Commercial Bank Control Panel OpenAPI Documentation v1.0.0

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_00_0100	Network anomaly
MSG_00_0101	Data acquisition error
MSG_00_0102	Data parsing error
MSG_00_0103	Business logic error
MSG_02_1000	Custodian Bank does not exist
MSG_02_1001	Bank name cannot be empty
MSG_02_1002	Bank BIC cannot be empty
MSG_02_1003	Bank brief introduction cannot be empty
MSG_02_1004	Bank description cannot be empty
MSG_02_1005	Bank wallet address cannot be empty
MSG_02_1006	BN code cannot be empty
MSG_02_1007	Contact name cannot be empty

Response code	Description
MSG_02_1008	Mobile number cannot be empty
MSG_02_1009	Email cannot be empty
MSG_02_1010	Business license file name cannot be empty
MSG_02_1011	Detailed address cannot be empty
MSG_02_1012	Business license file hash cannot be empty
MSG_02_1013	Bank code cannot be empty
MSG_02_1014	Already exists
MSG_02_1015	Does not exist
MSG_02_1016	The current status does not allow for editing
MSG_02_1018	The Commercial Bank does not belong to this Custodian Bank
MSG_02_1019	Invalid account address
MSG_02_1020	Bank info already exists
MSG_02_1021	Account address already exists
MSG_02_1022	Business license file already exists
MSG_02_1030	Currency cannot be empty
MSG_02_1031	Amount cannot be empty
MSG_02_1035	Custodian Bank does not exist
MSG_02_1036	Chain address does not exist
MSG_02_1037	Wallet does not exist
MSG_02_1038	The wallet status is incorrect
MSG_00_1000	Creation method cannot be empty
MSG_00_1001	Custodian Bank cannot be empty
MSG_00_1002	Business node cannot be empty
MSG_00_1003	Exceeded the maximum number of wallets that can be created
MSG_00_1004	Custodian Bank does not exist
MSG_00_1005	Wallet creation failed
MSG_00_1006	Failed to save chain account
MSG_00_1007	The wallet address must be start with '0x'
MSG_00_1008	The wallet address must be 42 characters long

Response code	Description
MSG_00_1009	The wallet address must be a 40-digit hexadecimal string starting with '0x' (42 bytes in total)
MSG_00_1010	The wallet address is invalid, regex validation pattern is: [0]{1}[x]{1}[A-Fa-f0-9]{40}
MSG_00_1011	Wallet address cannot be empty
MSG_00_1012	The wallet has already been created under this Custodian Bank
MSG_00_1013	Submission error
MSG_00_1014	Amount cannot be empty
MSG_00_1015	Password cannot be empty
MSG_00_1016	Wallet account is unavailable
MSG_00_1017	Failed to save
MSG_00_1018	Please select a top-up or withdrawal type
MSG_00_1019	Failed to update wallet balance
MSG_00_1020	Wallet address is unavailable
MSG_00_1021	Bank account ID cannot be empty
MSG_00_1022	The password and confirmation password do not match. Please ensure both entries are the same.
MSG_00_1023	Failed to read file
MSG_00_1024	Password cannot be parsed by keystore
MSG_00_1025	Wallet address already exists
MSG_00_1026	Insufficient wallet balance
MSG_00_1027	Incorrect password
MSG_00_1028	Password verification failed
MSG_00_1029	Incorrect wallet address
MSG_00_2001	The transfer information cannot be empty
MSG_03_0001	The last transaction authorization has not been completed
MSG_03_0002	Remitter account does not exist
MSG_03_0003	There is no Custodian Bank linked to current Commercial Bank
MSG_03_0004	The receiving bank does not exist
MSG_03_0005	The remittance account and the receiving account cannot be the same

Response code	Description
MSG_03_0006	Receiving account does not exist
MSG_03_0007	The remittance bank and receiving bank cannot be affiliated with the same Custodian Bank
MSG_03_0008	The remittance bank and the receiving bank cannot be the same
MSG_03_0009	Exchange rate information cannot be empty
MSG_03_0010	Exchange rate information does not exist
MSG_03_0011	Abnormal exchange rate information
MSG_00_3000	Reserve account cannot be empty
MSG_00_3001	Transaction reference number cannot be empty
MSG_00_3002	Incorrect amount format
MSG_00_3003	Amount conversion error
MSG_00_3004	Transaction reference number already exists
MSG_00_3005	The remittance account and receiving account cannot be the same
MSG_00_3006	The token access information does not exist
MSG_00_3007	Insufficient fiat balance in the reserve account
MSG_03_8001	Task code cannot be empty
MSG_03_8002	The contract package does not exist
MSG_03_8003	The contract package has been disabled
MSG_03_8004	The contract task does not exist
MSG_03_8005	The contract info does not exist
MSG_03_8006	The Token ID cannot be empty
MSG_03_8007	The business mapping cannot be empty
MSG_03_8008	The rule ID cannot be empty
MSG_03_8009	The business mapping does not exist
MSG_03_8010	The business mapping has been configured for this Token; please do not duplicate the settings.
MSG_03_8011	The account mapping record already exists
MSG_03_8012	The account mapping record does not exist
MSG_03_8013	Failed to download financial statement
MSG_03_8014	Account type cannot be empty

