Commercial Bank OpenAPI Documentation for Enterprise Users v1.1.0

This document provides an overview of the functional interfaces for enterprise users integrating with the Commercial Bank's Control Panel, enabling the seamless incorporation of Project Kissen functionalities into existing business processes or software.

Introduction

Communication Mode

1. All APIs outlined in this document require HTTPS requests (POST mode) with UTF-8 encoded characters.
2. Outputs are delivered in JSON format.
3. Each request must be signed by the Enterprise. Responses from the Commercial Bank System are verifiable by the Enterprise.

Definition of Terms

1. **accessKey**: A credential issued by Commercial Bank System Operator to the Enterprise for accessing OpenAPIs.
2. **Bus-Trace-ID**: The unique identifier provided by the Enterprise when calling OpenAPIs and sending requests to the System. It consists of 32 digits or letters, or a combination of both. If the Enterprise calls the same interface multiple times with the same Bus-Trace-ID, only the first request will be processed.
3. **transCode**: The unique identifier provided by the Enterprise when calling OpenAPIs and sending requests to the System for parameter signing. It consists of 32 digits or letters, or a combination of both.
4. **reqTime**: The timestamp (13-digit decimal, millisecond precision) at which the Enterprise sends a request. The System will verify timestamp and reject requests if the time difference between the current System time and the request time exceeds 5 minutes.
5. **Public and private key pairs for signing**: The Enterprise must securely store the private key used for signing requests, while the public key is used by the System for verification.
6. **Public and private key pairs for data encryption/decryption**: When Enterprises making OpenAPI calls, certain sensitive data will be encrypted using the public key. Upon receiving the request, the System will use the corresponding data decryption private key of the Enterprise to decrypt the data.
7. **respTime**: The timestamp of the System's response (13-digit decimal, millisecond precision).

Data Signing and Signature Verification Rules

1. All requests in the OpenAPI are signed and verified by RSA asymmetric encryption.
2. To sign a request, the Enterprise concatenates the values of accessKey, transCode, and reqTime into strings in sequence, converts them to lowercase, and signs them with the private key.
3. In the signature verification of the response information, the original data for the signature is formed by concatenating the values of the request accessKey, the transCode in the response body, the response code, and the respTime in sequence. The concatenated values are then converted to lowercase and signed using the System's private key. The signed data is returned to the Enterprise and requires verification using the provided public key.

Interface Specification

Request Rules

1. Configure the HTTP/HTTPS request header to include `key = accessKey`, where `value` is the access key provided by the Commercial Bank System.
2. Configure the HTTP/HTTPS request header to include `key = Bus-Trace-ID`, where value is the unique identifier for the request, consisting of 32 digits or letters, or a combination of both.
3. Below is an overview of the requests and responses.

Format of the Request Message

Parameter name	Description	Data type	Required
transCode	Unique code of the request	String	true
reqTime	The timestamp (13-digit decimal, millisecond precision) at which the Service Provider makes a request.	Long	true
body	Request body (defined by the specific business)	Object	true
mac	Signature	String	true

Sample Request:

```
{
  "body": {},
  "mac":
"V8mGaMKEcKuqKDMX9VsVOY4HgNZLgF0zhuW8nyLLZFfYvpBwxK8wWMuQqJ2XmSZcUhpyKwvKXfSw8PD
OnYTfsmnz8XEMTZIOpinXhTALobddT4xlMvguMctFpE6W4O4Kzi45kmxwViDKXvJWsPhts7alnZLS+Ju
CB7rzSYd/rjA=",
  "reqTime": 1712739465790
}
```

Format of Response Message

Parameter name	Description	Data type	Required
code	Response code	String	true
transCode	Unique code of the request	String	true
respTime	The timestamp (13-digit decimal, millisecond precision) at which the System makes a response.	Long	true
body	Request body (as defined by the specific business)	Object	true
mac	Signature	String	true

Sample Response:

```
{
  "code": "MSG_90_0000",
  "respTime": 1712740004308,
  "transCode": "37e8117701164bcfb8e11f099d201d80",
  "body": {
    "fromWalletAddress": "",
    "signedMessage": "",
    "toWalletAddress": "",
    "tokenAmount": 0,
    "tokenCode": "",
    "txHash": ""
  },
  "mac":
"Thu7/X5yczVG0eHG0ARML8BULJv4vHLcVoyziLn9JxKDOqO3UWRgXAz+RN/yF7592uH7qXGmr1mME7q
0NXsmXkPT/Akt2ETVv1JAFf+nH1ZLexp8LAzW06P9kxDc7XBHuQZ7h4225KH0sgRj4Hrkaoqk4OdYFPr
2D2FPAfgW/20="
}
```

Response Status Code

Response Code	Description
MSG_90_0000	Success
MSG_90_0001	Access key cannot be empty
MSG_90_0002	Request format is incorrect
MSG_90_0003	Request time cannot be empty
MSG_90_0004	Transaction identifier cannot be empty
MSG_90_0005	MAC cannot be empty
MSG_90_0006	Invalid access key
MSG_90_0007	Signature verification failed
MSG_90_0008	Bus-Trace-ID cannot be empty
MSG_90_0009	Invalid request
MSG_90_0010	System exception
MSG_90_0011	Transaction identifier length is incorrect
MSG_90_0012	Transaction identifier format is incorrect; it must consist of letters, numbers, or a combination.
MSG_90_0013	System is busy. Please try again later
MSG_90_0014	Parameter format error; conversion failed

Response Code	Description
MSG_90_0016	Bus-Trace-ID length is incorrect
MSG_90_0017	Bus-Trace-ID format is incorrect; it must consist of letters, numbers, or a combination.
MSG_90_0018	Invalid wallet address
MSG_90_0019	Tx Hash cannot be empty
MSG_90_0020	Custodian Bank or Token information does not exist
MSG_90_0021	Contract does not exist
MSG_90_0022	Commercial Bank does not exist
MSG_90_1000	Token currency information cannot be empty
MSG_90_1001	Token currency information does not exist or is unavailable
MSG_90_1002	Wallet address is too long; it cannot exceed 100 characters
MSG_90_1003	Wallet address cannot be empty
MSG_90_1004	Wallet name is too long; it cannot exceed 100 characters
MSG_90_1005	Wallet address already exists
MSG_90_1007	Wallet address does not exist
MSG_90_1008	Wallet public key cannot be empty
MSG_90_1009	Wallet public key verification failed; please check the public key format
MSG_90_1010	Wallet public key and wallet address do not match
MSG_90_1011	Wallet is being created
MSG_90_1012	Wallet has been created
MSG_90_1013	Invalid wallet address
MSG_90_1014	Master wallet address does not exist
MSG_90_1015	Invalid master wallet address
MSG_90_1016	Creating wallet exceeds the limit for sub-wallets
MSG_90_1017	This enterprise does not have a master wallet; sub-wallets cannot be created
MSG_90_2010	From wallet address cannot be empty
MSG_90_2011	To wallet address cannot be empty
MSG_90_2012	Transaction amount cannot be empty

