08-07 15:56:50"
    },
    {
      "id": 240605852,
      "status": 5,
```

```

        "clientId": 8524010424,
        "sellCurrency": "SGD",
        "buyCurrency": "USDT",
        "rate": "0.737340457162302",
        "sellAmount": "10000",
        "buyAmount": "7373.4",
        "operator": "limit@003.com",
        "completedTime": "2024-06-05 17:42:49"
    },
    {
        "id": 240124012,
        "status": 5,
        "clientId": 8524010424,
        "sellCurrency": "SGD",
        "buyCurrency": "USDT",
        "rate": "0.71037",
        "sellAmount": "10",
        "buyAmount": "7.1",
        "operator": "k.tfqbishe@ykckjkg.net.cn",
        "completedTime": "2024-01-24 18:13:44"
    }
],
"header": {
    "success": true
},
"pagination": {
    "current": 1,
    "size": 50,
    "total": 3
}
}

```

① resultList array contents can be multiple, single, or empty.

4.3.3. Get OTC Info by Reference Number

Request

```
[GET] /openapi/v1/otc/reference-number/{referenceNo}
```

Path Variables

No	Name	Type	Required	Description
1	referenceNo	String	Yes	Reference number (unique ID from your system)

Response

Name	Length	Type	Required	Description	Remark
id	-	Long	M	OTC ID	

Name	Length	Type	Required	Description	Remark
status	-	Integer	M	OTC status	Refer to Enum List - OtcStatus
clientId	-	Long	M	Client ID	
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Currency
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Currency
rate	16	String	M	Exchange rate	
dtcQuoteId	-	Long	M	DTC quote ID	
sellAmount	16	String	M	Amount to sell	
buyAmount	16	String	M	Amount to buy	
operator	63	String	M	Operator	
referenceNo	63	String	M	Reference number	
approver	63	String	M	Approver	
completedTime	18	String	M	Completed time	Format: yyyy-MM-dd HH:mm:ss
lastUpdatedDate	18	String	M	Last updated time	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00006	Access denied
00999	ReferenceNo not exist
00001	Failed to fetch data

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 2510101421380624067,
    "status": 5,
    "clientId": 8524010424,
    "sellCurrency": "GBP",
    "buyCurrency": "USD",
    "rate": "1.345119262388",
    "dtcQuoteId": 1155389391925733471,
    "sellAmount": "0.75",
    "buyAmount": "1",
    "operator": "[8524010424] openapi",
    "referenceNo": "173561238951",
    "approver": "[8524010424] openapi",
    "completedTime": "2025-10-10 14:21:39",
    "lastUpdatedDate": "2025-10-10 14:21:39"
  }
}
```

```
}  
}
```

4.3.4. Get OTC Rate

[POST] /openapi/v1/otc/get-otc-rate

Request

Name	Length	Type	Required	Description	Remark
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Available Currency
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Available Currency
sellAmount	3	String	O	Sell amount	
buyAmount	3	String	O	Buy amount	

NOTE

- At least one of sellAmount or buyAmount must be provided.
- If both are omitted, the system will only return indicative rate information.

Response

Name	Length	Type	Required	Description	Remark
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Available Currency
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Available Currency
clientId	128	Long	M	Client ID	
rateExpiryTime	18	String	M	Rate expiry time	Format: yyyy-MM-dd HH:mm:ss
rate	16	String	M	Exchange rate	
quoteId	19	String	M	DTC quote ID	
sellWalletBalance	16	String	M	Sell wallet balance	
buyWalletBalance	16	String	M	Buy wallet balance	
sellAmount	16	String	O	Sell amount	
buyAmount	16	String	O	Buy amount	

Name	Length	Type	Required	Description	Remark
balance	16	String	0	Sell wallet balance	

Possible Error Codes

Code	Description
00001	Failed to fetch data
00018	Currency is invalid
11002	OTC quote error

Request Body Sample (formatted)

```
{
  "query": {
    "buyCurrency": "USD",
    "sellCurrency": "GBP"
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "sellCurrency": "GBP",
    "buyCurrency": "USD",
    "clientId": 8524010424,
    "rateExpiryTime": "2025-10-11 14:21:15",
    "rate": "1.345119262388",
    "quoteId": "1155389391925733471",
    "balance": "10.79",
    "sellWalletBalance": "10.79",
    "buyWalletBalance": "4725.06"
  }
}
```

4.3.5. Request OTC

[POST] /openapi/v1/otc/request

Request

Name	Length	Type	Required	Description	Remark
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Available Currency
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Available Currency
sellAmount	-	String	C	Sell amount	Sell amount (optional; either sellAmount or buyAmount must be provided)
buyAmount	-	String	C	Buy amount	Buy amount (optional; either buyAmount or sellAmount must be provided)
dtcQuoteId	19	Long	M	DTC quote ID	Get from GET OTC Rate API quoteId response
referenceNo	19	String	M	Reference number (unique ID from your system)	
clientId	20	Long	M	Client ID	

NOTE

dtcQuoteId is a one-time quotation reference. Once used it becomes invalid and cannot be reused. A new dtcQuoteId must be obtained by calling /get-otc-rate(Get OTC Rate) for the next request.

Response

Name	Length	Type	Required	Description	Remark
id	-	Long	M	OTC ID	
status	-	Integer	M	OTC Status	Refer to Enum List - OtcStatus
clientId	-	Long	M	Client ID	
sellCurrency	3	String	M	Currency to sell	Refer to Enum List - Available Currency
buyCurrency	3	String	M	Currency to buy	Refer to Enum List - Available Currency
rate	16	String	M	Exchange rate	
dtcQuoteId		Long	M	DTC quote ID	
sellAmount	16	String	M	Sell amount	
buyAmount	16	String	M	Buy amount	

Name	Length	Type	Required	Description	Remark
operator	128	String	M	Operator	
referenceNo	19	String	M	Reference number (unique ID from your system)	
approver	128	String	M	Approver	
completedTime	18	String	M	Completed time	Format: yyyy-MM-dd HH:mm:ss
lastUpdatedDate	18	String	M	Last updated date	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00006	Access denied
00018	Currency is invalid
20999	Wallet account error (Other error)
11999	OTC error (Other error)

Request Body Sample (formatted)

```
{
  "query": {
    "buyCurrency": "USD",
    "buyAmount": "1",
    "sellCurrency": "GBP",
    "sellAmount": "1",
    "dtcQuoteId": "1155389391925733471",
    "referenceNo": "173561238951"
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 2510101421380624067,
    "status": 5,
    "clientId": 8524010424,
    "sellCurrency": "GBP",
    "buyCurrency": "USD",
  }
}
```

```

    "rate": "1.345119262388",
    "dtcQuoteId": 1155389391925733471,
    "sellAmount": "0.75",
    "buyAmount": "1",
    "operator": "[8524010424] openapi",
    "referenceNo": "173561238951",
    "approver": "[8524010424] openapi",
    "completedTime": "2025-10-10 14:21:39",
    "lastUpdatedDate": "2025-10-10 14:21:39"
  }
}

```

4.4. Crypto

4.4.1. Get Whitelist Address

Request

```
[POST] /openapi/v1/crypto-account/whitelist/search-obj
```

This API returns the whitelist address list. It is used to check or query whether a wallet address is in the whitelist.

- For deposit: the sending address must be added to the whitelist, otherwise the transaction will be blocked with Risk Withheld status.
- For withdrawal: the receiving address must be selected from the whitelist.

Request

No.	Name	Type	Length	Required	Description	Remark
1	mainNet	String	1	O	Mainnet identifier	Refer to Enum List - MainNet

Response

resultList Object

Name	Type	Length	Required	Description	Remark
address	String	255	M	Wallet address	
enabled	Boolean	1	M	Whether the address is enabled	
id	Long	19	M	Unique whitelist address ID	
label	String	63	O	Custom label for the address	

Name	Type	Length	Required	Description	Remark
lastUpdatedDate	LocalDateTime	—	M	Last update time of the address	Format: yyyy-MM-dd HH:mm:ss
level	Integer	1	M	Trust level of the address	Refer to Enum List - RiskLevel
mainNet	Enum	1	M	Mainnet type	Refer to Enum List - MainNet
type	Enum	1	M	Address type	Refer to Enum List - WalletAddressType

Request Body Sample (formatted)

```
{
  "query": {
    "mainNet": "3"
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "resultList": [
    {
      "address": "0x39c929CCB23116e39aE1297B0157F38ecf501D6a",
      "enabled": true,
      "id": 2851,
      "label": "testetsesss",
      "lastUpdatedDate": "2025-08-20 10:17:10",
      "level": 20,
      "mainNet": 3,
      "type": 1
    },
    {
      "address": "0x73868CA4bdeEe665FEF2d62C25D64bfD58B63327",
      "enabled": true,
      "id": 2845,
      "label": "whitelist",
      "lastUpdatedDate": "2025-08-20 09:16:24",
      "level": 20,
      "mainNet": 3,
      "type": 1
    }
  ]
}
```

```
}
```

4.4.2. Get Crypto Transaction History

Request

```
[POST] /openapi/v1/crypto-txn/search
```

This API retrieves the paginated transaction history of a client. It supports filtering by time range, transaction state, transaction type, address ID, mainnet, and currency.

The transaction list is ordered by id in descending order, so the latest transaction appears first.