Response code	Description
MSG_03_8015	Downloaded file data is empty
MSG_03_5012	Abnormal status
MSG_03_7200	The billing rule does not exist
MSG_03_7201	Billing rule has been deactivated
MSG_03_7202	Billing rule has been activated
MSG_03_7203	The module type to be exported does not exist
MSG_03_7204	The export type does not exist
MSG_03_7205	Export param error
MSG_03_7206	Token ID does not exist
MSG_03_7207	The export strategy does not exist
MSG_03_7208	The transaction types cannot be empty
MSG_03_7209	The task name cannot be empty
MSG_03_7210	The export rule already exists
MSG_03_7211	The export task name already exists
MSG_03_7212	The export task rule ID cannot be empty
MSG_03_7213	The export task rule is inactive
MSG_03_7214	The export task rule is active
MSG_03_7215	The export task rule is active and cannot be deleted
MSG_03_7216	The export rule does not exist
MSG_03_7217	The export task does not exist
MSG_03_7218	The export task has not been completed and cannot be deleted
MSG_03_7219	The export task has been deleted
MSG_03_7220	The export task ID cannot be empty
MSG_03_7221	The ledger name cannot be empty
MSG_03_7222	The ledger name already exists
MSG_03_7223	Rule info already exists
MSG_03_7224	Rule info does not exist
MSG_03_7225	Scheduler error
MSG_03_7226	Rule name cannot be empty

Response code	Description
MSG_03_7227	Rule business type cannot be empty
MSG_03_7228	Rule frequency cannot be empty
MSG_03_7229	Rule frequency type cannot be empty
MSG_03_7230	Rule details cannot be empty
MSG_03_7231	Risk score cannot be empty
MSG_03_7232	Priority cannot be empty
MSG_03_7233	Token info does not exist
MSG_03_7234	Rule ID cannot be empty
MSG_03_7235	Rule suspicious ID cannot be empty
MSG_03_7236	Rule process name cannot be empty
MSG_03_7237	Rule process date cannot be empty
MSG_03_7238	Rule process remark cannot be empty
MSG_03_7241	Token does not exist
MSG_03_7242	Token ID cannot be empty
MSG_03_7243	Rule handle type cannot be empty
MSG_03_7245	Rule name already exists
MSG_03_7246	Rule is inactive
MSG_03_7247	Rule is active
MSG_03_7248	Notify type cannot be empty
MSG_03_7249	Contact info cannot be empty
MSG_03_7250	Alert info cannot be empty
MSG_03_7251	Priority repeated
MSG_03_7252	Rank interval error
MSG_03_7253	Work in progress and additional operations cannot be proceeded
MSG_03_7254	Account type cannot be empty
MSG_03_7255	Interest type cannot be empty
MSG_03_7256	Interest policy name cannot be empty
MSG_03_7257	Interest calculation method cannot be empty
MSG_03_7258	Effective date cannot be empty

Response code	Description
MSG_03_7259	Daily interest calculation time cannot be empty
MSG_03_7260	Monthly interest calculation time cannot be empty
MSG_03_7261	Daily interest decimal place cannot be empty
MSG_03_7262	Monthly interest decimal place cannot be empty
MSG_03_7263	Account balance min value cannot be empty
MSG_03_7264	Account balance max value cannot be empty
MSG_03_7265	Annual interest rate cannot be empty
MSG_03_7266	The interest policy name already exists
MSG_03_7267	The interest type error
MSG_03_7268	The account type error
MSG_03_7269	The interest calculation method error
MSG_03_7270	Interest rule ID cannot be empty
MSG_03_7271	Interest info does not exist
MSG_03_7272	Date of interest calculation cannot be empty
MSG_03_7273	Transaction type duplication
MSG_03_7274	Transaction type error
MSG_03_7275	Transaction type does not support the selected Token
MSG_03_7276	Accrual record ID cannot be empty
MSG_03_7277	Rule wallet ID cannot be empty
MSG_03_7278	Fee type cannot be empty
MSG_03_7279	Fee period cannot be empty
MSG_03_7281	Interest is being issued and cannot be processed
MSG_03_7282	Admin wallet info duplication
MSG_03_7283	The effective time must be later than the current time
MSG_03_7284	Keystore and password mismatch
MSG_03_7285	Account not found
MSG_03_7286	Wallet address already exists. Please choose a different one.
MSG_03_7287	The reserve account does not exist
MSG_00_1000	System error