Response Code	Description
MSG_90_2013	Token cannot be empty
MSG_90_2014	Token code cannot be empty
MSG_90_2015	Transaction signature data cannot be empty
MSG_90_2016	Token access information does not exist
MSG_90_2017	Custodian Bank does not exist
MSG_90_2018	Token amount must be greater than 0
MSG_90_2019	Decimal precision exceeds limit
MSG_90_2020	Wallet address does not exist
MSG_90_2022	Contract pre-execution failed
MSG_90_2023	Contract packaging transaction failed
MSG_90_2024	Transaction does not exist
MSG_90_2025	Withdrawal amount must be greater than 0
MSG_90_2026	Top-up/withdrawal order information does not exist
MSG_90_2027	Number of Tokens topped up must be an integer
MSG_90_2028	Top-up/withdrawal order information already exists
MSG_90_2029	Insufficient balance
MSG_90_2030	This wallet has ongoing transactions
MSG_90_2031	Liquidity pool does not exist
MSG_90_2032	No exchange rate available
MSG_90_2033	Account authorization record does not exist
MSG_90_2034	Transaction information does not exist or permission denied to query
MSG_90_2035	Exchange rate type cannot be empty
MSG_90_2036	Bank BIC cannot be empty
MSG_90_2037	Liquidity provider does not exist

Interface List

Token Transfer Module

Transfer - Create Transaction

url: `/api/openapi/v1/transaction/transfer/createTransaction`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Transfer - Create Transaction" interface to compile the data into the transaction body, including the Token code, from wallet address, to wallet address, and transfer amount.

Step 2: The user uses the from wallet's private key to sign the transaction.

Step 3: The user calls the "Transfer - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Transfer - Query Transaction Result" interface to query the transfer results.

Request Example:

```
{
  "body": {
    "fromWalletAddress": "",
    "toWalletAddress": "",
    "tokenAmount": 0,
    "tokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfCreateTransferTransactionReqVO	ApiReqInfoOfCreateTransferTransactionReqVO	body	true	ApiReqInfoOfCreateTransferTransactionReqVO	ApiReqInfoOfCreateTransferTransactionReqVO
body			false	CreateTransferTransactionReqVO	CreateTransferTransactionReqVO
fromWalletAddress	From wallet address		true	string	
toWalletAddress	To wallet address		true	string	
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCreateTransferTransactionRespVO

Response Params:

Name	Description	Type	Schema
body		CreateTransferTransactionRespVO	CreateTransferTransactionRespVO
chainId	Blockchain ID	string	
rawTransaction	Raw transaction	RawTransactionRespVO	RawTransactionRespVO
data		string	
feeCap		number	
gasLimit		number	
gasPremium		number	
gasPrice		number	
nonce		number	
to		string	
value		number	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "chainId": "",
        "rawTransaction": {
            "data": "",
            "feeCap": 0,
            "gasLimit": 0,
            "gasPremium": 0,
            "gasPrice": 0,
            "nonce": 0,
            "to": "",
            "value": 0
        }
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Transfer - Send Transaction

url: `/api/openapi/v1/transaction/transfer/sendTransaction`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Transfer - Create Transaction" interface to compile the data into the transaction body, including the Token code, from wallet address, to wallet address, and transfer amount.

Step 2: The user uses the from wallet's private key to sign the transaction.

Step 3: The user calls the "Transfer - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Transfer - Query Transaction Result" interface to query the transfer results.

Request Example:

```
{
  "body": {
    "fromWalletAddress": "",
    "signedMessage": "",
    "toWalletAddress": "",
    "tokenAmount": 0,
    "tokenCode": "",
    "txHash": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfSendTransferTransactionReqVO	ApiReqInfoOfSendTransferTransactionReqVO	body	true	ApiReqInfoOfSendTransferTransactionReqVO	ApiReqInfoOfSendTransferTransactionReqVO
body			false	SendTransferTransactionReqVO	SendTransferTransactionReqVO
fromWalletAddress	From wallet address		true	string	
signedMessage	Signed message		true	string	
toWalletAddress	To wallet address		true	string	
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
txHash	Tx Hash		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfSendTransferTransactionRespVO

Response Params:

Name	Description	Type	Schema
body		SendTransferTransactionRespVO	SendTransferTransactionRespVO
txHash	Tx Hash	string	
code		string	
mac		string	

Name	Description	Type	Schema
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "txHash": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Transfer - Query Transaction Result

url: `/api/openapi/v1/transaction/transfer/queryResult`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Transfer - Create Transaction" interface to compile the data into the transaction body, including the Token code, from wallet address, to wallet address, and transfer amount.

Step 2: The user uses the from wallet's private key to sign the transaction.

Step 3: The user calls the "Transfer - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Transfer - Query Transaction Result" interface to query the transfer results.

Request Example:

```
{
  "body": {
    "txHash": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema

Name	Description	In	Required	Type	Schema
apiReqInfoOfQueryTransactionResultReqVO	apiReqInfoOfQueryTransactionResultReqVO	body	true	ApiReqInfoOfQueryTransactionResultReqVO	ApiReqInfoOfQueryTransactionResultReqVO
body			false	QueryTransactionResultReqVO	QueryTransactionResultReqVO
txHash	Tx Hash		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfQueryTransactionResultRespVO

Response Params:

Name	Description	Type	Schema
body		QueryTransactionResultRespVO	QueryTransactionResultRespVO
fromWalletAddress	From wallet address	string	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
toWalletAddress	To wallet address	string	
tokenAmount	Token amount	number(bigdecimal)	
transCode	Transaction code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "fromWalletAddress": "",
        "status": 0,
        "toWalletAddress": "",
        "tokenAmount": 0,
        "transCode": "",
        "txHash": "",
        "txTime": 0
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Authorization - Create Transaction

url: `/api/openapi/v1/transaction/authorized/createTransaction`

method: `POST`

produces: `application/json`

consumes: `*/*`

Note:

For a Token swap using the local Commercial Bank's exchange rate, enterprise users are not required to request authorization.
For a Token swap using the platform's exchange rate, enterprise users are required to request authorization. The detailed steps are as follows:

Step 1: The user calls the "Authorization - Create Transaction" interface to compile the data into the transaction body, including the Token code, from wallet address, to wallet address, and transfer amount.

Step 2: The user uses the private key to sign the transaction.

Step 3: The user calls the "Authorization - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Authorization - Query Transaction Result" interface to query the authorization results.

Upon successful authorization, the enterprise users can submit Token swap transaction.

Request Example:

```
{
  "body": {
    "fromWalletAddress": "",
    "toWalletAddress": "",
    "tokenAmount": 0,
    "tokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfCreateAuthorizedTransactionReqVO	ApiReqInfoOfCreateAuthorizedTransactionReqVO
body			false	CreateAuthorizedTransactionReqVO	CreateAuthorizedTransactionReqVO
fromWalletAddress	From wallet address		true	string	
toWalletAddress	To wallet address		false	string	
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCreateTransactionRespVO

Response Params:

Name	Description	Type	Schema
body		CreateTransactionRespVO	CreateTransactionRespVO
chainId	Blockchain ID	string	
rawTransaction	Raw transaction	RawTransactionRespVO	RawTransactionRespVO
data		string	
feeCap		integer	
gasLimit		integer	
gasPremium		integer	
gasPrice		integer	
nonce		integer	
to		string	
value		integer	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response example:

```
{
    "body": {
        "chainId": "",
        "rawTransaction": {
            "data": "",
            "feeCap": 0,
            "gasLimit": 0,
            "gasPremium": 0,
            "gasPrice": 0,
            "nonce": 0,
            "to": "",
            "value": 0
        }
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Authorization - Send Transaction

url: `/api/openapi/v1/transaction/authorized/sendTransaction`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

Step 1: The user calls the "Authorization - Create Transaction" interface to compile the data into the transaction body, including the Token code, from wallet address, to wallet address, and transfer amount.

Step 2: The user uses the private key to sign the transaction.

Step 3: The user calls the "Authorization - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Authorization - Query Transaction Result" interface to query the authorization results.

Request Example:

```
{
  "body": {
    "fromWalletAddress": "",
    "signedMessage": "",
    "toWalletAddress": "",
    "tokenAmount": 0,
    "tokenCode": "",
    "txHash": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfSendAuthorizedTransactionReqVO	ApiReqInfoOfSendAuthorizedTransactionReqVO
body			false	SendAuthorizedTransactionReqVO	SendAuthorizedTransactionReqVO
fromWalletAddress	From wallet address		true	string	
signedMessage	Signed message		true	string	
toWalletAddress	To wallet address		false	string	
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
txHash	Tx Hash		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfSendTransactionRespVO

Response Params:

Name	Description	Type	Schema
body		SendTransactionRespVO	SendTransactionRespVO
txHash	Tx Hash	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "txHash": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Authorization - Query Transaction Result

url: `/api/openapi/v1/transaction/authorized/queryResult`

method: `POST`

produces: `application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Authorization - Create Transaction" interface to compile the data into the transaction body, including the Token code, from wallet address, to wallet address, and transfer amount.

Step 2: The user uses the private key to sign the transaction.

Step 3: The user calls the "Authorization - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Authorization - Query Transaction Result" interface to query the authorization results.

Request Example:

```
{
  "body": {
    "txHash": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfQueryTransactionResultReqVO	ApiReqInfoOfQueryTransactionResultReqVO
body			false	QueryTransactionResultReqVO	QueryTransactionResultReqVO
txHash	Tx Hash		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfQueryApproveResultRespVO

Response Params:

Name	Description	Type	Schema

Name	Description	Type	Schema
body		QueryApproveResultRespVO	QueryApproveResultRespVO
count	Count	number	
fromWalletAddress	From wallet address	string	
status	Status 20: Pending submission 30: Submitting 35: Submitted 40: Submission failed	integer(int32)	
toWalletAddress	To wallet address	string	
transCode	Trans code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "count": 0,
        "fromWalletAddress": "",
        "status": 0,
        "toWalletAddress": "",
        "transCode": "",
        "txHash": "",
        "txTime": 0
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Common Module

Query Token Information

url: `/api/openapi/v1/common/token`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to retrieve information about the available Token for enterprise access.

Request Example:

```
{
  "body": {},
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfo	ApiReqInfo	body	true	ApiReqInfo	ApiReqInfo
body			false	object	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfListOfSpAccessTdRespVO

Response Params:

Name	Description	Type	Schema
body		array	SpAccessTdRespVO
blockChainId	Blockchain ID	string	
blockChainName	Blockchain name	string	
blockChainVersion	Blockchain version	string	
currencySymbol	Currency symbol	string	
tokenCode	Token code	string	
tokenName	Token name	string	
tokenPrice	Token price	number(bigdecimal)	

Name	Description	Type	Schema
tokenSymbol	Token symbol	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": [
        {
            "blockChainId": "",
            "blockChainName": "",
            "blockChainVersion": "",
            "currencySymbol": "",
            "tokenCode": "",
            "tokenName": "",
            "tokenPrice": 0,
            "tokenSymbol": ""
        }
    ],
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Update Access Key

url: `/api/openapi/v1/common/access/key/update`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to update the access key for accessing the Project Kissen. After the update, all new requests should be sent with the latest access key.

Example:

```json
{
  "body": {},
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfo	ApiReqInfo	body	true	ApiReqInfo	ApiReqInfo
body			false	object	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Request Status:

Code	Description	Schema
200	OK	ApiRespInfoOfRefreshAccessKeyRespVO

Response Params:

Name	Description	Type	Schema
body		RefreshAccessKeyRespVO	RefreshAccessKeyRespVO
accessKey	Access key	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```json
{
    "body": {
        "accessKey": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Transfer-Enabled Commercial Banks by Token

url: `/api/openapi/v1/common/getReceiveBankInfoByToken`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to retrieve a list of Commercial Banks that accept transfers based on a specified Token.

Request Example:

```
{
  "body": {
    "tokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqVO	reqVO	body	true	ApiReqInfoOfReceiveBankByTokenReqVO	ApiReqInfoOfReceiveBankByTokenReqVO
body			false	ReceiveBankByTokenReqVO	ReceiveBankByTokenReqVO
tokenCode	Token code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfListOfReceiveBankByTokenRespVO

Response Params:

Name	Description	Type	Schema
body		array	ReceiveBankByTokenRespVO
bankBic	Bank BIC	string	
bankName	Bank name	string	
tokenInfoRespVOList		array	TokenInfoRespVO
tokenCode	Token code	string	

Name	Description	Type	Schema
tokenName	Token name	string	
tokenSymbol	Token symbol	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": [
        {
            "bankBic": "",
            "bankName": "",
            "tokenInfoRespVOList": [
                {
                    "tokenCode": "",
                    "tokenName": "",
                    "tokenSymbol": ""
                }
            ]
        }
    ],
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Wallet Address List by Commercial Bank

url: `/api/openapi/v1/common/getWalletListByBank`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to retrieve a list of wallet address associated with a specified Commercial Bank.