Request Parameters

Name	Type	Length	Required	Description	Remark
state	Enum	1	O	Transaction state	Refer to Enum List - CryptoTransactionState
type	Enum	1	O	Transaction type	Refer to Enum List - CryptoTransactionType
senderAddressId	Long	11	O	Sender address ID	
recipientAddressId	Long	11	O	Recipient address ID	
currency	String/Enum	7	O	Currency	Example: USDT
mainNet	Enum	1	O	Mainnet type	Refer to Enum List - MainNet
transactionDateStart	LocalDateTime	—	O	Start date (00:00:00 of the day)	Automatically truncated
transactionDateEnd	LocalDateTime	—	O	End date (23:59:59 of the day)	Automatically truncated

Response

resultList Object

Name	Type	Length	Required	Description	Remark
id	Long	19	M	Unique transaction ID	
state	Enum	1	M	Transaction state	Refer to Enum List - CryptoTransactionState

Name	Type	Length	Required	Description	Remark
type	Enum	1	M	Transaction type	See Enum List - CryptoTransactionType
mainNet	Enum	1	M	Mainnet type	Refer to Enum List - MainNet
amount	String	—	M	Transaction amount	
currency	String	255	M	Currency	Example: USDT
transactionFee	String	—	O	Transaction fee	May be null
txnHash	String	255	O	Blockchain transaction hash	May be null
referenceNo	String	255	O	Business reference number	May be null
remark	String	255	O	Transaction remark	May be null
requestTimestamp	LocalDateTime	—	M	Transaction request time	Format: yyyy-MM-dd HH:mm:ss
lastUpdatedDate	LocalDateTime	—	M	Last updated time	Format: yyyy-MM-dd HH:mm:ss
gasFee	String	63	O	gas fee	-

Error Response

Scenario	Code	Message
Query stablecoin transaction failed	11999	Query stablecoin transaction failed

Request Body Sample (formatted)

```
{
  "page": {
    "current": 1,
    "size": 10
  },
  "query": {
    "state": "5",
    "type": "1",
    "senderAddressId": "2233",
    "recipientAddressId": "112",
    "currency": "USDT",
    "mainNet": "3",
    "transactionDateStart": "2025-09-01",
    "transactionDateEnd": "2025-09-30"
  }
}
```

```
}
```

Response Sample (formatted)

```
{
  "resultList": [
    {
      "id": 2508261801180309870,
      "type": 1,
      "mainNet": 3,
      "amount": "1.00",
      "currency": "USDC",
      "transactionFee": "30.00",
      "txnHash":
"0x05207c3f87c77ffbf0e038d1667794d1fde8f0081ad118f8c4a01673f4a88fe1",
      "referenceNo": "REF20250901001",
      "remark": "User deposit",
      "requestTimestamp": "2025-08-26 18:01:18",
      "lastUpdatedDate": "2025-08-26 18:01:18"
    },
    {
      "id": 2508261755380309864,
      "type": 1,
      "mainNet": 3,
      "amount": "3.00",
      "currency": "USDC",
      "txnHash":
"0x9a37ece5b6b7301fbc5f85241b40b7b46a5b7ed9feab2fb6777a055a9f45b127",
      "requestTimestamp": "2025-08-26 17:55:39",
      "lastUpdatedDate": "2025-08-26 17:55:39"
    }
  ],
  "header": {
    "success": true
  },
  "pagination": {
    "current": 1,
    "size": 50,
    "total": 2
  }
}
```

4.4.3. Get Deposit Address

Request

```
[POST] /openapi/v1/crypto-account/deposit-address/search-obj
```

This API is used to retrieve the client's deposit addresses (assigned by dtcpay and owned by DTC).

- If a specific mainnet is provided, the API returns deposit addresses under that network.
- If no mainnet is specified, the API returns deposit addresses across all available networks.

Request Parameters

No.	Name	Type	Length	Required	Description	Remark
1	mainNet	String	1	O	Mainnet identifier	Refer to Enum List - MainNet

Response

resultList Object

Name	Type	Length	Required	Description	Remark
id	Long	19	M	Unique deposit address ID	
address	String	255	M	Wallet address	
enabled	Boolean	1	M	Whether the address is enabled	
label	String	63	O	Custom label for the address	
lastUpdatedDate	LocalDateTime	—	M	Last update time of the address	Format: yyyy-MM-dd HH:mm:ss
level	Integer	1	M	Risk level of the address	Refer to Enum List - RiskLevel
mainNet	Enum	1	M	Mainnet type	Refer to Enum List - MainNet
type	Enum	1	M	Address type (fixed as DTC_CLIENT_WALLET)	Refer to Enum List - WalletAddress Type

Request Body Sample (formatted)

```
{
  "query": {
    "mainNet": "3"
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "resultList": [
    {
      "address": "0x29451adE1244Ce8F6E6fBA87E7A27c06dbc07249",
      "enabled": true,
      "id": 1999,
      "label": "label",
      "lastUpdatedDate": "2025-07-30 22:01:44",
      "level": 10,
      "mainNet": 3,
      "type": 2
    }
  ]
}
```

4.4.4. Get Crypto Transaction Fee V2

Request

[POST] /openapi/v1/crypto-txn/get-txn-fee

This API is used to retrieve the transaction fee for a withdrawal request.

Request Parameters

Name	Type	Length	Required	Description	Remark
amount	String	—	M	Transaction amount	Example: 1000
currency	Enum/String	7	M	Withdrawal currency (BTC/ETH/USDT etc.)	Refer to Enum List - Currency

Response

Name	Type	Length	Required	Description	Remark
txnFee	String	—	M	Calculated withdrawal fee	Value is scaled to the currency exponent

Error Response

Scenario	Code	Message
Currency not provided or unsupported	00018	Currency is invalid

Request Body Sample (formatted)

```
{
  "query": {
    "amount": "0.01",
    "currency": "ETH"
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "txnFee": "25.00"
  }
}
```

4.4.5. Crypto Withdraw

Request

[POST] /openapi/v1/crypto-txn/withdraw

This API allows the client to withdraw cryptocurrency from dtc platform account to an external whitelisted wallet address.

Request Parameters

Name	Type	Length	Required	Description	Remark
kycWalletAddressId	Long	19	M	Whitelisted wallet address ID	
amount	String	—	M	Withdrawal amount	
currency	Enum	7	M	Withdrawal currency (BTC/ETH/USDT etc.)	Refer to Enum List - Currency
referenceNo	String	255	O	Reference number (must be unique if provided)	

Response

Name	Type	Length	Required	Description	Remark
txnId	Long	19	M	Unique withdrawal transaction ID	
success	Boolean	1	M	Whether the withdrawal request was successfully created	
txnHash	String	255	O	Blockchain transaction hash	

Error Response

Scenario	Code	Message
Currency not supported	00018	Currency is invalid
Client status invalid (frozen/suspended)	20999	Client ID is invalid
Wallet address does not belong to client	00006	Access denied
Invalid or non-whitelisted address	19999	Invalid recipient address id
Transaction fee greater than amount	19999	Transaction fee is greater than withdrawal amount
Insufficient balance	19999	Insufficient fund
Duplicate reference number	19999	ReferenceNo exists
System error	11999	Withdrawal request failed

Request Body Sample (formatted)

```
{
  "query": {
    "amount": "0.01",
    "currency": "ETH",
    "kycWalletAddressId": 516 ,
    "referenceNo": ""
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "success": true,
    "txnHash": "Crypto transaction on-chain hash/ID"
  }
}
```

```
}
```

4.4.6. Get Crypto Transaction

Request

```
[GET] /openapi/v1/crypto-txn/{txnId}
```

This API retrieves the details of a single crypto transaction by its unique transaction ID.

Request Parameters

No.	Name	Type	Length	Required	Description	Remark
1	txnId	String	19	Yes	Unique transaction ID from DTC	

Response

Name	Type	Length	Required	Description	Remark
id	Long	19	M	Unique transaction ID	
amount	String	63	M	Transaction amount	
clientId	Long	19	M	Client ID	
currency	String/Enum	7	M	Transaction currency (USDT, BTC, ETH etc.)	
transactionFee	String	63	O	Transaction fee	
lastUpdatedDate	LocalDateTime	—	M	Last updated time	Format: yyyy-MM-dd HH:mm:ss
mainNet	Enum	1	M	Mainnet type (TRON, ETH, BTC etc.)	Refer to Enum List - MainNet
operator	String	255	O	Operator executing the transaction	Example: dtc-settlement-engine
recipientAddressId	Long	11	O	Recipient address ID	
requestTimeStamp	LocalDateTime	—	M	Request initiation time	Format: yyyy-MM-dd HH:mm:ss
txnHash	String	255	O	Blockchain transaction hash	May be null
gasFee	String	63	O	gas fee	-
type	Enum	1	M	Transaction type	Refer to Enum List - CryptoTransactionType

Name	Type	Length	Required	Description	Remark
state	Enum	19	M	Transaction state	Refer to Enum List - CryptoTransactionState

Error Response

Scenario	Code	Message
Querying a transaction not owned by clientId	00006	Access denied
Invalid txnId provided	25005	The transaction hash is invalid

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "amount": "1.00",
    "clientId": 8525030164,
    "currency": "USDC",
    "transactionFee": "30.00",
    "id": 2508261801180309870,
    "lastUpdatedDate": "2025-08-26 18:01:18",
    "mainNet": 3,
    "operator": "dtc-settlement-engine",
    "recipientAddressId": 1999,
    "requestTimestamp": "2025-08-26 18:01:18",
    "txnHash": "0x05207c3f87c77ffbf0e038d1667794d1fde8f0081ad118f8c4a01673f4a88fe1",
    "type": 1,
    "state": 200
  }
}
```

4.4.7. Get Crypto Transaction By ReferenceNo

Request

```
[GET] /openapi/v1/crypto-txn/reference-number/{referenceNo}
```

This API retrieves a single crypto transaction by its unique business reference number.

Response

Name	Type	Length	Required	Description	Remark
id	Long	19	M	Unique transaction ID	
type	Enum	1	M	Transaction type (DEPOSIT / WITHDRAWAL etc.)	Refer to Enum List - CryptoTransactionType
state	Enum	1	M	Transaction state	Refer to Enum List - CryptoTransactionState
clientId	Long	19	M	Client ID	
mainNet	Enum	1	M	Mainnet type (ETH, TRON, BTC etc.)	Refer to Enum List - MainNet
amount	String	63	M	Transaction amount	
currency	String/Enum	7	M	Transaction currency (USDT, BTC, ETH etc.)	
transaction Fee	String	63	O	Transaction fee	
txnHash	String	255	O	Blockchain transaction hash	May be null
referenceNo	String	255	M	Business reference number	
remark	String	255	O	Transaction remark	May be null
requestTimestamp	LocalDate Time	—	M	Transaction request initiation time	Format: yyyy-MM-dd HH:mm:ss
lastUpdated Date	LocalDate Time	—	M	Last updated time	Format: yyyy-MM-dd HH:mm:ss
gasFee	String	63	O	gas fee	-
operator	String	255	O	Operator executing the transaction	-

Error Response

Scenario	Code	Message
Transaction not found by referenceNo	00001	Failed to fetch data

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 2509261123590312822,
    "type": 2,
    "state": 101,
    "clientId": 8525030164,
    "mainNet": 3,

```

```

    "amount": "100.00",
    "currency": "USDT",
    "gasFee": "0",
    "transactionFee": "5.00",
    "recipientAddressId": 2851,
    "referenceNo": "1122334455123",
    "operator": "test",
    "requestTimestamp": "2025-09-26 11:23:59",
    "lastUpdatedDate": "2025-09-26 11:24:00"
  }
}

```

4.5. Fiat

4.5.1. Add Remit Info

Endpoint

[POST] /openapi/v1/bank-account

Description

Add a new remit information (bank account) for the client. Used to store recipient bank account details for fiat withdrawals or invoice payments.

