Commercial Bank OpenAPI Documentation for Management System Local FX Rate v1.1.0

This document outlines the functional interface specifications for the Commercial Bank Control Panel. Utilizing the Spring Cloud framework, OpenAPI enables unified management and routing of backend services. It ensures secure API access by validating visitor permissions and forwarding requests to the Panel, facilitating seamless integration with existing business processes and software.

Introduction

Communication Mode

1. All APIs outlined in this document are accessed via HTTPS requests (POST mode) with UTF-8 encoding for request characters.
2. Outputs are delivered in JSON format.

Interface Specification

Format of the Request Message

1. Configure the HTTP request header to include `key = token`, where value is the access key provided by the Commercial Bank Control Panel. The request information can be encapsulated in the body according to the specific interface requirements.

Sample Request:

```
curl -X POST  -H  "token:fc84******39f4" -H  "Content-Type:application/json" -d "{\"param_key\":\"param_value\"}" "[panel_address://url]"
```

Definition of Terms

1. **token**: A credential for accessing the Commercial Bank Control Panel OpenAPIs. Users can find their token by logging into the Commercial Bank Control Panel and navigating to Account > OpenAPI Access Key.
2. **panel_address**: The address for accessing the Commercial Bank Control Panel OpenAPIs. Users can find their token by logging into the Commercial Bank Control Panel and navigating to Account > OpenAPI Access Key.
3. **url**: The specific business interface URL, which can be reviewed in the Interface List section.

Format of Response Message

Parameter name	Description	Data type	Required
code	Response code	integer(int32)	integer(int32)
data	Request data, tailored to specific business needs	Object	Object
message	Response message code	string	

Sample Response:

```
{
    "code": 0,
    "data": {
    },
    "message": ""
}
```

Response Status Code

Response code	Description
MSG_00_0000	Success
MSG_00_0001	The system is experiencing issues. Please try again later.
MSG_00_0002	The system is busy, please try again later.
MSG_00_0005	Token verification failed
MSG_00_0009	Paging information cannot be empty
MSG_00_0010	The number of pages cannot be empty
MSG_00_0011	The quantity per page cannot be empty
MSG_00_0012	Token verification failed
MSG_00_0013	The role has been assigned to the user and cannot be deleted
MSG_00_0014	Permission is required
MSG_00_0015	Token decryption failed
MSG_05_4000	Currency cannot be empty
MSG_05_4001	Exchange rate cannot be empty
MSG_05_4002	Exchange rate pair already exists
MSG_05_4003	Exchange rate pair does not exist
MSG_05_4004	Currency does not exist
MSG_05_4005	Exchange rate ID cannot be empty
MSG_05_4006	Exchange rate value is incorrect
MSG_05_4007	Notification to Management System failed
MSG_05_4008	The status cannot be empty
MSG_05_4009	The status is incorrect

Interface List

Local FX Rate Management

Upload Local FX Rate

Url: `/api/manage/v2/liquidity/rate/local/updateRate`

Method: `POST`

Produces: `application/json`

Produces: `*/*`

Notes: This interface is called to upload the FX Provider's local exchange rate. When the enterprises you are connected to perform cross-Token transfers, the local exchange rates will be applied.

Request Example:

```
{
  "exchangeRate": 0,
  "fromCurrency": "",
  "toCurrency": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
req	req	body	true	LocalRateEditReqVO	LocalRateEditReqVO
exchangeRate	Exchange rate		true	number	
fromCurrency	Source Token symbol		true	string	
toCurrency	Target Token symbol		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOfboolean

Response Params:

Name	Description	Type	Schema
code	Code	integer(int32)	integer(int32)
data	Data	boolean	
message	Message	string	

Response Example:

```
{
    "code": 0,
    "data": true,
    "message": ""
}
```

Query Local FX Rate

Url: `/api/manage/v2/liquidity/rate/local/getRateInfo`

Method: `POST`

Produces: `application/json`

Produces: `*/*`

Notes: This interface is called to query the local exchange rate details.

Request Example:

```
{
  "fromCurrency": "",
  "toCurrency": ""
}
```

Request Params:

Name	Description	In	Required	Type	Schema
req	req	body	true	LocalRateQueryReqVO	LocalRateQueryReqVO
fromCurrency	Source Token symbol		true	string	
toCurrency	Target Token symbol		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOfListOfLocalRateListRespVO

Response Params:

Name	Description	Type	Schema
code	Code	integer(int32)	integer(int32)
data	Data	array	LocalRateListRespVO
createTime	Created on	integer(int64)	
createUser	Created by	string	
exchangeRate	Exchange rate	number	

Name	Description	Type	Schema
fromCurrency	Source Token	string	
rateId	Rate ID	integer(int64)	
state	Status 0: Activation in progress 1: Active 2: Failed 3: Inactive	integer(int32)	
toCurrency	Target Token	string	
updateTime	Updated on	integer(int64)	
message	Message	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "createTime": 0,
            "createUser": "",
            "exchangeRate": 0,
            "fromCurrency": "",
            "rateId": 0,
            "state": 0,
            "toCurrency": "",
            "updateTime": 0
        }
    ],
    "message": ""
}
```