Request Example:

```
{
  "body": {
    "bankBic": "",
    "tokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqVO	reqVO	body	true	ApiReqInfoOfWalletListByBankReqVO	ApiReqInfoOfWalletListByBankReqVO
body			false	WalletListByBankReqVO	WalletListByBankReqVO
bankBic	Bank BIC		true	string	
tokenCode	Token Code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfListOfWalletListByBankRespVO

Response Params:

Name	Description	Type	Schema
body		array	WalletListByBankRespVO
enterpriseCode	Enterprise code	string	
enterpriseName	Enterprise name	string	
tokenAmount	Token amount	number	
walletAddress	Wallet address	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": [
        {
            "enterpriseCode": "",
            "enterpriseName": "",
            "tokenAmount": 0,
            "walletAddress": ""
        }
    ],
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Top-up and Withdrawal Module

Token Top-up - Submit Request

url: `/api/openapi/v1/token/buy`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to submit Token top-up request. It compiles the data into the transaction body, including the Token code, wallet address and purchase amount.

Request Example:

```
{
  "body": {
    "tokenAmount": 0,
    "tokenCode": "",
    "walletAddress": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfCbdcBuyReqVO	ApiReqInfoOfCbdcBuyReqVO	body	true	ApiReqInfoOfCbdcBuyReqVO	ApiReqInfoOfCbdcBuyReqVO
body			false	CbdcBuyReqVO	CbdcBuyReqVO
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
walletAddress	Wallet address		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCbdcBuyRespVO

Response Params:

Name	Description	Type	Schema
body		CbdcBuyRespVO	CbdcBuyRespVO
orderNumber	Order number	string	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenAmount	Token amount	number(bigdecimal)	
tokenCode	Token code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
walletAddress	Wallet address	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "orderNumber": "",
        "status": 0,
        "tokenAmount": 0,
        "tokenCode": "",
        "txHash": "",
        "txTime": 0,
        "walletAddress": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Top-up - Query Request Result

url: `/api/openapi/v1/token/buy/result`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is used to query the results of a Token top-up request.

Request Example:

```
{
  "body": {
    "orderNumber": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfCbdcBuyResultReqVO	ApiReqInfoOfCbdcBuyResultReqVO	body	true	ApiReqInfoOfCbdcBuyResultReqVO	ApiReqInfoOfCbdcBuyResultReqVO
body			false	CbdcBuyResultReqVO	CbdcBuyResultReqVO
orderNumber	Order number		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCbdcBuyRespVO

Response Params:

Name	Description	Type	Schema
body		CbdcBuyRespVO	CbdcBuyRespVO
orderNumber	Order number	string	

Name	Description	Type	Schema
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenAmount	Token amount	number(bigdecimal)	
tokenCode	Token code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
walletAddress	Wallet address	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "orderNumber": "",
        "status": 0,
        "tokenAmount": 0,
        "tokenCode": "",
        "txHash": "",
        "txTime": 0,
        "walletAddress": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Token Top-up/Withdrawal Request Details

url: `/api/openapi/v1/token/detail`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to query the details of a Token purchase or Token withdrawal request.

Request Example:

```
{
  "body": {
    "orderNumber": "",
    "tokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfCbdcOrderDetailReqVO	ApiReqInfoOfCbdcOrderDetailReqVO	body	true	ApiReqInfoOfCbdcOrderDetailReqVO	ApiReqInfoOfCbdcOrderDetailReqVO
body			false	CbdcOrderDetailReqVO	CbdcOrderDetailReqVO
orderNumber	Order number		true	string	
tokenCode	Token code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCbdcOrderDetailRespVO

Response Params:

Name	Description	Type	Schema
body		CbdcOrderDetailRespVO	CbdcOrderDetailRespVO
createTime	Created on	integer(int64)	
orderNumber	Order number	string	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenAmount	Token amount	number(bigdecimal)	

Name	Description	Type	Schema
tokenCode	Token code	string	
tokenName	Token name	string	
transCode	Transaction code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
txType	Transaction Type 3: Purchase 4: Withdrawal	integer(int32)	
walletAddress	Wallet address	string	
walletName	Wallet name	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "createTime": 0,
        "orderNumber": "",
        "status": 0,
        "tokenAmount": 0,
        "tokenCode": "",
        "tokenName": "",
        "transCode": "",
        "txHash": "",
        "txTime": 0,
        "txType": 0,
        "walletAddress": "",
        "walletName": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Token Top-up/Withdrawal Requests by Pagination

url: `/api/openapi/v1/token/query`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is used to query the details of Token top-up or Token withdrawal requests by pagination.

Request Example:

```json
{
  "body": {
    "data": {
      "endCreateTime": 0,
      "endTxTime": 0,
      "orderNumber": "",
      "startCreateTime": 0,
      "startTxTime": 0,
      "status": 0,
      "tokenCode": "",
      "txHash": "",
      "txType": 0,
      "walletAddress": ""
    },
    "page": {
      "pageNum": 0,
      "pageSize": 0
    }
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfDataTableOfCbdcOrderReqVO	ApiReqInfoOfDataTableOfCbdcOrderReqVO	body	true	ApiReqInfoOfDataTableOfCbdcOrderReqVO	ApiReqInfoOfDataTableOfCbdcOrderReqVO
body			false	DataTableOfCbdcOrderReqVO	DataTableOfCbdcOrderReqVO
data	Data		true	CbdcOrderReqVO	CbdcOrderReqVO
endCreateTime	End created on		false	integer	
endTxTime	Transaction end time		false	integer	
orderNumber	Order number		false	string	
startCreateTime	Start created on		false	integer	
startTxTime	Transaction start time		false	integer	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed		false	integer	
tokenCode	Token code		false	string	
txHash	Tx Hash		false	string	

Name	Description	In	Required	Type	Schema
txType	Transaction Type 3: Purchase 4: Withdrawal		false	integer	
walletAddress	Wallet address		false	string	
page	Page		true	PageInfo	PageInfo
pageNum	Page number		true	integer	
pageSize	Page size		true	integer	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfResultDataOfCbdcOrderRespVO

Response Params:

Name	Description	Type	Schema
body		ResultDataOfCbdcOrderRespVO	ResultDataOfCbdcOrderRespVO
page	Page	ResultPageInfo	ResultPageInfo
isFirstPage	First page	boolean	
isLastPage	Last page	boolean	
pageNum	Page num	integer	
pageSize	Page size	integer	
pages	Pages	integer	
total	Total	integer	
rows	Rows	array	CbdcOrderRespVO
createTime	Created on	integer	
orderNumber	Order number	string	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer	
tokenAmount	Token amount	number	
tokenCode	Token code	string	
tokenName	Token name	string	
txHash	Tx Hash	string	
txTime	Transaction time	integer	

Name	Description	Type	Schema
txType	Transaction Type 3: Purchase 4: Withdrawal	integer	
walletAddress	Wallet address	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
            "pageNum": 0,
            "pageSize": 0,
            "pages": 0,
            "total": 0
        },
        "rows": [
            {
                "createTime": 0,
                "orderNumber": "",
                "status": 0,
                "tokenAmount": 0,
                "tokenCode": "",
                "tokenName": "",
                "txHash": "",
                "txTime": 0,
                "txType": 0,
                "walletAddress": ""
            }
        ]
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Withdrawal - Create Request

url: `/api/openapi/v1/token/withdrawal/createTransaction`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Token Withdrawal - Create Request" interface to compile the data into the transaction body, including the Token code, wallet address and withdrawal amount.

Step 2: The user uses the from wallet's private key to sign the transaction and calculates the offline transaction hash.

Step 3: The user calls the "Token Withdrawal - Submit Request" interface to upload the transaction to the blockchain.

Step 4: The user calls "Token Withdrawal - Query Request Result" interface to query the withdrawal results.

Request Example:

```
{
  "body": {
    "tokenAmount": 0,
    "tokenCode": "",
    "walletAddress": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfCreateWithdrawReqVO	ApiReqInfoOfCreateWithdrawReqVO	body	true	ApiReqInfoOfCreateWithdrawReqVO	ApiReqInfoOfCreateWithdrawReqVO
body			false	CreateWithdrawReqVO	CreateWithdrawReqVO
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
walletAddress	Wallet address		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCreateWithdrawRespVO

Response Params:

Name	Description	Type	Schema
body		CreateWithdrawRespVO	CreateWithdrawRespVO
chainId	Blockchain ID	string	

Name	Description	Type	Schema
rawTransaction	Raw transaction	RawTransactionRespVO	RawTransactionRespVO
data		string	
feeCap		number	
gasLimit		number	
gasPremium		number	
gasPrice		number	
nonce		number	
to		string	
value		number	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```json
{
    "body": {
        "chainId": "",
        "rawTransaction": {
            "data": "",
            "feeCap": 0,
            "gasLimit": 0,
            "gasPremium": 0,
            "gasPrice": 0,
            "nonce": 0,
            "to": "",
            "value": 0
        }
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Withdrawal - Submit Request

url: `/api/openapi/v1/token/withdrawal/sendTransaction`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Token Withdrawal - Create Request" interface to compile the data into the transaction body, including the Token code, wallet address and withdrawal amount.

Step 2: The user uses the from wallet's private key to sign the transaction and calculates the offline transaction hash.

Step 3: The user calls the "Token Withdrawal - Submit Request" interface to upload the transaction to the blockchain.

Step 4: The user calls "Token Withdrawal - Query Request Result" interface to query the withdrawal results.

Example:

```
{
  "body": {
    "signedMessage": "",
    "tokenAmount": 0,
    "tokenCode": "",
    "txHash": "",
    "walletAddress": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfSendWithdrawReqVO	ApiReqInfoOfSendWithdrawReqVO	body	true	ApiReqInfoOfSendWithdrawReqVO	ApiReqInfoOfSendWithdrawReqVO
body			false	SendWithdrawReqVO	SendWithdrawReqVO
signedMessage	Signed message		true	string	
tokenAmount	Token amount		true	number	
tokenCode	Token code		true	string	
txHash	Tx Hash		true	string	
walletAddress	Wallet address		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCbdcWithdrawalRespVO

Response Params:

Name	Description	Type	Schema
body		CbdcWithdrawalRespVO	CbdcWithdrawalRespVO

Name	Description	Type	Schema
orderNumber	Order number	string	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenAmount	Token amount	number(bigdecimal)	
tokenCode	Token code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
walletAddress	Wallet address	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "orderNumber": "",
        "status": 0,
        "tokenAmount": 0,
        "tokenCode": "",
        "txHash": "",
        "txTime": 0,
        "walletAddress": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Withdrawal - Query Request Result

url: `/api/openapi/v1/token/withdrawal/result`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

Step 1: The user calls the "Token Withdrawal - Create Request" interface to compile the data into the transaction body, including the Token code, wallet address and withdrawal amount.

Step 2: The user uses the from wallet's private key to sign the transaction and calculates the offline transaction hash.

Step 3: The user calls the "Token Withdrawal - Submit Request" interface to upload the transaction to the blockchain.

Step 4: The user calls "Token Withdrawal - Query Request Result" interface to query the withdrawal results.

Request Example:

```
{
  "body": {
    "orderNumber": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfCbdcWithdrawalResultReqVO	ApiReqInfoOfCbdcWithdrawalResultReqVO	body	true	ApiReqInfoOfCbdcWithdrawalResultReqVO	ApiReqInfoOfCbdcWithdrawalResultReqVO
body			false	CbdcWithdrawalResultReqVO	CbdcWithdrawalResultReqVO
orderNumber	Order number		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfCbdcWithdrawalRespVO

Response Params:

Name	Description	Type	Schema
body		CbdcWithdrawalRespVO	CbdcWithdrawalRespVO

Name	Description	Type	Schema
orderNumber	Order number	string	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenAmount	Token amount	number(bigdecimal)	
tokenCode	Token code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
walletAddress	Wallet address	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "orderNumber": "",
        "status": 0,
        "tokenAmount": 0,
        "tokenCode": "",
        "txHash": "",
        "txTime": 0,
        "walletAddress": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Wallet Module

Register Wallet

url: `/api/openapi/v1/wallet/create`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to register a wallet by uploading the wallet address, wallet name, public key, and the associated Token.

Request Example:

```
{
  "body": {
    "tokenCode": "",
    "walletAddress": "",
    "walletName": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfWalletCreateReqVO	ApiReqInfoOfWalletCreateReqVO	body	true	ApiReqInfoOfWalletCreateReqVO	ApiReqInfoOfWalletCreateReqVO
body			false	WalletCreateReqVO	WalletCreateReqVO
tokenCode	Token code		true	string	
walletAddress	Wallet address		true	string	
walletName	Wallet name		false	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfWalletCreateResultRespVO

Response Params:

Name	Description	Type	Schema
body		WalletCreateResultRespVO	WalletCreateResultRespVO
createTime	Created on	integer(int64)	

Name	Description	Type	Schema
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenCode	Token code	string	
txHash	Tx Hash	string	
txTime	Transaction time	integer(int64)	
walletAddress	Wallet address	string	
walletName	Wallet name	string	
walletType	Wallet type	integer(int32)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "createTime": 0,
        "status": 0,
        "tokenCode": "",
        "txHash": "",
        "txTime": 0,
        "walletAddress": "",
        "walletName": "",
        "walletType": 0
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Wallet Registration Result

url: `/api/openapi/v1/wallet/create/result`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to check the wallet registration result.

Example:

```
{
  "body": {
    "tokenCode": "",
    "walletAddress": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfWalletCreateResultReqVO	ApiReqInfoOfWalletCreateResultReqVO	body	true	ApiReqInfoOfWalletCreateResultReqVO	ApiReqInfoOfWalletCreateResultReqVO
body			false	WalletCreateResultReqVO	WalletCreateResultReqVO
tokenCode	Token code		true	string	
walletAddress	Wallet address		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfWalletCreateResultRespVO

Response Params:

Name	Description	Type	Schema
body		WalletCreateResultRespVO	WalletCreateResultRespVO
createTime	Created on	integer(int64)	

Name	Description	Type	Schema
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenCode	Token code	string	
txHash	Tx Hash	string	
txTime	Transaction time	integer(int64)	
walletAddress	Wallet address	string	
walletName	Wallet name	string	
walletType	Wallet type	integer(int32)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "createTime": 0,
        "status": 0,
        "tokenCode": "",
        "txHash": "",
        "txTime": 0,
        "walletAddress": "",
        "walletName": "",
        "walletType": 0
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Wallet Details

url: `/api/openapi/v1/wallet/detail`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to query wallet details

Request Example:

```
{
  "body": {
    "tokenCode": "",
    "walletAddress": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfWalletCreateResultReqVO	ApiReqInfoOfWalletCreateResultReqVO	body	true	ApiReqInfoOfWalletCreateResultReqVO	ApiReqInfoOfWalletCreateResultReqVO
body			false	WalletCreateResultReqVO	WalletCreateResultReqVO
tokenCode	Token code		true	string	
walletAddress	Wallet address		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfWalletQueryRespVO

Response Params:

Name	Description	Type	Schema
body		WalletQueryRespVO	WalletQueryRespVO
balance	Wallet balance	number(bigdecimal)	
createTime	Created on	integer(int64)	

Name	Description	Type	Schema
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer(int32)	
tokenCode	Token code	string	
tokenName	Token name	string	
walletAddress	Wallet address	string	
walletName	Wallet name	string	
walletType	Wallet type	integer(int32)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "balance": 0,
        "createTime": 0,
        "status": 0,
        "tokenCode": "",
        "tokenName": "",
        "walletAddress": "",
        "walletName": "",
        "walletType": 0
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Wallet List by Pagination

url: `/api/openapi/v1/wallet/query`

method: `POST`

produces: `application/x-www-form-urlencoded,application/json`

consumes: `*/*`

Note:

This interface is called to query the wallet list by pagination.

Request Example:

```
{
  "body": {
    "data": {
      "endCreateTime": 0,
      "startCreateTime": 0,
      "status": 0,
      "tokenCode": "",
      "walletAddress": "",
      "walletType": 0
    },
    "page": {
      "pageNum": 0,
      "pageSize": 0
    }
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
apiReqInfoOfDataTableOfWalletQueryReqVO	apiReqInfoOfDataTableOfWalletQueryReqVO	body	true	ApiReqInfoOfDataTableOfWalletQueryReqVO	ApiReqInfoOfDataTableOfWalletQueryReqVO
body			false	DataTableOfWalletQueryReqVO	DataTableOfWalletQueryReqVO
data	Data		true	WalletQueryReqVO	WalletQueryReqVO
endCreateTime	End created on		false	integer	
startCreateTime	Start created on		false	integer	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed		false	integer	
tokenCode	Token code		true	string	
walletAddress	Wallet address		false	string	
walletType	Wallet type		false	integer	
page	Page		true	PageInfo	PageInfo
pageNum	Page number		true	integer	
pageSize	Page size		true	integer	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfResultDataOfWalletQueryRespVO

Response Params:

Name	Description	Type	Schema
body		ResultDataOfWalletQueryRespVO	ResultDataOfWalletQueryRespVO
page	page	ResultPageInfo	ResultPageInfo
isFirstPage	First page	boolean	
isLastPage	Last page	boolean	
pageNum	Page number	integer	
pageSize	Page size	integer	
pages	Pages	integer	
total	Total	integer	
rows	Rows	array	WalletQueryRespVO
balance	Wallet balance	number	
createTime	Created on	integer	
status	Status 5: Pending approval 10: Approval in progress 15: Rejected 20: Pending submission 30: Submitting 35: Success 40: Failed	integer	
tokenCode	Token code	string	
tokenName	Token name	string	
walletAddress	Wallet address	string	
walletName	Wallet name	string	
walletType	Wallet type	integer	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
            "pageNum": 0,
            "pageSize": 0,
            "pages": 0,
            "total": 0
        },
        "rows": [
            {
                "balance": 0,
                "createTime": 0,
                "status": 0,
                "tokenCode": "",
                "tokenName": "",
                "walletAddress": "",
                "walletName": "",
                "walletType": 0
            }
        ]
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Swap Module

Token Swap - Create Transaction

url: `/api/openapi/v1/transaction/cross/createTransaction`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to create Token swap transaction.

For a Token swap using the platform's exchange rate, enterprise users have already requested authorization in advance and do not need to sign the transaction. Please begin at step 3.

Step 1: The user calls the "Token Swap - Create Transaction" interface to compile the data into the transaction body.

Step 2: The user uses the private key to sign the transaction.

Step 3: The user calls the "Token Swap - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Token Swap - Query Transaction Result" interface to query the transfer results.

Request Example:

```json
{
  "body": {
    "exchangeRate": 0,
    "fromAccountAddress": "",
    "fromBankBic": "",
    "fromTokenAmount": 0,
    "fromTokenCode": "",
    "fxProviderBic": "",
    "rateType": 0,
    "remarks": "",
    "toAccountAddress": "",
    "toBankBic": "",
    "toTokenAmount": 0,
    "toTokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfCreateCrossTransactionReqVO	ApiReqInfoOfCreateCrossTransactionReqVO
body			false	CreateCrossTransactionReqVO	CreateCrossTransactionReqVO
authId	Auth Id		true	integer	
exchangeRate	Exchange rate		true	number	
fromAccountAddress	From wallet address		true	string	
fromBankBic	Sender bank BIC		true	string	
fromTokenAmount	Source Token amount		true	number	
fromTokenCode	Source Token code		true	string	
fxProviderBic	FX Provider BIC		true	string	
rateType	Exchange Rate Source 1: Local Commercial Bank 2: Kissen network		true	integer	
remarks	Remark		true	string	
toAccountAddress	To wallet address		true	string	
toBankBic	Recipient bank BIC		true	string	
toTokenAmount	Target Token amount		true	number	
toTokenCode	Target Token code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema

Code	Description	Schema
200	OK	ApiRespInfoOfCreateTransactionRespVO

Response Params:

Name	Description	Type	Schema
body		CreateTransactionRespVO	CreateTransactionRespVO
chainId	Blockchain ID	string	
rawTransaction	Raw transaction	RawTransactionRespVO	RawTransactionRespVO
data		string	
feeCap		integer	
gasLimit		integer	
gasPremium		integer	
gasPrice		integer	
nonce		integer	
to		string	
value		integer	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "chainId": "",
        "rawTransaction": {
            "data": "",
            "feeCap": 0,
            "gasLimit": 0,
            "gasPremium": 0,
            "gasPrice": 0,
            "nonce": 0,
            "to": "",
            "value": 0
        }
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Swap - Send Transaction

url: `/api/openapi/v1/transaction/cross/sendTransaction`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to query Token swap transaction result.

Step 1: The user calls the "Token Swap - Create Transaction" interface to compile the data into the transaction body.

Step 2: The user uses the private key to sign the transaction.

Step 3: The user calls the "Token Swap - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Token Swap - Query Transaction Result" interface to query the transfer results.

Request Example:

```
{
  "body": {
    "exchangeRate": 0,
    "fromAccountAddress": "",
    "fromBankBic": "",
    "fromTokenAmount": 0,
    "fromTokenCode": "",
    "fxProviderBic": "",
    "rateType": 0,
    "remarks": "",
    "signedMessage": "",
    "toAccountAddress": "",
    "toBankBic": "",
    "toTokenAmount": 0,
    "toTokenCode": "",
    "txHash": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfSendCrossTransactionReqVO	ApiReqInfoOfSendCrossTransactionReqVO
body			false	SendCrossTransactionReqVO	SendCrossTransactionReqVO

Name	Description	In	Required	Type	Schema
authId	Auth Id		true	integer	
exchangeRate	Exchange rate		true	number	
fromAccountAddress	From wallet address		true	string	
fromBankBic	Sender bank BIC		true	string	
fromTokenAmount	Source Token amount		true	number	
fromTokenCode	Source Token code		true	string	
fxProviderBic	FX Provider BIC		true	string	
rateType	Exchange Rate Source 1: Local Commercial Bank 2: Kissen network		true	integer	
remarks	Remark		true	string	
signedMessage	Signed message		true	string	
toAccountAddress	To wallet address		true	string	
toBankBic	Receiver bank BIC		true	string	
toTokenAmount	Target Token amount		true	number	
toTokenCode	Target Token code		true	string	
txHash	Tx Hash		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfSendTransactionRespVO

Response Params:

Name	Description	Type	Schema
body		SendTransactionRespVO	SendTransactionRespVO
serialNumber	Serial number	string	
txHash	Tx Hash	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "txHash": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Token Swap - Query Transaction Result

url: `/api/openapi/v1/transaction/cross/queryResult`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to query Token swap transaction result.

Step 1: The user calls the "Token Swap - Create Transaction" interface to compile the data into the transaction body.

Step 2: The user uses the private key to sign the transaction.

Step 3: The user calls the "Token Swap - Send Transaction" interface to upload the transaction to the blockchain.

Step 4: The user calls "Token Swap - Query Transaction Result" interface to query the transfer results.

Request Example:

```
{
  "body": {
    "txHash": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfQueryTransactionResultReqVO	ApiReqInfoOfQueryTransactionResultReqVO
body			false	QueryTransactionResultReqVO	QueryTransactionResultReqVO
serialNumber	Serial number		true	string	
txHash	Tx Hash		false	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	

Name	Description	In	Required	Type	Schema
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfQueryCrossResultRespVO

Response Params:

Name	Description	Type	Schema
body		QueryCrossResultRespVO	QueryCrossResultRespVO
commission	Commission fee	number	
contractBankCode	FX Provider code	string	
crossTxDetailRespVOS	Token swap tx details	array	CrossTxDetailsRespVO
createTime	Created on	integer	
detailType	Transaction Type 1: User-initiated transaction 5: User authorization 10: User transfers to source Token's liquidity pool 20: Target Token's liquidity pool transfers to recipient's wallet	integer	
fromAccountAddress	From wallet address	string	
handlerCode	Handler code	string	
remarks	Remark	string	
senderAccountAddress	Sender wallet address	string	
toAccountAddress	To wallet address	string	
tokenCode	Source Token code	string	

Name	Description	Type	Schema
tokenCount	Token amount	number	
txHash	Tx Hash	string	
txStatus	Tx Status 20: Pending on-chain submission 30: Submitting 35: Successful 40: Failed	integer	
txTime	Tx time	integer	
exchangeRate	Exchange rate	number	
failureReason	Failure reasons	string	
fromAccountAddress	From wallet address	string	
fromCapitalAddress	Source Token's liquidity pool address	string	
fromTokenAmount	Source Token amount	number	
fromTokenCode	Souerce Token code	string	
fxProviderBic	FX Provider BIC	string	
rateId	Exchange rate ID	string	
rateType	Exchange Rate Source 1: Local Commercial Bank 2: Kissen network	integer(int32)	
remarks	Remark	string	
serialNumber	Serial number	string	
spBankName	FX Provider name	string	

Name	Description	Type	Schema
status	Status 1: Saved 3: Withdrawn 5: Pending approval 10: Approval in progress 15: Rejected 20: Approved and pending submission 30: Submitting 35: Successful 40: Failed 45: Deleted	integer(int32)	
toAccountAddress	To wallet address	string	
toCapitalAddress	Target Token's capital pool	string	
toTokenAmount	Target Token amount	number	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
type	Type 1: Token swap 2: FX purchase 3: Token transfer	integer(int32)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "commission": 0,
        "contractBankCode": "",
        "crossTxDetailRespVOS": [
            {
                "createTime": 0,
                "detailType": 0,
                "fromAccountAddress": "",
                "handlerCode": "",
                "remarks": "",
                "senderAccountAddress": "",
                "toAccountAddress": "",
                "tokenCode": "",
                "tokenCount": 0,
                "txHash": "",
                "txStatus": 0,
                "txTime": 0
            }
        ],
        "exchangeRate": 0,
        "failureReason": "",
        "fromAccountAddress": "",
        "fromCapitalAddress": "",
        "fromTokenAmount": 0,
        "fromTokenCode": "",
        "fxProviderBic": "",
        "rateId": "",
        "rateType": 0,
        "remarks": "",
        "serialNumber": "",
        "spBankName": "",
        "status": 0,
        "toAccountAddress": "",
        "toCapitalAddress": "",
        "toTokenAmount": 0,
        "toTokenCode": "",
        "txHash": "",
        "txTime": 0,
        "type": 0
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Transaction Result

url: `/api/openapi/v1/transaction/cross/tx/list`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to query the transaction list where the current enterprise user acts as the recipient.

Request Example:

```
{
  "body": {
    "endTime": 0,
    "limit": 0,
    "startTime": 0,
    "type": 0
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfQueryTransactionListReqVO	ApiReqInfoOfQueryTransactionListReqVO
body			false	QueryTransactionListReqVO	QueryTransactionListReqVO
endTime	End time		false	integer	
limit	Number of records to query, default 30 and maximum 100. If exceeded, it defaults to 100.		false	integer	
startTime	Start time		false	integer	
type	Type 1: Token swap 2: FX purchase 3: Token transfer		false	integer	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfQueryCrossResultRespVO

Response Params:

Name	Description	Type	Schema
body		array	QueryCrossResultRespVO
commission	Commission fee	number	
contractBankCode	FX Provider bank code	string	
crossTxDetailRespVOS	Token swap tx details	array	CrossTxDetailsRespVO
createTime	Created on	integer	

Name	Description	Type	Schema
detailType	Transaction Type 1: User-initiated transaction 5: User authorization 10: User transfers to source Token's liquidity pool 20: Target Token's liquidity pool transfers to recipient's wallet	integer	
fromAccountAddress	From wallet address	string	
handlerCode	Handler code	string	
remarks	Remark	string	
senderAccountAddress	Sender wallet address	string	
toAccountAddress	To wallet address	string	
tokenCode	Source Token code	string	
tokenCount	Token amount	number	
txHash	Tx Hash	string	
txStatus	Status 20: Approved and pending submission 30: Submitting 35: Successful 40: Failed	integer	
txTime	Tx time	integer	
exchangeRate	Exchange rate	number	

Name	Description	Type	Schema
failureReason	Failure reasons	string	
fromAccountAddress	From wallet address	string	
fromCapitalAddress	Source Token's liquidity pool address	string	
fromTokenAmount	Source Token amount	number	
fromTokenCode	Souerce Token code	string	
fxProviderBic	FX Provider BIC	string	
rateId	Exchange rate ID	string	
rateType	Exchange Rate Source 1: Local Commercial Bank 2: Kissen network	integer(int32)	
remarks	Remark	string	
serialNumber	Serial number	string	
spBankName	FX Provider name	string	

Name	Description	Type	Schema
status	Status 1: Saved 3: Withdrawn 5: Pending approval 10: Approval in progress 15: Rejected 20: Approved and pending submission 30: Submitting 35: Successful 40: Failed 45: Deleted	integer(int32)	
toAccountAddress	To wallet address	string	
toCapitalAddress	Target Token's capital pool address	string	
toTokenAmount	Target Token amount	number	
toTokenCode	Target Token code	string	
txHash	Tx Hash	string	
txTime	Tx time	integer(int64)	
type	Type 1: Token swap 2: FX purchase 3: Token transfer	integer(int32)	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": [
        {
            "commission": 0,
            "contractBankCode": "",
            "crossTxDetailRespVOS": [
                {
                    "createTime": 0,
                    "detailType": 0,
                    "fromAccountAddress": "",
                    "handlerCode": "",
                    "remarks": "",
                    "senderAccountAddress": "",
                    "toAccountAddress": "",
                    "tokenCode": "",
                    "tokenCount": 0,
                    "txHash": "",
                    "txStatus": 0,
                    "txTime": 0
                }
            ],
            "exchangeRate": 0,
            "failureReason": "",
            "fromAccountAddress": "",
            "fromCapitalAddress": "",
            "fromTokenAmount": 0,
            "fromTokenCode": "",
            "fxProviderBic": "",
            "rateId": "",
            "rateType": 0,
            "remarks": "",
            "serialNumber": "",
            "spBankName": "",
            "status": 0,
            "toAccountAddress": "",
            "toCapitalAddress": "",
            "toTokenAmount": 0,
            "toTokenCode": "",
            "txHash": "",
            "txTime": 0,
            "type": 0
        }
    ],
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```

Query Exchange Rate List

url: `/api/openapi/v1/transaction/rate/list`

Method: `POST`

produces: `application/json`

consumes: `*/*`

Notes:

This interface is called to query the exchange rate list based on the source and target Tokens.

Request Example:

```
{
  "body": {
    "fromTokenAmount": 0,
    "fromTokenCode": "",
    "toTokenCode": ""
  },
  "mac": "",
  "reqTime": 0,
  "transCode": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
reqInfo	reqInfo	body	true	ApiReqInfoOfQueryTxRateListReqVO	ApiReqInfoOfQueryTxRateListReqVO
body			false	QueryTxRateListReqVO	QueryTxRateListReqVO
fromTokenAmount	Source Token amount		false	number	
fromTokenCode	Source Token code		true	string	
toTokenCode	Target Token code		true	string	
mac	Signature		true	string	
reqTime	Request time		true	integer(int64)	
transCode	Request ID		true	string	

Status:

Code	Description	Schema
200	OK	ApiRespInfoOfQueryTxRateListRespVO

Response Params:

Name	Description	Type	Schema
body		QueryTxRateListRespVO	QueryTxRateListRespVO
exchangeRate	Exchange rate	number	
fromTokenAmount	Source Token amount	number	
fromTokenCode	Source Token code	string	

Name	Description	Type	Schema
fromTokenName	Source Token name	string	
fromTokenSymbol	Source Token symbol	string	
rateProviderRespVOS	FX Provider list	array	RateProviderRespVO
contractBankCode	FX Provider bank code	string	
fromCapitalAddress	Source Token's capital pool address	string	
fxProviderBankName	FX Provider bank name	string	
fxProviderBic	FX Provider BIC	string	
toCapitalAddress	Target Token's capital pool address	string	
rateType	Exchange Rate Source 1: Local Commercial Bank 2: Kissen network	integer(int32)	
toTokenAmount	Target Token amount	number	
toTokenCode	Target Token code	string	
toTokenName	Target Token name	string	
toTokenSymbol	Target Token symbol	string	
code		string	
mac		string	
respTime		integer(int64)	integer(int64)
transCode		string	

Response Example:

```
{
    "body": {
        "exchangeRate": 0,
        "fromTokenAmount": 0,
        "fromTokenCode": "",
        "fromTokenName": "",
        "fromTokenSymbol": "",
        "rateProviderRespVOS": [
            {
                "contractBankCode": "",
                "fromCapitalAddress": "",
                "fxProviderBankName": "",
                "fxProviderBic": "",
                "toCapitalAddress": ""
            }
        ],
        "rateType": 0,
        "toTokenAmount": 0,
        "toTokenCode": "",
        "toTokenName": "",
        "toTokenSymbol": ""
    },
    "code": "",
    "mac": "",
    "respTime": 0,
    "transCode": ""
}
```