Response code	Description
MSG_00_1001	Value busId cannot be empty
MSG_00_1002	Value busType cannot be empty
MSG_00_1003	File cannot be empty
MSG_00_1004	Record does not exist
MSG_00_1005	File does not exist
MSG_00_1006	Failed to download sftp file
MSG_00_1007	Failed to upload sftp file

Interface List

Authentication Module

Shiro Login

url: `/lpcp/fxsp/login/user`

method: `POST`

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

consumes: `*/*`

Note:

Shiro login

Params:

Name	Description	In	Required	Type	Schema
code	User info	query	false	string	

Response Params:

None

Response Example:

```
```

Log-out

url: `/lpcp/fxsp/logout/user`

method: `POST`

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

consumes: `*/*`

Note:

Login out

Params:

None

Response Params:

None

Response Example:

```
```

User Management Module

Update User Status

url: `/lpcp/v1/fxsp/sys/user/manage/status/update`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to enable/disable the user.

Example:

```
{
  "userId": 0,
  "lockable": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserStatusUpdate	InUserStatusUpdate	body	true	InUserStatusUpdate	InUserStatusUpdate
userId	User ID		true	integer(int64)	

Name	Description	In	Required	Type	Schema
lockable	Action 1: Disable 2: Enable 3: Delete		true	integer(int32)	

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

Query User List by Pagination

url: `/lpcp/v1/fxsp/sys/user/manage/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to get the user list by pagination.

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "userName": "",
  "realName": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserSearches	InUserSearches	body	true	InUserSearches	InUserSearches
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
userName	User name		false	string	
realName	Real name		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoListOutUserSearches
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	array	OutUserSearches

Name	Description	Type	Schema
userId	User ID	integer(int64)	
userName	User name	string	
realName	Real name	string	
telephone	Telephone	string	
email	Email	string	
lockable	Status 1: Inactive 2: Active 3: Deleted	integer(int32)	
superUser	Superadmin 1: Yes 2: No	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "userId": 0,
            "userName": "",
            "realName": "",
            "telephone": "",
            "email": "",
            "lockable": 0,
            "superUser": 0
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

Create User

url: `/lpcp/v1/fxsp/sys/user/manage/save`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to create a user.

Example:

```
{
  "userName": "",
  "realName": "",
  "telephone": "",
  "email": "",
  "lockable": 0,
  "roleIdList": [],
  "pwd": "",
  "superUser": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserSave	InUserSave	body	true	InUserSave	InUserSave
userName	User name		true	string	
realName	Real name		true	string	
telephone	Telephone		true	string	
email	Email		true	string	
lockable	Status 1: Inactive 2: Active 3: Deleted		true	integer(int32)	
roleIdList	Role ID list		true	array	integer(int64)
pwd	Password		false	string	

Name	Description	In	Required	Type	Schema
superUser	Superadmin 1: Yes 2: No		false	integer(int32)	

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

Change User Password

url: `/lpcp/v1/fxsp/sys/user/manage/password/update`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to change a user's password.

Example:

```
{
  "oldPwd": "",
  "newPwd": "",
  "verifyPwd": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserManagePasswordUpdate	InUserManagePasswordUpdate	body	true	InUserManagePasswordUpdate	InUserManagePasswordUpdate
oldPwd	Old password		true	string	
newPwd	New password		true	string	
verifyPwd	Verify password		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

Reset User Password

url: `/lpcp/v1/fxsp/sys/user/manage/password/reset`

method: `POST`

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

consumes: `*/*`

Note:

This interface is used to reset a user's password to the default password: 'Abcd1234'.

Example:

```
{
  "userId": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserManagePasswordReset	InUserManagePasswordReset	body	true	InUserManagePasswordReset	InUserManagePasswordReset
userId	User ID		true	integer(int64)	

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	API return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

First-Time Login Check

url: `/lpcp/v1/fxsp/sys/user/manage/first/login`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to check user first-time login status. The password can only be set during the initial login process for a user.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutUserManageFirstLogin
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutUserManageFirstLogin	OutUserManageFirstLogin
firstLogin	User first-time login True False	string	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Update User Information

url: `/lpcp/v1/fxsp/sys/user/manage/edit`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to update user information.