Request Parameters (wrapped in query object)

N o.	Name	Length	Type	Required	Description	Remark
1	ownerId	-	Integer	M	Client ID who owns this bank account	Must match the authenticated client ID
2	type	-	Integer	M	Remit info type	Refer to Enum List - RemitInfoType. 1=OWN (own account), 2=PAYEE (third party)
3	currency	3	String	M	Currency of the bank account	ISO 4217 Code. Refer to Enum List - Available Currency
4	bankType	-	Integer	M	Bank type	Refer to Enum List - BankType. 1=SWIFT, 2=SEPA, etc.
5	beneficiaryType	-	Integer	M	Beneficiary type	Refer to Enum List - BeneficiaryType. 1=INDIVIDUAL, 2=CORPORATE
6	beneficiaryFirstName	100	String	C	Beneficiary first name	Required when beneficiaryType=INDIVIDUAL
7	beneficiaryLastName	100	String	C	Beneficiary last name	Required when beneficiaryType=INDIVIDUAL

N o.	Name	Len gth	Typ e	Requ ired	Description	Remark
8	beneficiaryName	200	String	O	Full beneficiary name	Auto-generated for INDIVIDUAL type. Manual input for CORPORATE. For OWN type, will be overwritten by client name
9	beneficiaryAccount	50	String	M	Beneficiary account number	Must be strictly alphanumeric, no hyphens or spaces
10	beneficiaryAddress	500	String	O	Beneficiary address	
11	beneficiaryBankName	200	String	M	Beneficiary bank name	
12	beneficiaryBankAddress	500	String	O	Beneficiary bank address	
13	beneficiaryBankCountry	3	String	M	Beneficiary bank country	ISO 3166-1 alpha-3 code (e.g., SGP, USA, GBR). Must be valid
14	beneficiaryBankSwiftCode	11	String	C	Beneficiary bank SWIFT/BIC code	Required when bankType=SWIFT. Must be valid SWIFT code
15	iban	34	String	O	IBAN (International Bank Account Number)	For SEPA transfers
16	relationship	-	Integer	O	Relationship with beneficiary	Refer to Enum List - Relationship
17	isIntermediaryRequired	-	boolean	O	Whether intermediary bank is required	Default false
18	intermediaryBankCountry	3	String	C	Intermediary bank country	ISO 3166-1 alpha-3 code. Required if intermediary bank info provided
19	intermediaryBankName	200	String	C	Intermediary bank name	Required when isIntermediaryRequired=true
20	intermediaryBankSwiftCode	11	String	C	Intermediary bank SWIFT code	Required when isIntermediaryRequired=true
21	intermediaryBankAddress	500	String	O	Intermediary bank address	
22	intermediaryBankCity	100	String	O	Intermediary bank city	

N o.	Name	Len gth	Type	Required	Description	Remark
23	enabled	-	boolean	O	Whether this account is enabled	System will set to true automatically

Response

Name	Length	Type	Required	Description	Remark
result	-	Integer	M	ID of the created remit info record	

Possible Error Codes

Code	Description
00006	Access denied
00027	Params check failed
00007	Failed to add data
20999	Wallet account error (Other error)
24007	Invalid swift code
24009	Invalid country code
24015	Adding this recipient bank account is not supported

Request Body Sample (formatted)

```
{
  "query": {
    "beneficiaryType": 2,
    "beneficiaryAccount": "1231231123",
    "beneficiaryAddress": "12",
    "beneficiaryBankAddress": "12 Marina Blvd, Marina Bay Financia",
    "beneficiaryBankCity": "Singapore",
    "beneficiaryBankCountry": "SGP",
    "beneficiaryBankName": "DBS Bank Ltd",
    "beneficiaryBankSwiftCode": "DBSSSGSGXXX",
    "beneficiaryFirstName": "mirya",
    "beneficiaryLastName": "aa",
    "currency": "SGD",
    "enabled": true,
    "type": 1,
    "bankType": 1,
    "ownerId": 1669025274037,
    "relationship": 0,
    "isIntermediaryRequired": false
  }
}
```

NOTE | The swiftCode and bankCountry must be valid and correspond to real existing banks.

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": 1234
}
```

4.5.2. Update Remit Info

Endpoint

[PUT] /openapi/v1/bank-account/{id}

Description

Update an existing remit information. Note: Cannot update if there are pending transactions (UNPAID, PARTIAL_PAID, or PROCESSING status) associated with this account.

Path Parameters

No.	Name	Type	Required	Description
1	id	Integer	M	Remit info ID to update

Request Parameters (wrapped in query object)

No.	Name	Length	Type	Required	Description	Remark
1	ownerId	-	Integer	M	Client ID who owns this bank account	Must match the authenticated client ID
2	type	-	Integer	M	Remit info type	Refer to Enum List - RemitInfoType
3	currency	3	String	M	Currency of the bank account	ISO 4217 Code
4	bankType	-	Integer	M	Bank type	Refer to Enum List - BankType
5	beneficiaryType	-	Integer	M	Beneficiary type	Refer to Enum List - BeneficiaryType
6	beneficiaryFirstName	100	String	C	Beneficiary first name	Required when beneficiaryType=INDIVIDUAL
7	beneficiaryLastName	100	String	C	Beneficiary last name	Required when beneficiaryType=INDIVIDUAL
8	beneficiaryName	200	String	O	Full beneficiary name	For OWN type, will be overwritten by client name

No .	Name	Length	Type	Required	Description	Remark
9	beneficiaryAccount	50	String	M	Beneficiary account number	Must be strictly alphanumeric
10	beneficiaryAddress	500	String	O	Beneficiary address	
11	beneficiaryBankName	200	String	M	Beneficiary bank name	
12	beneficiaryBankAddress	500	String	O	Beneficiary bank address	
13	beneficiaryBankCountry	3	String	M	Beneficiary bank country	ISO 3166-1 alpha-3 code. Must be valid
14	beneficiaryBankSwiftCode	11	String	C	Beneficiary bank SWIFT/BIC code	Required when bankType=SWIFT
15	iban	34	String	O	IBAN	
16	isIntermediaryRequired	-	boolean	O	Whether intermediary bank is required	
17	intermediaryBankCountry	3	String	C	Intermediary bank country	ISO 3166-1 alpha-3 code. Required if intermediary bank info provided
18	intermediaryBankName	200	String	C	Intermediary bank name	Required when isIntermediaryRequired=true
19	intermediaryBankSwiftCode	11	String	C	Intermediary bank SWIFT code	Required when isIntermediaryRequired=true
20	intermediaryBankAddress	500	String	O	Intermediary bank address	
21	enabled	-	boolean	O	Whether this account is enabled	

Response

Name	Length	Type	Required	Description	Remark
result	-	Integer	M	ID of the updated remit info record	

Possible Error Codes

Code	Description
00002	Failed to update data
00006	Access denied
24007	Invalid swift code
24009	Invalid country code

Code	Description
24015	Adding this recipient bank account is not supported
25002	Recipient information already exists
25005	Bank details cannot be changed during a pending transaction

Request Body Sample (formatted)

```
{
  "query": {
    "beneficiaryType": 1,
    "beneficiaryAccount": "1230006789",
    "beneficiaryAddress": "7 STRAITS VIEW #05-01 MARINA ONE EAST TOWER SINGAPORE (018936)",
    "beneficiaryBankAddress": "OCBC Office",
    "beneficiaryBankCountry": "SGP",
    "beneficiaryBankName": "OCBC",
    "beneficiaryBankSwiftCode": "OCCBSGSGXXX",
    "beneficiaryFirstName": "John",
    "beneficiaryLastName": "Doe",
    "currency": "USD",
    "enabled": true,
    "type": 2,
    "bankType": 1,
    "ownerId": 8520120007,
    "isIntermediaryRequired": false
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": 1235
}
```

4.5.3. Delete Remit Info

Endpoint

```
[POST] /openapi/v1/bank-account/disable
```

Description

Disable (soft delete) a remit information. The record is not physically deleted but marked as disabled (enabled=false).

Request Parameters (wrapped in query object)

No.	Name	Length	Type	Required	Description	Remark
1	id	-	Integer	M	Remit info ID to disable	

Response

Name	Length	Type	Required	Description	Remark
result	-	Integer	M	ID of the disabled remit info record	

Possible Error Codes

Code	Description
00006	Access denied
25999	Bank account error (Other error)

Request Body Sample (formatted)

```
{
  "query": {
    "id": 1234
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": 1234
}
```

4.5.4. Get Fiat Deposit Account Info

Endpoint

```
[GET] /openapi/v1/bank-account/remit-info/{currency}
```

Description

Get the list of remit information (bank accounts) for a specific currency. Returns all enabled bank accounts owned by the authenticated client for the specified currency.

Path Parameters

No.	Name	Type	Required	Description
1	currency	String	M	Fiat currency code (ISO 4217, e.g., USD, SGD, EUR)

Response (Array of RemitInfo objects)

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Remit info ID	
ownerId	-	Integer	M	Client ID who owns this account	
currency	7	String	M	Currency code	ISO 4217
beneficiaryType	-	Integer	M	Beneficiary type	Refer to Enum List - BeneficiaryType
beneficiaryFirstName	140	String	O	Beneficiary first name	
beneficiaryLastName	140	String	O	Beneficiary last name	
beneficiaryName	255	String	M	Full beneficiary name	
beneficiaryAccount	255	String	M	Beneficiary account number	
beneficiaryAddresses	255	String	O	Beneficiary address	
beneficiaryBankName	255	String	M	Beneficiary bank name	
beneficiaryBankAddress	255	String	O	Beneficiary bank address	
beneficiaryBankCountry	3	String	M	Beneficiary bank country	ISO 3166-1 alpha-3
beneficiaryBankSwiftCode	255	String	O	Beneficiary bank SWIFT code	
enabled	-	boolean	M	Whether this account is enabled	
isIntermediaryRequired	-	boolean	O	Whether intermediary bank is required	
intermediaryBankCountry	3	String	O	Intermediary bank country	
lastUpdatedDate	19	String	M	Last update timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00006	Access denied

Code	Description
00001	Failed to fetch data

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": [
    {
      "id": 1234,
      "ownerId": 8520120007,
      "currency": "USD",
      "beneficiaryType": 1,
      "beneficiaryFirstName": "John",
      "beneficiaryLastName": "Doe",
      "beneficiaryName": "John Doe",
      "beneficiaryAccount": "1230006789",
      "beneficiaryAddress": "7 STRAITS VIEW #05-01 MARINA ONE EAST TOWER SINGAPORE (018936)",
      "beneficiaryBankName": "OCBC",
      "beneficiaryBankAddress": "OCBC Office",
      "beneficiaryBankCountry": "SGP",
      "beneficiaryBankSwiftCode": "OCBCSGSGXXX",
      "enabled": true,
      "isIntermediaryRequired": false,
      "lastUpdatedDate": "2025-01-15 10:30:00"
    }
  ]
}
```

4.5.5. Get Remit Info

Endpoint

```
[GET] /openapi/v1/get-config-remit-info-by-currency/{currency}
```

Description

Get the default DTC platform remit information (deposit account) for a specific currency. This endpoint returns the DTC's receiving bank account information where clients should transfer funds for deposits. This is typically used in the fiat deposit flow to obtain the destination bank account details.

Path Parameters

No .	Name	Type	Required	Description	Remark
1	currency	String	M	Fiat currency code	Supported: SGD, USD, EUR, GBP, HKD, JPY, AUD, CAD, CNH, MYR, AED

Response (RemitInfo object)

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Remit info ID	
ownerId	-	Integer	M	Owner ID	Always 1 for DTC platform accounts
type	-	Integer	M	Remit info type	Always PAYEE type for platform accounts
currency	7	String	M	Currency code	ISO 4217
bankType	-	Integer	M	Bank type	Refer to Enum List - BankType
beneficiaryType	-	Integer	M	Beneficiary type	Refer to Enum List - BeneficiaryType
beneficiaryName	255	String	M	Beneficiary name (DTC company name)	
beneficiaryAccount	255	String	M	DTC's receiving account number	
beneficiaryAddresses	255	String	O	DTC's address	
beneficiaryBankName	255	String	M	Bank name	
beneficiaryBankAddress	255	String	O	Bank address	
beneficiaryBankCountry	3	String	M	Bank country	ISO 3166-1 alpha-3
beneficiaryBankSwiftCode	255	String	O	Bank SWIFT code	
iban	255	String	O	IBAN	For SEPA transfers
enabled	-	boolean	M	Whether this account is enabled	Should always be true
isIntermediaryRequired	-	boolean	O	Whether intermediary bank is required	
lastUpdatedDate	19	String	M	Last update timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
21999	Fiat transaction error - No supporting bank account for the specified currency

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 5678,
    "ownerId": 1,
    "type": 2,
    "currency": "USD",
    "bankType": 1,
    "beneficiaryType": 2,
    "beneficiaryName": "Digital Treasures Center Pte Ltd",
    "beneficiaryAccount": "7001234567",
    "beneficiaryAddress": "7 Straits View, Marina One East Tower, Singapore 018936",
    "beneficiaryBankName": "DBS Bank Ltd",
    "beneficiaryBankAddress": "12 Marina Boulevard, Singapore 018982",
    "beneficiaryBankCountry": "SGP",
    "beneficiaryBankSwiftCode": "DBSSSGSGXXX",
    "enabled": true,
    "isIntermediaryRequired": false,
    "lastUpdatedDate": "2025-01-01 00:00:00"
  }
}
```

4.5.6. Get Fiat Transaction

Endpoint

```
[GET] /openapi/v1/fiat-txn/{txnId}
```

Description

Get details of a specific fiat transaction by its transaction ID. Returns transaction information including amount, currency, state, type, and associated account details.

Path Parameters

No.	Name	Type	Required	Description
1	txnId	Integer	M	Transaction ID (unique identifier from DTC)

Response

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Transaction ID	
clientId	-	Integer	M	Client ID who owns this transaction	
type	-	Integer	M	Transaction type	Refer to Enum List - FiatTransactionType
state	-	Integer	M	Transaction state	Refer to Enum List - FiatTransactionState
currency	7	String	M	Transaction currency	ISO 4217 Code
amount	-	String	M	Transaction amount	
transactionFee	-	String	O	Transaction fee charged	
senderAccountId	-	Integer	O	Sender account ID	
recipientAccountId	-	Integer	O	Recipient account ID	
remitInfoId	-	Integer	O	Associated remit info ID	Deprecated, use recipientAccountId
lastUpdatedDate	19	String	M	Last update timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00006	Access denied
21999	Invalid Transaction Id (Fiat transaction error - Other error)

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 220124170942593,
    "clientId": 8521110075,
    "type": 1,
    "state": 101,
    "currency": "USD",
    "amount": "200000",
    "transactionFee": "12",
    "senderAccountId": null,
    "recipientAccountId": 22,
```

```

    "remitInfoId": 22,
    "lastUpdatedDate": "2022-01-24 17:09:42"
  }
}

```

4.5.7. Get Fiat Transaction History

Endpoint

[POST] /openapi/v1/fiat-txn/search

Description

Search and retrieve fiat transaction history with filtering and pagination support. Allows filtering by transaction state, type, currency, and date range.

Request Parameters (wrapped in query object)

No.	Name	Length	Type	Required	Description	Remark
1	state	-	Integer	O	Transaction state filter	Refer to Enum List - FiatTransactionState
2	type	-	Integer	O	Transaction type filter	Refer to Enum List - FiatTransactionType
3	currency	3	String	O	Currency filter	ISO 4217 Code
4	createTimeStart	-	String	O	Start date for filtering	Format: yyyy-MM-dd or yyyy-MM-dd HH:mm:ss. Time will be truncated to day start
5	createTimeEnd	-	String	O	End date for filtering	Format: yyyy-MM-dd or yyyy-MM-dd HH:mm:ss. Time will be truncated to day end

Pagination Parameters (in page object)

No.	Name	Type	Required	Description	Remark
1	current	Integer	O	Current page number	Default: 1
2	size	Integer	O	Page size	Default: 10, Max: 100

Response

Name	Length	Type	Required	Description	Remark
header		object	M	Response header	
header.success	-	boolean	M	Whether request succeeded	
pagination		object	M	Pagination information	
pagination.currentPage	-	Integer	M	Current page number	
pagination.pageSize	-	Integer	M	Number of items per page	

Name	Length	Type	Required	Description	Remark
pagination.totalSize	-	Integer	M	Total number of items	
resultList		array	M	Array of FiatTransaction objects	

FiatTransaction Object Fields

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Transaction ID	
clientId	-	Integer	M	Client ID	
type	-	Integer	M	Transaction type	Refer to Enum List - FiatTransactionType
state	-	Integer	M	Transaction state	Refer to Enum List - FiatTransactionState
currency	7	String	M	Currency code	ISO 4217
amount	-	String	M	Transaction amount	
transactionFee	-	String	O	Transaction fee	
senderAccountId	-	Integer	O	Sender account ID	
recipientAccountId	-	Integer	O	Recipient account ID	
remitInfoId	-	Integer	O	Associated remit info ID	
referenceNo	255	String	O	Reference number	
operator	255	String	O	Operator who created transaction	
approver	255	String	O	Approver who approved transaction	
remark	255	String	O	Transaction remark	
purpose	-	Integer	O	Transfer purpose	Refer to Enum List - TransferPurpose
sourceOfIncome	-	Integer	O	Source of income	Refer to Enum List - SourceOfIncome
createdTime	19	String	M	Created timestamp	Format: yyyy-MM-dd HH:mm:ss
completedTime	19	String	O	Completed timestamp	Format: yyyy-MM-dd HH:mm:ss

Name	Length	Type	Required	Description	Remark
lastUpdatedDate	19	String	M	Last updated timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
11999	Query Fiat transaction failed

Request Body Sample (formatted)

```
{
  "query": {
    "state": 101,
    "type": 1,
    "currency": "USD",
    "createTimeStart": "2025-01-01",
    "createTimeEnd": "2025-01-31"
  },
  "page": {
    "current": 1,
    "size": 20
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "pagination": {
    "currentPage": 1,
    "pageSize": 20,
    "totalSize": 45
  },
  "resultList": [
    {
      "id": 220124173604595,
      "clientId": 1618551970036,
      "type": 1,
      "state": 101,
      "currency": "USD",
      "amount": "100",
      "transactionFee": "2.5",
      "recipientAccountId": 57,
      "remitInfoId": 57,
      "referenceNo": "REF123456",
    }
  ]
}
```

```

    "operator": "[8520120007] Company Name",
    "createdTime": "2022-01-24 17:36:04",
    "lastUpdatedDate": "2022-01-24 17:36:04"
  }
]
}

```

4.5.8. Get Upload Token

[POST] /openapi/v1/file/get-upload-token

Description

This endpoint generates a temporary token that authorizes the upcoming file upload operation. Use this endpoint whenever a document needs to be uploaded. The token must be included in the /file/upload-by-token/{token} request to complete the upload securely.

When a file is required to be uploaded

Fiat Deposit - Uploading a file is optional but recommended. - When uploading a file, set documentType = 200 (Fiat Deposit Receipt).

Fiat Withdrawal - type = 2 (Withdraw to own account): File upload is **not required**. - type = 3 (Invoice Payment to a third party): File upload **is required**, using documentType = 201 (Invoice).

Request

Name	Length	Type	Required	Description	Remark
documentType	-	Integer	M	Identifier for the document type	refer to Enum List [FileDocumentType]

Response

Name	Length	Type	Required	Description	Remark
token	128	String	M	upload token	This parameter will be used when uploading the file.

Request Body (Formatted)

```

{
  "query": {
    "documentType": 200
  }
}

```

Response Sample (formatted)

```
{
  "header":{
    "success":true
  },
  "result":{
    "token":"5e8a0b2afcc24abfa207c0b902d82f00"
  }
}
```

4.5.9. Upload File by Token

[POST] /openapi/v1/file/upload-by-token/{token}

Path Variables

No	Name	Type	Required	Description
1	token	String	Yes	This parameter is obtained from Get Upload Token

Request form-data

Name	Length	Type	Required	Description	Remark
file	file	File	M	The file to upload	

Response

Name	Length	Type	Required	Description	Remark
id	128	Long	M	File ID	
status	-	Integer	M	File status	refer to Enum List [FileStatus]
contentType	128	String	M	File MIME type	

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 11368,
    "status": 3,
    "contentType": "image/jpeg"
  }
}
```

4.5.10. Get Fiat Transaction Fee

[POST] /openapi/v1/fiat/get-txn-fee

Request

Name	Length	Type	Required	Description	Remark
recipientRemitInfoId	-	Long	M	remit info ID	
amount	-	String	M	The amount to withdraw	

Response

Name	Length	Type	Required	Description	Remark
result	-	String	M	The transaction fee	

Request Body (Formatted)