Example:

```
{
  "userId": 0,
  "userName": "",
  "realName": "",
  "telephone": "",
  "email": "",
  "lockable": 0,
  "roleIdList": []
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserEdit	InUserEdit	body	true	InUserEdit	InUserEdit
userId	User ID		true	integer(int64)	
userName	User name		true	string	
realName	Real name		true	string	
telephone	Telephone		true	string	
email	Email		true	string	
lockable	Status 1: Disable 2: Enable 3: Delete		true	integer(int32)	

Name	Description	In	Required	Type	Schema
roleIdList	Role ID		true	array	integer(int64)

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query User Details

url: `/lpcp/v1/fxsp/sys/user/manage/edit/detail/search`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to get user details by user ID.

Example:

```
{
  "userId": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inUserEditDetailSearch	InUserEditDetailSearch	body	true	InUserEditDetailSearch	InUserEditDetailSearch
userId	User ID		true	integer(int64)	

Status:

Code	Description	Schema
200	OK	ResultInfoOutUserEditDetailSearch
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutUserEditDetailSearch	OutUserEditDetailSearch
userId	User ID	integer(int64)	
userName	User name	string	
realName	Real name	string	
telephone	Telephone	string	
email	Email	string	

Name	Description	Type	Schema
lockable	Status 1: Inactive 2: Active 3: Deleted	integer(int32)	
roleList	Role list	array	OutRole
roleId	Role ID	integer	
roleName	Role name	string	
roleCode	Role code	string	
description	Description	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "userId": 0,
        "userName": "",
        "realName": "",
        "telephone": "",
        "email": "",
        "lockable": 0,
        "roleList": [
            {
                "roleId": 0,
                "roleName": "",
                "roleCode": "",
                "description": ""
            }
        ]
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Upgrade manage

Upgrade to Settlement Bank

url: `/lpcp/v1/fxsp/sys/upgrade/save`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to submit a request to update the Commercial Bank portfolio to the Settlement Bank.

Example:

```
{
  "fileList": [
    {
      "fileUrl": "",
      "fileCode": ""
    }
  ],
  "capitalPoolList": [
    {
      "capitalPoolAddress": "",
      "capitalPoolPlatform": "",
      "capitalPoolCurrency": ""
    }
  ]
}
```

Params:

Name	Description	In	Required	Type	Schema
inUpgradeInfo	InUpgradeInfo	body	true	InUpgradeInfo	InUpgradeInfo
fileList	File list		true	array	UpgradeFile
fileUrl	File URL		true	string	
fileCode	File hash		true	string	
capitalPoolList	Capital pool list		true	array	UpgradeCapitalPool
capitalPoolAddress	Capital pool address		true	string	

Name	Description	In	Required	Type	Schema
capitalPoolPlatform	Capital pool platform		true	string	
capitalPoolCurrency	Capital pool currency		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Upgrade Request Status

url: `/lpcp/v1/fxsp/sys/upgrade/detail`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to check the status of the upgrade request.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutUpgradeDetail
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutUpgradeDetail	OutUpgradeDetail
bankType	Bank Type 1: Commercial Bank 2: Settlement Bank	integer(int32)	
updateStatus	Upgrade Status 0:init 5: Pending approval 10: Approved 15: Rejected	integer(int32)	
remark	Remark	string	
fileList	File list	array	UpgradeFile
fileUrl	File URL	string	

Name	Description	Type	Schema
fileCode	File hash	string	
capitalPoolList	Capital pool list	array	UpgradeCapitalPool
capitalPoolAddress	Capital pool address	string	
capitalPoolPlatform	Capital pool platform	string	
capitalPoolCurrency	Capital pool currency	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "bankType": 0,
        "updateStatus": 0,
        "remark": "",
        "fileList": [
            {
                "fileUrl": "",
                "fileCode": ""
            }
        ],
        "capitalPoolList": [
            {
                "capitalPoolAddress": "",
                "capitalPoolPlatform": "",
                "capitalPoolCurrency": ""
            }
        ]
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Transaction Module

Query Transaction List by Pagination

url: `/lpcp/v1/fxsp/sys/transaction/page/list`

method: `POST`

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

consumes: `*/*`

Note:

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

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "transactionNo": "",
  "transactionHash": "",
  "fromBnId": "",
  "formPlatform": "",
  "formCurrency": "",
  "toPlatform": "",
  "toCurrency": "",
  "startDate": 0,
  "endDate": 0,
  "commercialBankId": "",
  "commercialBankName": "",
  "receivingBankName": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inTransactionPage	InTransactionPage	body	true	InTransactionPage	InTransactionPage
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
transactionNo	Transaction numbrt		false	string	
transactionHash	Transaction hash		false	string	
fromBnId	From BN ID		false	string	
formPlatform	From platform		false	string	

Name	Description	In	Required	Type	Schema
formCurrency	From currency		false	string	
toPlatform	To platform		false	string	
toCurrency	To currency		false	string	
startDate	Begin date		false	integer(int64)	
endDate	End date		false	integer(int64)	
commercialBankId	Commercial Bank ID		false	string	
commercialBankName	Commercial Bank Name		false	string	
receivingBankName	Receiving Bank Name		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoPage
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	
resultPageInfo		PageInfo	PageInfo
total	Total	integer(int64)	
pages	Pages	integer(int32)	
pageSize	Page size	integer(int32)	
pageNum	Page number	integer(int32)	
lastPage		boolean	
firstPage		boolean	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {},
    "resultPageInfo": {
        "total": 0,
        "pages": 0,
        "pageSize": 0,
        "pageNum": 0,
        "lastPage": true,
        "firstPage": true
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Transaction Details by Transaction Number

url: `/lpcp/v1/fxsp/sys/transaction/detail`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to retreieve the transaction details using its transaction number.