```
{
  "query": {
    "recipientRemitInfoId": 12,
    "amount": "100"
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": 10
}
```

4.5.11. Fiat Withdraw

[POST] /openapi/v1/fiat/withdraw

Request

Name	Length	Type	Required	Description	Remark
remitInfoId	20	Long	M	Identifier for the remittance information	

Name	Length	Type	Required	Description	Remark
amount	20	String	M	The amount to withdraw	
currency	3	String	M	The currency code (e.g., USD, EUR)	
fileIds	3	Array	M	[]	
fiatTransaction		Object	M	Detailed transaction information for the fiat withdrawal	
fiatTransaction.type		FiatTransactionType	M	Type of the transaction (e.g., INVOICE, WITHDRAWAL)	
fiatTransaction.referenceNo		Number	M	Your system's unique ID	
fiatTransaction.recipientAccountId	20	Long	M	ID of the recipient's account, should be equivalent to remitInfoId	
fiatTransaction.recipientAmount	20	String	O	Optional. Use this if the sender should pay the fee	
fileIds		Array	C	fileIds	type is INVOICE, attachment invoice file is required, the ids from Upload File by Token API response, the file status must be NORMAL

Get fiat transaction details by client's reference number. This allows clients to retrieve transaction information using their own unique identifier rather than the DTC-generated transaction ID.

Response

Name	Length	Type	Required	Description	Remark
amount	-	String	M	The amount to withdraw	
clientId	-	Long	M	Client ID	
createdTime	19	String	M	Transaction creation time	the format is yyyy-MM-dd HH:mm:ss
currency	3	String	M	The currency code (e.g., USD, EUR)	

Name	Length	Type	Required	Description	Remark
id	-	Long	M	Transaction ID (unique id from DTC)	
lastUpdatedDate	19	String	M	Transaction last updated time	the format is yyyy-MM-dd HH:mm:ss
remitInfoId	-	Long	M	Identifier for the remittance information	
state		FiatTransactionState	M	State of the transaction (e.g., PENDING, COMPLETED)	refer to Enum List [FiatTransactionState]
type		FiatTransactionType	M	Type of the transaction (e.g., INVOICE, WITHDRAWAL)	refer to Enum List [FiatTransactionType]

Possible Error Codes

Code	Description
20999	Wallet account error (Other error) – e.g. ReferenceNo exists
21999	Fiat transaction error (Other error) – e.g. validation failed, withdrawal request failed
50011	Balance have exceed the limit.
00999	Common error (Other error) – e.g. Fiat Transaction type incorrect, Remit info not exist, RemitInfoId is empty

Request Body (Formatted)

```
{
  "query": {
    "remitInfoId": 2,
    "amount": "23.33",
    "currency": "USD",
    "fiatTransaction": {
      "type": 3,
      "recipientAccountId": 2,
      "referenceNo": "test-referenceNo",
      "recipientAmount": "23.33"
    },
    "fileIds": [
      3
    ]
  }
}
```

Response Sample (formatted)

```
{
  "header": {
```

```

    "success": true
  },
  "result": {
    "amount": "100",
    "clientId": 1618551970036,
    "createTime": "2022-01-24 17:36:04",
    "currency": "USD",
    "id": 220124173604595,
    "lastUpdatedDate": "2022-01-24 17:36:04",
    "remitInfoId": 57,
    "state": 101,
    "type": 2
  }
}

```

4.5.12. Fiat Deposit

[POST] /openapi/v2/fiat/deposit

Request

Name	Length	Type	Required	Description	Remark
amount	-	String	M	The amount to deposit	
referenceNo	255	String	M	The reference number for the transaction	
currency	3	String	M	The currency code (e.g., USD, EUR)	
fileIds		Array	C	fileIds	type is INVOICE, attachment invoice file is required, the ids from Upload File by Token API response, the file status must be NORMAL

Response

Name	Length	Type	Required	Description	Remark
id	-	Long	M	Transaction ID (unique id from DTC)	
amount	-	String	M	The amount to deposit	
clientId	-	Long	M	Client ID	
currency	3	String	M	The currency code (e.g., USD, EUR)	

Name	Length	Type	Required	Description	Remark
lastUpdatedDate	19	String	M	Transaction last updated time	the format is yyyy-MM-dd HH:mm:ss
state		FiatTransactionState	M	State of the transaction (e.g., PENDING, COMPLETED)	refer to Enum List [FiatTransactionState]
type		FiatTransactionType	M	Type of the transaction (e.g., INVOICE, WITHDRAWAL)	refer to Enum List [FiatTransactionType]
transactionFee	-	String	M	Transaction fee	
recipientAccountId	-	Long	M	Recipient account ID	

Possible Error Codes

Code	Description
21002	Validation error – e.g. Invalid Deposit Data, Invalid recipient bank account
20002	Wallet account does not exist
20004	Wallet account is inactive
50011	Balance have exceed the limit.

Request Body (Formatted)

```
{
  "query": {
    "amount": "1.00",
    "referenceNo": "saf5354534",
    "currency": "USD",
    "fileIds": [
      11374
    ]
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 250321145247611,
    "amount": "1.00",
    "clientId": 1721717098145,
    "currency": "USD",

```

```

    "lastUpdatedDate": "2025-03-21 14:52:47",
    "transactionFee": 0,
    "state": 101,
    "type": 1,
    "recipientAccountId": 408
  }
}

```

4.5.13. Get Fiat Transaction by Reference Number

Endpoint

```
[GET] /openapi/v1/fiat-txn/reference-number/{referenceNo}
```

Path Parameter

No.	Name	Type	Required	Description
1	referenceNo	String	M	Client's reference number (unique identifier provided during transaction creation)

Response (FiatTransaction object)

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Transaction ID	
clientId	-	Integer	M	Client ID	
type	-	Integer	M	Transaction type	Refer to Enum List - FiatTransactionType
state	-	Integer	M	Transaction state	Refer to Enum List - FiatTransactionState
currency	7	String	M	Currency code	ISO 4217
amount	-	String	M	Transaction amount	
transactionFee	-	String	O	Transaction fee	
senderAccountId	-	Integer	O	Sender account ID	
recipientAccountId	-	Integer	O	Recipient account ID	
remitInfoId	-	Integer	O	Associated remit info ID	
referenceNo	255	String	M	Reference number provided by client	

Name	Length	Type	Required	Description	Remark
operator	255	String	O	Operator who created transaction	
approver	255	String	O	Approver who approved transaction	
remark	255	String	O	Transaction remark	
purpose	-	Integer	O	Transfer purpose	Refer to Enum List - TransferPurpose
sourceOfIncome	-	Integer	O	Source of income	Refer to Enum List - SourceOfIncome
createdTime	19	String	M	Created timestamp	Format: yyyy-MM-dd HH:mm:ss
completedTime	19	String	O	Completed timestamp	Format: yyyy-MM-dd HH:mm:ss
lastUpdatedDate	19	String	M	Last updated timestamp	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00999	ReferenceNo not exist (Common error - Other error)
00001	Failed to fetch data

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 220124173604595,
    "clientId": 1618551970036,
    "type": 2,
    "state": 101,
    "currency": "USD",
    "amount": "100",
    "transactionFee": "2.5",
    "recipientAccountId": 57,
    "remitInfoId": 57,
    "referenceNo": "REF123456",
    "operator": "[8520120007] Company Name",
    "createdTime": "2022-01-24 17:36:04",
    "lastUpdatedDate": "2022-01-24 17:36:04"
  }
}
```

```
}
```

4.5.14. Open Currencies for Wallet Account

Request

[POST] /openapi/v1/wallet/open-currencies

Field	Type	Required	Description	
currencies	Array	Yes	List of currency codes to open wallet accounts for wallet	

Request Body (Formatted)

```
{
  "query": {
    "currencies": ["HKD", "SGD", "USD", "USDT"]
  }
}
```

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": {
    "openedCurrencies": [
      "SGD",
      "HKD",
      "USD",
      "USDT"
    ]
  }
}
```

Possible Error Codes

Code	Description
00006	Access denied
00010	Invalid parameters

4.6. Webhook Service

DTC supports webhook callbacks in various business scenarios. Customers can implement their own business logic in the callback interface.

To configure the webhook URL, please go to the **API key management** section in the wallet platform and update the corresponding API key information.

Callbacks use the HTTP protocol with the POST method. All request bodies are in JSON format and encoded in UTF-8.

A callback is considered **Failed** if any of the following conditions occur: - Request fails - Request times out - HTTP response status is not 2XX

Callbacks are divided into **Synchronous** and **Asynchronous** types: - **Synchronous** callbacks must respond in a timely manner and do **not** support retries. - **Asynchronous** callbacks do **not** require an immediate response and will be retried if they fail.

Webhook List

Name	Type	Retry Count	Retry Intervals
OPENAPI_FIAT_TXN_WEBHOOK	Asynchronous	6	15s, 30s, 60s, 120s, 240s, 480s
OPENAPI_CRYPTOTXN_WEBHOOK	Asynchronous	6	15s, 30s, 60s, 120s, 240s, 480s

Webhook Request Format

If you want to approve the transaction during a consuming webhook, the webhook must respond within the defined timeframe with the following:

Webhook fiat transaction request

```
{
  "event": "FIAT_TXN",
  "clientId": "852042402430001",
  "signature":
  "MxrYnCm9Q7J0Av0rISf8+T2kuTW1d/w0at8aaPaoiX08VWfun3XPokVlIx1TkHXdcitls09wzfUGtXQZq23xdg==",
  "data": {
    "transactionId": "2510119876543210",
    "referenceNo": "TXN20251109001",
    "amount": "1250.75",
    "currency": "USD",
    "transactionFee": "2.51",
    "type": 1,
    "state": 200,
    "recipientAccountId": "2433434",
    "createdTime": "2025-11-09 10:23:45",
    "lastUpdatedTime": "2025-11-09 10:25:12"
  }
}
```

Request Parameters

No .	Name	Length	Type	Required	Description
1	event	-	String	M	FIAT_TXN represents a fiat transaction, such as fiat withdrawal or other type
2	clientId	-	Long	M	Client id
4	signature	-	String	M	Refer 2.4 Request Signature on how to decrypt and check the originality of data (only encrypt value inside data)
5	transactionId	-	String	M	fiat transaction id, current length 19
6	referenceNo	-	Integer	M	The transaction referenceNo is usually a parameter provided by the customer
7	amount	-	Object	M	Transaction amount
8	currency	-	String	M	Currency code (e.g., USD, SGD). Refer to Enum List - Available Currency
9	transactionFee	-	String	M	The transaction fee is already included in the amount field
10	type	-	String	M	Refer to Enum List - FiatTransactionType
11	state	-	String	M	Refer to Enum List - FiatTransactionState
12	recipientAccountId	-	Object	C	Recipient account ID, deposit transaction return null
13	createTime	-	Integer	M	Create time
14	lastUpdatedTime	-	Integer	M	Last update time

Webhook crypto transaction request

```
{
  "event": "CRYPTO_TXN",
  "clientId": 852042402430001,
  "signature":
  "MxrYnCm9Q7J0Av0rISf8+T2kuTW1d/w0at8aaPaoiX08VWfun3XPokVLIx1TkHXdcitls09wzfUGtXQZq23xd
  ==",
  "data": {
    "transactionId": 2510119876543210,
    "referenceNo": "CRYPTO20251109001",
    "amount": "9.56",
    "transactionFee": "5",
    "currency": "USDC",
    "mainNet": 1,
  }
}
```