Example:

```
{
  "transactionNo": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inTransactionDetail	InTransactionDetail	body	true	InTransactionDetail	InTransactionDetail
transactionNo	Transaction no		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoTransactionDTO
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		TransactionDTO	TransactionDTO
transactionsInfoId		integer(int64)	
transactionNo	Transaction no.	string	
transactionHash	Transaction hash	string	
sendSysNo		string	
formPlatform	From platform	string	
formCurrency	From currency	string	
fromAccount	From account	string	
fromBankId	From Commercial Bank ID	string	
fromBankName	From Commercial Bank name	string	
fromBnId	From BN ID	string	
fromTxnNo	From transaction no.	string	
fromAmount	From amount	number	
toPlatform	To platform	string	
toCurrency	To currency	string	
toAccount	To account	string	

Name	Description	Type	Schema
toBankId	To bank ID	string	
toBankName	To bank name	string	
toBnId	To BN ID	string	
toTxnNo	To transaction no.	string	
toAmount	To amount	number	
transactionDate	Transaction date	integer(int64)	
settlementModelCode	Settlement model code	string	
chargingModel	Charging Model 1: Proportional charges 2: Fixed charges	integer(int32)	
regularCommission	Fixed commission	number	
ratioCommission	Proportional commission	number	
maxCommission	Maximum commission	number	
commissionPlatform		string	
commissionCurrency		string	
settlementModelName	Settlement model name	string	
spCode		string	
exchangeRate	Exchange rate	number	
status	Status 1: Success 0: Failed	integer(int32)	
fee	Commission fee	number	
commercialBankId	Commercial Bank ID	string	
commercialBankName	Commercial Bank name	string	
receivingBankName	Receiving bank name	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "transactionsInfoId": 0,
        "transactionNo": "",
        "transactionHash": "",
        "sendSysNo": "",
        "formPlatform": "",
        "formCurrency": "",
        "fromAccount": "",
        "fromBankId": "",
        "fromBankName": "",
        "fromBnId": "",
        "fromTxnNo": "",
        "fromAmount": 0,
        "toPlatform": "",
        "toCurrency": "",
        "toAccount": "",
        "toBankId": "",
        "toBankName": "",
        "toBnId": "",
        "toTxnNo": "",
        "toAmount": 0,
        "transactionDate": 0,
        "settlementModelCode": "",
        "chargingModel": 0,
        "regularCommission": 0,
        "ratioCommission": 0,
        "maxCommission": 0,
        "commissionPlatform": "",
        "commissionCurrency": "",
        "settlementModelName": "",
        "spCode": "",
        "exchangeRate": 0,
        "status": 0,
        "fee": 0,
        "commercialBankId": "",
        "commercialBankName": "",
        "receivingBankName": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Token Verification Module

Token Validation and Verification of Menu Permissions

url:`/lpcp/v1/fxsp/sys/token/verify/search`

method:`POST`

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

consumes:`*/*`

Note:

This interface is called to verify and validate the permissions associated with a user's menu access.

Example:

```
{
  "requestUrl": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inTokenVerifySearch	InTokenVerifySearch	body	true	InTokenVerifySearch	InTokenVerifySearch
requestUrl	Request URL		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoBoolean
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	boolean	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

UI Configuration Module

UI Configuration

url: `/lpcp/v1/fxsp/sys/style/manage/submit`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to save and submit the UI configurations.

Example:

```
{
  "systemName": "",
  "logoFileHash": "",
  "themeColor": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inSystemStyleSubmit	InSystemStyleSubmit	body	true	InSystemStyleSubmit	InSystemStyleSubmit
systemName	System subject name		true	string	
logoFileHash	Logo file hash		true	string	
themeColor	Theme color		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoBoolean
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	boolean	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Get sp wallet address

url: `/lpcp/v1/fxsp/sys/sp/getWalletAddress`

method: `POST`

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

consumes: `*/*`

Note:

Query the wallet address of SP on the blockchain

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoString
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	string	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	

Name	Description	Type	Schema
message	Message	string	
data	Api return data	object	

Response Example:

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

Update sp info

url: `/lpcp/v1/fxsp/sys/sp/edit`

method: `POST`

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

consumes: `*/*`

Note:

Update sp info

Example:

```
{
  "spBriefIntroduction": "",
  "spDescription": "",
  "bnCode": "",
  "contactName": "",
  "mobileNumber": "",
  "email": "",
  "detailedAddress": "",
  "businessLicenseUrl": "",
  "spCode": ""
}
```

Params:

name	description	in	require	type	schema
inSpEdit	InSpEdit	body	true	InSpEdit	InSpEdit
spBriefIntroduction	Sp brief introduction		true	string	
spDescription	Sp description		true	string	
bnCode	Bn code		true	string	

name	description	in	require	type	schema
contactName	Contact name		true	string	
mobileNumber	Mobile number		false	string	
email	Email		true	string	
detailedAddress	Detailed address		true	string	
businessLicenseUrl	Business license file hash		true	string	
spCode	Sp code		true	string	

Status:

code	description	schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200:

Response Params:

name	description	type	schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500:

Response Params:

name	description	type	schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query sp info details

url: `/lpcp/v1/fxsp/sys/sp/detail`

method: `POST`

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

consumes: `*/*`

Note:

Query sp info details

Params:

None

Status:

code	description	schema
200	OK	ResultInfoSpInfoDTO
500	Internal Server Error	ResultInfo

code-200:

Response Params:

name	description	type	schema
code	Code	string	
message	Message	string	
data		SpInfoDTO	SpInfoDTO
spCode	Sp code	string	
spName	Sp name	string	
countryInfoId	Country id	integer(int64)	
countryName	Country name	string	
spBriefIntroduction	Sp brief introduction	string	
spDescription	Sp description	string	

name	description	type	schema
spBesuWalletAddress	Sp besu wallet address	string	
bnCode	Bn code	string	
contactName	Contact name	string	
mobileNumber	Mobile number	string	
email	Email	string	
detailedAddress	Detailed address	string	
businessLicenseUrl	Business license file hash	string	
spStatus	Sp status. 5=Pending 10=Approved 15=Rejected	integer(int32)	
remark	Remark	string	
bicName	Bic	string	
centralBankName	Central bank	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "spCode": "",
        "spName": "",
        "countryInfoId": 0,
        "countryName": "",
        "spBriefIntroduction": "",
        "spDescription": "",
        "spBesuWalletAddress": "",
        "bnCode": "",
        "contactName": "",
        "mobileNumber": "",
        "email": "",
        "detailedAddress": "",
        "businessLicenseUrl": "",
        "spStatus": 0,
        "remark": "",
        "bicName": "",
        "centralBankName": ""
    }
}
```

code-500:

Response Params:

name	description	type	schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Register a new SP

url: `/lpcp/v1/fxsp/sys/sp/add`

method: `POST`

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

consumes: `*/*`

Note:

Register a new SP

Example:

```
{
  "spBriefIntroduction": "",
  "spDescription": "",
  "bnCode": "",
  "contactName": "",
  "mobileNumber": "",
  "email": "",
  "detailedAddress": "",
  "businessLicenseUrl": "",
  "spName": "",
  "countryInfoId": 0
}
```

Params:

name	description	in	require	type	schema
inSpAdd	InSpAdd	body	true	InSpAdd	InSpAdd
spBriefIntroduction	Sp brief introduction		true	string	

name	description	in	require	type	schema
spDescription	Sp description		true	string	
bnCode	Bn code		true	string	
contactName	Contact name		true	string	
mobileNumber	Mobile number		false	string	
email	Email		true	string	
detailedAddress	Detailed address		true	string	
businessLicenseUrl	Business license file hash		true	string	
spName	Sp name		true	string	
countryInfoId	Country id		true	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200:

Response Params:

name	description	type	schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500:

Response Params:

name	description	type	schema
code	Code	string	

name	description	type	schema
message	Message	string	
data	Api return data	object	

Response Example:

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

Settlement Model Management Module

Query Settlement Models by Pagination

url: `/lpcp/v1/fxsp/sys/settlement/model/page/list`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the settlement model list by pagination.

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "settlementModelCode": "",
  "settlementModelName": "",
  "formRatePlatform": "",
  "formRateCurrency": "",
  "toRatePlatform": "",
  "toRateCurrency": "",
  "chargingModel": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inSettlementModelPage	InSettlementModelPage	body	true	InSettlementModelPage	InSettlementModelPage
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
settlementModelCode	Settlement model code		false	string	
settlementModelName	Settlement model name		false	string	
formRatePlatform	From platform		false	string	

Name	Description	In	Required	Type	Schema
formRateCurrency	From currency		false	string	
toRatePlatform	To platform		false	string	
toRateCurrency	To currency		false	string	
chargingModel	Charging Model 1: Proportional charges 2: Fixed charges		false	integer(int32)	

Status:

Code	Description	Schema
200	OK	ResultInfoPage
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	
resultPageInfo		PageInfo	PageInfo
total	Total	integer(int64)	
pages	Pages	integer(int32)	
pageSize	Page size	integer(int32)	
pageNum	Page num	integer(int32)	
lastPage		boolean	
firstPage		boolean	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {},
    "resultPageInfo": {
        "total": 0,
        "pages": 0,
        "pageSize": 0,
        "pageNum": 0,
        "lastPage": true,
        "firstPage": true
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Specific Settlement Model History

url: `/lpcp/v1/fxsp/sys/settlement/model/history/list`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the historical data of a specific settlement model.