```

    "txnHash": "0xxxxxxx",
    "type": 1,
    "state": 200,
    "recipientAddressId": 789456123,
    "senderAddress": "0xf5e9a8c9B6Ad00BC87E.....",
    "riskWithheldReason": "unwhitelisted_address",
    "createTime": "2025-11-09 14:12:30",
    "lastUpdateTime": "2025-11-09 14:18:45"
  }
}

```

Request Parameters

No .	Name	Length	Type	Required	Description
1	event	-	String	M	CRYPTO_TXN represents a crypto transaction, such as withdrawal or other type
2	clientId	-	Long	M	Client id
4	signature	-	String	M	Refer 2.4 Request Signature on how to decrypt and check the originality of data (only encrypt value inside data)
5	transactionId	-	String	M	fiat transaction id, current length 19
6	referenceNo	-	Integer	M	The transaction referenceNo is usually a parameter provided by the customer
7	amount	-	Object	M	Transaction amount
8	currency	-	String	M	Currency code (e.g., USDC, USDT). Refer to Enum List - Available Currency
9	transactionFee	-	String	M	The transaction fee is already included in the amount field
10	type	-	String	M	Refer to Enum List - FiatTransactionType
11	state	-	String	M	Refer to Enum List - FiatTransactionState
12	recipientAddressId	-	Object	C	Recipient address ID, , deposit transaction return null
13	mainNet	-	Object	M	Refer to Enum List - MainNet
14	txnHash	-	Object	C	Crypto transaction hash
15	riskWithheldReason	-	Object	C	unwhitelisted_address, unmatched_address_owner, institution_limit

No.	Name	Length	Type	Required	Description
16	senderAddress	-	String	M	Sender's Address
17	createdTime	-	Integer	M	Create time
18	lastUpdatedTime	-	Integer	M	Last update time

Response Sample

```
{
  "response": "OK"
}
```

event

Name	Descriptor
CRYPTO_TXN	Crypto transaction event
FIAT_TXN	Fiat transaction event

4.7. Swap Order

4.7.1. Create Swap Order

Endpoint

[POST] /openapi/v1/swap-order

Description

Create a new swap order based on the provided quote and user input.

Request Parameters

No.	Name	Length	Type	Required	Description
1	quoteId	-	Integer	M	Rate ID, must be obtained from the query-quote API
2	amount	-	String	M	Deposit amount, must match the amount used in the quote
3	currency	5	String	M	Stablecoin currency code (e.g., USDC, USDT). Refer to Enum List - Available Currency
4	payoutMethod	-	Integer	M	Payout method. 1=PAY_NOW_WITHDRAW, 2=CARD_TOP_UP, 3=SGQR_WITHDRAW

No .	Name	Length	Type	Required	Description
5	qrCodeRecipientInfo	-	Object	C	Required when payoutMethod is 1 (PayNow) or 3 (SGQR)
5.1	qrCodeRecipientInfo.referenceNo	50	String	C	Reference number for recipient identification. Required if qrStr is not provided
5.2	qrCodeRecipientInfo.proxyType	-	String	C	PayNow proxy type: UEN, MOBILE, or NRIC. Required if qrStr is not provided
5.3	qrCodeRecipientInfo.proxyValue	50	String	C	PayNow proxy value (actual identifier). Required if qrStr is not provided
5.4	qrCodeRecipientInfo.qrStr	500	String	O	Full QR code string. If provided, will override proxyType and proxyValue
6	cardInfo	-	Object	C	Required when payoutMethod is 2 (CARD_TOP_UP)
6.1	cardInfo.cardId	-	Integer	C	Card ID. Required when payoutMethod is 2
7	callbackUrl	500	String	O	Callback URL. After the order status is changed, the swapOrder ID will be carried in the body as a POST request, along with the status of the order and the current timestamp

Response

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Unique ID of the swap order	
clientId	-	Integer	M	ID of the client who created the order	
type	-	Integer	M	Order type	SwapOrderType enum: 1=PAY_NOW_WITHDRAW, 2=CARD_TOP_UP, 3=SGQR_WITHDRAW
state	-	Integer	M	Order status	SwapOrderState enum: 1=WAITING_PAYIN, 2=COMPLETED, etc.
payinTransactionId	-	Integer	O	Associated pay-in transaction ID	
payinCurrency	7	String	M	Currency used for the pay-in	Usually a stablecoin (USDC, USDT)
payinAmount	-	String	M	Amount paid in by the client	
otcId	-	Integer	O	OTC order ID, if applicable	

Name	Length	Type	Required	Description	Remark
quoteId	-	Integer	M	Quote ID used when creating the order	
payoutTransactionId	-	Integer	O	Associated payout transaction ID	
payoutAmount	-	String	M	Amount paid out to the recipient	
payoutCurrency	7	String	M	Currency of the payout	e.g., SGD
operator	63	String	O	Operator handling the order	Can be system or user
createdAt	19	String	M	Timestamp when the order was created	Format: yyyy-MM-dd HH:mm:ss
expiredAt	19	String	M	Expiration timestamp of the order	Format: yyyy-MM-dd HH:mm:ss
updatedAt	19	String	M	Last updated timestamp of the order	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
11008	Invalid Quote ID
11004	Invalid amount
11020	Invalid currency
39002	Invalid payout method
25012	Invalid recipient address ID
39004	Reference number cannot be empty
39005	PayNow proxy type cannot be empty
39006	PayNow proxy value cannot be empty
39007	Card info cannot be empty
39008	Card ID cannot be empty

Request Body Sample (formatted)

```
{
  "quoteId": -7803018125589665165,
  "amount": "1",
  "currency": "USD",

```

```

    "payoutMethod": 1,
    "qrCodeRecipientInfo": {
      "referenceNo": "REF123456",
      "proxyType": "MOBILE",
      "proxyValue": "91234567",
      "qrStr":
"00020101021226370009SG.PAYNOW010120210201935231K030105204000053037025402105802SG5925D
IGITAL TREASURES CENTER 6009SINGAPORE62280124FPNI-25081213324002212966304E62B"
    },
    "cardInfo": {
      "cardId": 7890
    },
    "callbackUrl": "https://{callbackUrl}"
  }

```

Response Sample (formatted)

```

{
  "header": {
    "success": true
  },
  "result": {
    "id": 2507011357170784,
    "clientId": 1733914979001,
    "type": 2,
    "state": 1,
    "payinTransactionId": 123456789,
    "payinCurrency": "USDC",
    "payinAmount": "1.5",
    "otcId": 987654321,
    "quoteId": 1357924680,
    "payoutTransactionId": 246813579,
    "payoutAmount": "1.89",
    "payoutCurrency": "SGD",
    "operator": "system_auto",
    "createdAt": "2025-07-01T13:57:17",
    "expiredAt": "2025-07-02T13:56:51",
    "updatedAt": "2025-07-01T13:57:17"
  }
}

```

4.7.2. Query Quote

Endpoint

[POST] /openapi/v1/query-quote

Description

Get a quote for a swap order by specifying the processing currency, target currency, amount, and payment method type.

Request Parameters (wrapped in query object)

No .	Name	Length	Type	Required	Description
1	processingCurrency	5	String	C	Processing currency (the stablecoin being used, e.g., USDC, USDT). Required if processingAmount is provided
2	processingAmount	-	String	C	Processing amount. Either processingAmount or amount must be provided, but not both
3	currency	5	String	M	Target currency (the fiat currency to swap into, e.g., SGD). Refer to Enum List - Available Currency
4	amount	-	String	C	Transaction amount (amount in target currency). Either amount or processingAmount must be provided, but not both
5	paymentMethodType	-	Integer	M	Payment method type. Refer to Enum List - PaymentMethodType

Response

Name	Length	Type	Required	Description	Remark
sell	7	String	M	Currency being sold (processing currency)	e.g., USDC, USDT
buy	7	String	M	Currency being bought (target currency)	e.g., SGD, USD
sellAmount	-	String	M	Amount of sell currency required	
buyAmount	-	String	M	Amount of buy currency expected	
rate	-	String	M	Exchange rate between sell and buy currencies	
rateExpiryTime	19	String	M	Expiration time of this quote rate	After which it is invalid. Format: yyyy-MM-dd HH:mm:ss
quoteId	19	String	M	Quote ID, must be used when creating the swap order	

Possible Error Codes

Code	Description
00002	Parameter check failed
28001	Invalid QR code
00009	Unknown API error

Request Body Sample

```
{
  "query": {
    "processingCurrency": "USDC",
    "currency": "SGD",
    "amount": "5",
    "paymentMethodType": "4"
  }
}
```

Response Sample

```
{
  "header": {
    "success": true
  },
  "result": {
    "sell": "USDC",
    "buy": "SGD",
    "sellAmount": "3.92",
    "buyAmount": "5",
    "rate": "1.2765744944961372",
    "rateExpiryTime": "2025-09-16 12:00:28",
    "quoteId": "-183195842672497984"
  }
}
```

4.7.3. Cancel Swap Order

Endpoint

```
[DELETE] /openapi/v1/swap-order/{id}
```

Description

Cancel a swap order by its ID. Only orders in WAITING_PAYIN state can be cancelled.

Path Variables

No	Name	Type	Required	Description
1	id	Integer	M	The ID of the swap order to cancel

Response

Name	Length	Type	Required	Description	Remark
result		String	M	Result status	"success" when cancelled successfully

Possible Error Codes

Code	Description
00003	Invalid parameters
00001	Failed to fetch data
39009	The order status does not meet the requirements
39999	Swap order error (Other error) – e.g. cancel failed

Response Sample (formatted)

```
{
  "header": {
    "success": true
  },
  "result": "success"
}
```

4.7.4. Get Swap Order Details

Endpoint

```
[GET] /openapi/v1/swap-order/{id}
```

Description

Retrieve the details of a specific swap order.

Path Variables

No	Name	Type	Required	Description
1	id	Integer	M	The ID of the swap order

Response

Name	Length	Type	Required	Description	Remark
id	-	Integer	M	Unique ID of the swap order	
clientId	-	Integer	M	ID of the client who created the order	
type	-	Integer	M	Order type	SwapOrderType enum: 1=PAY_NOW_WITHDRAW, 2=CARD_TOP_UP, 3=SGQR_WITHDRAW
state	-	Integer	M	Order status	SwapOrderState enum: 1=WAITING_PAYIN, 2=COMPLETED, etc.
payinTransactionId	-	Integer	O	Associated pay-in transaction ID	

Name	Length	Type	Required	Description	Remark
payinCurrency	7	String	M	Currency used for the pay-in	Usually a stablecoin (USDC, USDT)
payinAmount	-	String	M	Amount paid in by the client	
otcId	-	Integer	O	OTC order ID, if applicable	
quoteId	-	Integer	M	Quote ID used when creating the order	
payoutTransactionId	-	Integer	O	Associated payout transaction ID	
payoutAmount	-	String	M	Amount paid out to the recipient	
payoutCurrency	7	String	M	Currency of the payout	e.g., SGD
callbackUrl	255	String	O	Callback URL provided during order creation	
operator	63	String	O	Operator handling the order	Can be system or user
createdAt	19	String	M	Timestamp when the order was created	Format: yyyy-MM-dd HH:mm:ss
expiredAt	19	String	M	Expiration timestamp of the order	Format: yyyy-MM-dd HH:mm:ss
updatedAt	19	String	M	Last updated timestamp of the order	Format: yyyy-MM-dd HH:mm:ss

Possible Error Codes

Code	Description
00003	Invalid parameters
00001	Failed to fetch data

Response Sample

```
{
  "header": {
    "success": true
  },
  "result": {
    "id": 2507011357170784,
    "clientId": 1733914979001,
  }
}
```

```

    "type": 2,
    "state": 1,
    "payinTransactionId": 123456789,
    "payinCurrency": "USDC",
    "payinAmount": "1.5",
    "otcId": 987654321,
    "quoteId": 1357924680,
    "payoutTransactionId": 246813579,
    "payoutAmount": "1.89",
    "payoutCurrency": "SGD",
    "operator": "system_auto",
    "createdAt": "2025-07-01T13:57:17",
    "expiredAt": "2025-07-02T13:56:51",
    "updatedAt": "2025-07-01T13:57:17"
  }
}

```

4.7.5. QR Code Query

Endpoint

[POST] /openapi/v1/qrcode/query

Description

QR code query API, supports PayNow and SGQR QR codes.

Request Parameters (wrapped in query object)

No.	Name	Length	Type	Required	Description
1	qrCode	500	String	M	Full QR code string (supports both PayNow and SGQR formats)

Response

Name	Length	Type	Required	Description	Remark
payoutMethod	-	Integer	M	Payout method	1=PAY_NOW_WITHDRAW, 3=SGQR_WITHDRAW
currency	7	String	M	Currency	Usually SGD
proxyIdType	-	Integer	C	PayNow proxy type	Required for PayNow. Refer to Enum List - ProxyType
proxyIdValue	255	String	C	PayNow proxy value	Required for PayNow
amountModifier		Boolean	M	Whether amount can be modified	false for fixed amount, true for variable amount

Name	Length	Type	Required	Description	Remark
amount	-	String	O	Transaction amount	Present for PayNow with fixed amount
note	255	String	O	Payment note	Present for PayNow
acceptorName	255	String	O	Merchant/acceptor name	
expiredTime	19	String	O	QR code expiration time	Present for SGQR
maxAmount	-	String	O	Maximum transaction amount	Present for SGQR

Possible Error Codes

Code	Description
00002	Parameter check failed
28001	Invalid QR code
00009	Unknown API error

Request Body Sample (formatted) - PayNow

```
{
  "query": {
    "qrCode":
    "00020101021226370009SG.PAYNOW010120210201935231K030105204000053037025402105802SG5925DIGITAL TREASURES CENTER 6009SINGAPORE62280124FPNI-25081213324002212966304E62B"
  }
}
```

Request Body Sample (formatted) - SGQR

```
{
  "query": {
    "qrCode":
    "00020101021126860011SG.COM.NETS01231012345678999123123590002111100001712030820003481100129908ADF7AF2451800007SG.SGQR0112000A34010103020701.0001030606911104020205028606040000708202004095204000053037025802SG5904Test6009Singapore63049B90"
  }
}
```

Response Sample (formatted) - PayNow

```
{
  "header": {
    "success": true
  },
  ...
}
```

```
"result": {
  "payoutMethod": 1,
  "currency": "SGD",
  "proxyIdType": 2,
  "proxyIdValue": "201935231KBXG",
  "amountModifier": false,
  "amount": "0.2",
  "note": "FPNI-2509151011350282819",
  "acceptorName": "DTC Mock"
}
```

Response Sample (formatted) - SGQR

```
{
  "header": {
    "success": true
  },
  "result": {
    "payoutMethod": 3,
    "currency": "SGD",
    "amountModifier": true,
    "expiredTime": "2025-09-15 10:28:00",
    "acceptorName": "TEST",
    "maxAmount": "20000"
  }
}
```

Appendix A: Data Structure

These resources are expressed as objects in JSON, and are described in more detail below.

WalletAccount

Name	Type	Descriptor
id	Integer	Transaction ID
clientId	Integer	Client ID
status	Enum	Status: WalletStatus
currency	String	Currency
balance	String	Balance
label	String	Label
createdDate	String	Account created time
lastUpdatedDate	String	Account updated time

FiatTransaction

Name	Type	Descriptor
id	Integer	Transaction ID
type	Enum	Transaction type: FiatTransactionType
state	Enum	Transaction state: FiatTransactionState
clientId	Integer	Transaction's clientId
senderAccountId	Integer	Sender account ID
recipientAccountId	Integer	Recipient account ID
recipientName	String	Recipient name
currency	String	Transaction currency
recipientCurrency	String	Transaction recipient currency
feeCurrency	String	Transaction fee currency
amount	String	Transaction amount
recipientAmount	String	Transaction recipient amount
transactionFee	String	Transaction fee
referenceNo	String	Transaction reference number
operator	String	Transaction operator
approver	String	Transaction approver
remark	String	Transaction remark
purpose	String	Transaction purpose
note	String	Transaction note

Name	Type	Descriptor
createdTime	String	Transaction created time
lastUpdatedDate	String	Transaction updated time

CryptoTransaction

Name	Type	Descriptor
id	Integer	Transaction ID
type	Enum	Transaction type: CryptoTransactionType
state	Enum	Transaction state: CryptoTransactionState
clientId	Integer	Transaction's clientId
mainNet	Enum	Blockchain network: MainNet
amount	String	Transaction amount
currency	String	Transaction currency
gasFee	String	Transaction recipient amount
transactionFee	String	Transaction fee
senderAddressId	Integer	Sender address ID
recipientAddressId	Integer	Recipient address ID
txnHash	Integer	Blockchain transaction hash
referenceNo	String	Transaction reference number
operator	String	Transaction operator
approver	String	Transaction approver
remark	String	Transaction remark
createdTime	String	Transaction created time
lastUpdatedDate	String	Transaction updated time

Otc

Name	Type	Descriptor
id	Integer	Transaction ID
status	Enum	Transaction statue: OtcStatus
clientId	Integer	Transaction's clientId
sellCurrency	String	Transaction sell currency
buyCurrency	String	Transaction buy currency
rate	String	Transaction rate
dtcQuoteId	Integer	Transaction quote ID
sellAmount	String	Transaction sell amount

Name	Type	Descriptor
buyAmount	String	Transaction buy amount
referenceNo	String	Transaction reference number
operator	String	Transaction operator
approver	String	Transaction approver
remark	String	Transaction remark
completedTime	String	Transaction created time
lastUpdatedDate	String	Transaction updated time

Appendix B: Error Structure

To ensure a consistent experience across all DTC Pay APIs, the system follows a standardized structure for all error responses.

When an API request fails, DTC Pay returns an error object under the header field containing details of the failure.

DTC Pay returns errors in the following structure:

```
{
  "header": {
    "success": false,
    "errCode": "<Unique error code>",
    "errMsg": "<Description of the error>"
  }
}
```


Appendix C: Enum List

ExchangeType

Name	ID	Descriptor
BUYING_RATE	1	Buying Rate
SELLING_RATE	2	Selling Rate
MIDDLE_RATE	3	Middle Rate
CASH_BUYING_RATE	4	Cash Buying Rate
CASH_SELLING_RATE	5	Cash Selling Rate

CryptoTransactionType

Name	ID	Descriptor
DEPOSIT	1	Deposit
WITHDRAW	2	Withdraw
SATOSHI	3	Satoshi Test
PAYMENT	4	Payment (Currently not in use)
SETTLEMENT	5	Settlement (Currently not in use)
TRANSFER_IN	6	Transfer In
TRANSFER_OUT	7	Transfer Out
CARD_FEE_DEBIT	8	Card Fee Debit
CARD_FEE_CREDIT	9	Card Fee Credit
CARD_FEE_REFUND	10	Card Fee Refund

CryptoTransactionState

Name	ID	Descriptor
PENDING	0	Pending Approval
AUTHORIZED	101	Authorized
RISK_WITHHELD	102	Risk Withheld
PROCESSING	110	Processing
COMPLETED	200	Completed
REFUNDED	401	Refunded
REJECTED	900	Rejected
CLOSED	990	Closed

FiatTransactionType

Name	ID	Descriptor
DEPOSIT	1	Deposit
WITHDRAW	2	Withdraw
INVOICE_PAYMENT	3	Invoice Payment
TOP_UP	4	Top-up
TRANSFER_IN	5	Transfer In
TRANSFER_OUT	6	Transfer Out
BOUNCE_BACK	7	Refund
QR_PAYMENT	8	QR Payment
CARD_FEE_DEBIT	9	Card Fee Debit
CARD_FEE_CREDIT	10	Card Fee Credit
CARD_FEE_REFUND	11	Card Fee Refund

FiatTransactionState

Name	ID	Descriptor
PENDING	0	Pending Approval
AUTHORIZED	101	Authorized
PROCESSING	110	Processing
COMPLETED	200	Completed
REJECTED	900	Rejected
CLOSED	990	Closed

PaymentTransactionState

Name	ID	Descriptor
PENDING	0	Pending
AUTHORIZED	101	Authorized
RISK_WITHHELD	102	Risk Withheld
PARTIAL_PAYMENT	103	Partial Payment
CAPTURED	221	Captured
SUCCESS	200	Success
REFUNDED	401	Refunded
REVERSED	301	Reversed
CANCELLED	302	Cancelled
DENIED	900	Denied
EXPIRED	990	Expired

OrderStatus

Name	ID	Descriptor
PAID	1	Order Paid
UNPAID	2	Order Unpaid
EXPIRED	3	Order Expired
UNLIMITED	4	Order Unlimited Use
OVERPAID	5	Order Overpaid
CANCELLED	99	Order Cancelled

CommissionStatus

Name	ID	Descriptor
PENDING	1	Pending Reconciliation
RECONCILED	2	Reconciled
PAID	3	Paid
CANCELLED	99	Cancelled

AdjustmentStatus

Name	ID	Descriptor
PENDING	0	Pending Approval
APPROVED	1	Approved
SETTLED	2	Settled
CANCELLED	99	Cancelled

GeoAddressType

Name	ID	Descriptor
RESIDENTIAL	1	Residential Address
BILLING	2	Billing Address
SHIPPING	3	Shipping Address
BUSINESS	4	Business Address

OtcStatus

Name	ID	Descriptor
INITIAL	1	OTC Order Initial
COMPLETED	5	Order Completed
EXPIRED	10	Rate Expired
CANCELLED	11	Order Cancelled

OtcType

Name	ID	Descriptor
SELLING	1	Selling Token
BUYING	2	Buying Token

MainNet

Name	ID	Descriptor
BTC	1	Bitcoin
ERC20	3	Ethereum
TRC20	6	Tron
BSC	7	BSC
BASE	8	Base
SOLANA	9	Solana

EkycReviewStatus

Name	ID	Description
INIT	0	Strat procedure
PASSPORT_REJECTED	1	Passport rejected
PASSPORT_PASSED	2	Passport passed
SELFIE_REJECTED	3	Selfie rejected
SELFIE_PASSED	4	Selfie passed
POA_REJECTED	5	Proof of address rejected
POA_PASSED	6	Proof of address passed
SOF_REJECTED	7	Source of funds rejected
SOF_PASSED	8	Source of funds passed
VISA_REJECTED	9	Visa rejected
VISA_PASSED	10	Visa passed
RAA_PASSED	11	Risk Awareness Assessment passed
UNDER_REVIEW	12	KYC under review
ADR_REQUIRED	13	Additional document(s) required
PASSED	88	KYC passed
REJECTED	99	KYC rejected

eKYCFileVerifyStatus

Name	ID	Description
UNVERIFIED	1	Unverified
VERIFYING	2	Verifying

Name	ID	Description
VERIFY_SUCCESS	3	Verify Success
VERIFY_FAILURE	4	Verify Failure

RemitInfoType

ID	Descriptor
1	Client's own
2	Client's payee

BankType

ID	Description
1	SWIFT
2	ACH
3	FPS
4	SEPA
5	HK_FPS

BeneficiaryType

ID	Description
1	Individual
2	Corporate

Relationship

Name	ID	Description
PARENT	1	Parent
CHILD	2	Child
SPOUSE	3	Spouse
BUYER	4	Buyer
SELLER	5	Seller
SELF	6	Self
SIBLING	7	Sibling
EMPLOYEE	8	Employee
FRIEND	9	Friend
IMPORTER_EXPORTER	10	Importer/Exporter
CONSULTANT	11	Consultant
CONTRACTOR	12	Contractor

Available Currency

Currency	Description
SGD	Singapore Dollar
USD	United States Dollar
GBP	Pound sterling
EUR	Euro
HKD	Hong Kong dollar
JPY	Japanese yen
AUD	Australian dollar
CAD	Canadian dollar
CNH	Chinese yuan HK
MYR	Malaysian ringgit
AED	United Arab Emirates dirham
ETH	Ether (Ethereum), will be deprecated on 01 Dec 2024
BTC	Bitcoin, will be deprecated on 01 Dec 2024
USDT	USD Tether
USDC	USD Coin
WUSD	Worldwide USD
FDUSD	First Digital USD

CardTransactionStatus

ID	Description
0	Pending
200	Success
101	Authorized
102	Risk Withheld
301	Reversed
900	Denied
401	Refunded
221	Captured
990	Expired
302	Cancelled

CardTransactionType

ID	Description
1	Top-up

ID	Description
2	Purchase
3	Purchase reversal
4	Mismatch debit
5	Mismatch reversal
6	No auth settlement
99	Others

CardTransactionSubType

Name	ID	Descriptor
AUTO_TOP_UP	1	Auto Top-up
AUTO_DEBIT	2	Auto Debit
FIRST_TOP_UP_CASHBACK	3	First Top-up Cashback

PayoutMethod

Name	ID	Descriptor
PAY_NOW_WITHDRAW	1	Paynow Withdraw
CARD_TOP_UP	2	Card Topup
SGQR WITHDRAW	3	SGQR withdraw

PaymentMethodType

Name	ID	Descriptor
PAYNOW	4	Paynow
NTES	5	Nets

ProxyType

Name	ID	Descriptor
MOBILE	1	PayNow/ Mobile
UEN	2	PayNow/ UEN
NRIC	3	PayNow/ NRIC/FIN
NETS	4	SGQR/NETS

SwapOrderType

Name	ID	Descriptor
SWAP_AND_SEND	0	Swap and Send
DEPOSIT_AND_SEND	1	Deposit and Send
DEPOSIT_AND_SWAP	2	Deposit and Swap

Name	ID	Descriptor
SWAP_AND_TOPUP_CARD	3	Swap and Topup Card
DEPOSIT_AND_TOP_UP_WALLET	4	Deposit and Top up Wallet

SwapOrderState

Name	ID	Descriptor
WAITING_PAYIN	0	Waiting Payin
PAYOUT_PROCESSING	1	Payout Processing
COMPLETED	2	Completed
COMPLETED_WITH_EXCESS_AMOUNT	3	Completed With Excess Amount
FAILED	4	Failed
EXPIRED	5	Expired
CANCELLED	6	Cancelled

RiskLevel

Name	ID	Descriptor
LOW	10	Low Risk Account
MEDIUM	20	Medium Risk Account
HIGH	30	High Risk Account

WalletAddressType

Name	ID	Descriptor
CLIENT_OWN	1	Client Owned Address
DTC_CLIENT_WALLET	2	DTC Assigned Client Address

FileDocumentType

Name	ID	Descriptor
FIAT_DEPOSIT_RECEIPT	200	Fiat Deposit Receipt
INVOICE	201	Invoice

FileStatus

Name	ID	Descriptor
UPLOAD_FAILED	2	Upload Failed
NORMAL	3	Normal
DELETED	5	Deleted

ClientStatus

Name	ID	Descriptor
SUSPENDED	0	Suspended
PENDING_KYC	1	Pending KYC
DORMANT	2	Dormant
REGISTERED	3	Self Registered
REJECTED	4	Rejected
OFF_BOARD	5	Off Board
REFERRER	8	Referrer
TERMINATED	9	Terminated
DEACTIVATED	10	Deactivated
RESTRICTED	11	Restricted
DELETED	12	Deleted
FROZEN	13	Frozen
DROP	14	Drop
ACTIVATED	99	Activated

ActivityType

Name	ID	Descriptor
INIT	0	Data Init
PAYMENT	1	Payment Settlement
REMIT	2	Remittance
RESERVE	3	Reserve
OTC	4	Over-the-Counter
FIAT_WITHDRAWAL	5	Fiat Withdrawal
FIAT_DEPOSIT	6	Fiat Deposit
OTC_COMMISSION	7	SWAP Commission
POBO	8	Payment On Behalf Of
CRYPTO_WITHDRAWAL	9	Stablecoin Withdrawal
CRYPTO_DEPOSIT	10	Stablecoin Deposit
ADJUSTMENT	11	Adjustment
OTC_BONUS	12	SWAP Bonus
DTC_WALLET	13	DTC Wallet Payment
CRYPTO_PAYMENT_REFUND	14	Stablecoin Payment Refund
TOP_UP_CARD	15	Top Up Card
SATOSHI_TEST	16	Satoshi Test

Name	ID	Descriptor
CRYPTO_WITHHELD_REFUND	17	Stablecoin Withheld Refund
PURCHASE_CRYPTOC	18	Purchase Stablecoin
TOP_UP_WALLET	19	Top Up Wallet
CARD_PAYMENT_REFUND	20	Card Payment Refund
TRANSFER_IN	21	Transfer In
TRANSFER_OUT	22	Transfer Out
SWAP_CANCEL	23	Swap Cancel
BOUNCE_BACK	24	Refund
SCAN_PAY	28	Scan Pay
CARD_TRANSACTION_REVERSAL	29	Card Transaction Reversal
CARD_APPLICATION_FEE_DEBIT	30	Application Fee Debit
CARD_DELIVERY_FEE_DEBIT	31	Delivery Fee Debit
CARD_APPLICATION_FEE_CREDIT	32	Application Fee Credit
CARD_DELIVERY_FEE_CREDIT	33	Delivery Fee Credit
CARD_APPLICATION_FEE_REFUND	34	Application Fee Refund
CARD_DELIVERY_FEE_REFUND	35	Delivery Fee Refund

WalletStatus

Name	ID	Descriptor
PENDING	0	Pending
ACTIVE	1	Active
SUSPENDED	2	Suspended
FROZEN	3	Frozen

TransferPurpose

Name	ID	Descriptor
ACCOUNT_TOP_UP	0	Account Top Up
ADVANCE_PAYMENT	1	Advance Payment
ALIMONY_PAYMENT	2	Alimony Payment
CARPARK_CHARGES	3	Carpark Charges
COLLECTION_PAYMENT	4	Collection Payment
CREDIT_CARD_PAYMENT	5	Credit Card Payment
EDUCATION	6	Education
GOVERNMENT_INSURANCE	7	Government Insurance

Name	ID	Descriptor
HEALTHCARE_SERVICES	8	Healthcare Services
INSURANCE_PREMIUM	9	Insurance Premium
INTEREST	10	Interest
INVESTMENT	11	Investment
INVESTMENT_SECURITIES	12	Investment Securities
LOAN_REPAYMENT	13	Loan Repayment
PAYMENT_FOR_FEE_AND_CHARGES	14	Payment for fee and Charges
PROPERTY_INSURANCE	15	Property Insurance
PROPERTY_TAX	16	Property Tax
REBATE	17	Rebate
RECURRING_INSTALLMENT_PAYMENT	18	Recurring Installment Payment
RENT	19	Rent
ROAD_TAX	20	Road Tax
SAVINGS	21	Savings
TELECO_BILL	22	Telco Bill
TRANSPORT_PAYMENT	23	Transport Payment
UTILITIES_PAYMENT	24	Utilities Payment
BONUS_PAYMENT	25	Bonus Payment
BUSINESS_EXPENSES	26	Business Expenses
CAPITAL_INJECTION	27	Capital Injection
CASH_DISBURSEMENT	28	Cash Disbursement
COMMISSION	29	Commission
DIVIDEND	30	Dividend
FOREIGN_WORKER_LEVY	31	Foreign Worker Levy
INVOICE_PAYMENT	32	Invoice Payment
LICENSE_FEE	33	License Fee
REFUND	34	Refund
SALARY_PAYMENT	35	Salary Payment
SUBSCRIPTION	36	Subscription
SUPPLIER_PAYMENT	37	Supplier Payment
TAX_PAYMENT	38	Tax Payment
OTHERS	99	Others