Example:

```
{
  "formRatePlatform": "",
  "formRateCurrency": "",
  "toRatePlatform": "",
  "toRateCurrency": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inSettlementModelCurrency	InSettlementModelCurrency	body	true	InSettlementModelCurrency	InSettlementModelCurrency
formRatePlatform	From platform		true	string	
formRateCurrency	From currency		true	string	
toRatePlatform	To platform		true	string	
toRateCurrency	To currency		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoListSettlementModelDTO
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	SettlementModelDTO
settlementModelCode	Settlement model code	string	
settlementModelName	Settlement model name	string	
formRatePlatform	From platform	string	
formRateCurrency	From currency	string	
toRatePlatform	To platform	string	
toRateCurrency	To currency	string	
chargingModel	Charging Model 1: Proportional charges 2: Fixed charges	integer(int32)	
regularCommission	Fixed commission	number	

Name	Description	Type	Schema
ratioCommission	Proportional commission	number	
maxCommission	Maximum commission	number	
createDate	Create date	integer(int64)	
updateDate	Update date	integer(int64)	
modifyUser	Updated by	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "settlementModelCode": "",
            "settlementModelName": "",
            "formRatePlatform": "",
            "formRateCurrency": "",
            "toRatePlatform": "",
            "toRateCurrency": "",
            "chargingModel": 0,
            "regularCommission": 0,
            "ratioCommission": 0,
            "maxCommission": 0,
            "createDate": 0,
            "updateDate": 0,
            "modifyUser": ""
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Update Settlement Model Settings

url: `/lpcp/v1/fxsp/sys/settlement/model/edit`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to update the settings of a settlement model.

Example:

```
{
  "formRatePlatform": "",
  "formRateCurrency": "",
  "toRatePlatform": "",
  "toRateCurrency": "",
  "settlementModelName": "",
  "chargingModel": 0,
  "regularCommission": 0,
  "ratioCommission": 0,
  "maxCommission": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inSettlementModelInfo	InSettlementModelInfo	body	true	InSettlementModelInfo	InSettlementModelInfo
formRatePlatform	From platform		true	string	
formRateCurrency	From currency		true	string	
toRatePlatform	To platform		true	string	
toRateCurrency	To currency		true	string	
settlementModelName	Settlement model name		true	string	
chargingModel	Charging Model 1: Proportional charges 2: Fixed charges		true	integer(int32)	
regularCommission	Fixed commission		false	number	
ratioCommission	Proportional commission		false	number	
maxCommission	Maximum commission		false	number	

Status:

Code	Description	Schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Settlement Model Details

url: `/lpcp/v1/fxsp/sys/settlement/model/detail`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the settlement model details.

Example:

```
{
  "formRatePlatform": "",
  "formRateCurrency": "",
  "toRatePlatform": "",
  "toRateCurrency": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inSettlementModelCurrency	InSettlementModelCurrency	body	true	InSettlementModelCurrency	InSettlementModelCurrency
formRatePlatform	From platform		true	string	
formRateCurrency	From currency		true	string	
toRatePlatform	To platform		true	string	
toRateCurrency	To currency		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoSettlementModelDTO
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		SettlementModelDTO	SettlementModelDTO
settlementModelCode	Settlement model code	string	
settlementModelName	Settlement model name	string	
formRatePlatform	From platform	string	
formRateCurrency	From currency	string	
toRatePlatform	To platform	string	
toRateCurrency	To currency	string	

Name	Description	Type	Schema
chargingModel	Charging Model 1: Proportional charges 2: Fixed charges	integer(int32)	
regularCommission	Fixed commission	number	
ratioCommission	Proportional commission	number	
maxCommission	Maximum commission	number	
createDate	Create date	integer(int64)	
updateDate	Update date	integer(int64)	
modifyUser	Updated by	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "settlementModelCode": "",
        "settlementModelName": "",
        "formRatePlatform": "",
        "formRateCurrency": "",
        "toRatePlatform": "",
        "toRateCurrency": "",
        "chargingModel": 0,
        "regularCommission": 0,
        "ratioCommission": 0,
        "maxCommission": 0,
        "createDate": 0,
        "updateDate": 0,
        "modifyUser": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Create New Settlement Model

url: `/lpcp/v1/fxsp/sys/settlement/model/add`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to create a new settlement model.

Example:

```
{
  "formRatePlatform": "",
  "formRateCurrency": "",
  "toRatePlatform": "",
  "toRateCurrency": "",
  "settlementModelName": "",
  "chargingModel": 0,
  "regularCommission": 0,
  "ratioCommission": 0,
  "maxCommission": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inSettlementModelInfo	InSettlementModelInfo	body	true	InSettlementModelInfo	InSettlementModelInfo
formRatePlatform	From platform		true	string	
formRateCurrency	From currency		true	string	
toRatePlatform	To platform		true	string	
toRateCurrency	To currency		true	string	
settlementModelName	Settlement model name		true	string	
chargingModel	Charging Model 1: Proportional charges 2: Fixed charges		true	integer(int32)	
regularCommission	Fixed commission		false	number	
ratioCommission	Proportional commission		false	number	
maxCommission	Maximum commission		false	number	

Status:

Code	Description	Schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Role Management Module

Activate/Deactivate Role

url: `/lpcp/v1/fxsp/sys/role/manage/status/update`

method: `POST`

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

consumes:`*/*`

Note:

This interface is called to activate or deactivate the role.

Example:

```
{
  "roleCode": "",
  "lockable": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inRoleStatusUpdate	InRoleStatusUpdate	body	true	InRoleStatusUpdate	InRoleStatusUpdate
roleCode	Role code		true	string	
lockable	Action 1: Deactivate 2: Activate 3: Delete		true	integer(int32)	

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Role List by Pagination

url: `/lpcp/v1/fxsp/sys/role/manage/searches`

method: `POST`

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

consumes: `*/*`

Note:

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

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "roleName": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inRoleSearches	InRoleSearches	body	true	InRoleSearches	InRoleSearches
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
roleName	Role name		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoListOutRoleSearches
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutRoleSearches
roleId	Role ID	integer(int64)	
roleName	Role name	string	
roleCode	Role code	string	
description	Description	string	
lockable	Status 1: Inactive 2: Active 3: Deleted	integer(int32)	
superRole	Superadmin 1: Yes 2: No	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "roleId": 0,
            "roleName": "",
            "roleCode": "",
            "description": "",
            "lockable": 0,
            "superRole": 0
        }
    ]
}
```

code-500:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Create a Role

url: `/lpcp/v1/fxsp/sys/role/manage/save`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to create a new role.

Example:

```
{
  "roleName": "",
  "description": "",
  "lockable": 0,
  "rsucIdList": []
}
```

Params:

Name	Description	In	Required	Type	Schema
inRoleSave	InRoleSave	body	true	InRoleSave	InRoleSave
roleName	Role name		true	string	
description	Role description		false	string	
lockable	Status 1: Deactivate 2: Activate 3: Delete		true	integer(int32)	

Name	Description	In	Required	Type	Schema
rsucIdList	Resource ID list		true	array	integer(int64)

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Update Role

url: `/lpcp/v1/fxsp/sys/role/manage/edit`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to save modiciations made to existing role.

Example:

```
{
  "roleCode": "",
  "roleName": "",
  "description": "",
  "lockable": 0,
  "rsucIdList": []
}
```

Params:

Name	Description	In	Required	Type	Schema
inRoleEdit	InRoleEdit	body	true	InRoleEdit	InRoleEdit
roleCode	Role code		true	string	
roleName	Role name		false	string	
description	Role description		false	string	
lockable	Status 1: Inactive 2: Active 3: Deleted		false	integer(int32)	
rsucIdList	Resource ID list		false	array	integer(int64)

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Role Details

url: `/lpcp/v1/fxsp/sys/role/manage/edit/detail/search`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the role details.

Example:

```
{
  "roleCode": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inRoleEditDetailSearch	InRoleEditDetailSearch	body	true	InRoleEditDetailSearch	InRoleEditDetailSearch
roleCode	Role code		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutRoleEditDetailSearch
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutRoleEditDetailSearch	OutRoleEditDetailSearch
roleId	Role ID	integer(int64)	
roleName	Role name	string	
roleCode	Role code	string	
description	Description	string	
lockable	Status 1: Inactive 2: Active 3: Deleted	integer(int32)	
myResourceList	My resource list	array	SysResourceTreeExtended
rsucId		integer	
parentId		integer	
rsucCode		string	
rsucName		string	
rsucUrl		string	
priority		integer	
createDate		integer	
updateDate		integer	

Name	Description	Type	Schema
lockable		integer	
rsucType		integer	
ischecked		integer	
bankPermissionType		integer	
systemType		integer	
childNodes	Child node	array	SysResourceTreeExtended

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "roleId": 0,
        "roleName": "",
        "roleCode": "",
        "description": "",
        "lockable": 0,
        "myResourceList": [
            {
                "rsucId": 0,
                "parentId": 0,
                "rsucCode": "",
                "rsucName": "",
                "rsucUrl": "",
                "priority": 0,
                "createDate": 0,
                "updateDate": 0,
                "lockable": 0,
                "rsucType": 0,
                "ischecked": 0,
                "bankPermissionType": 0,
                "systemType": 0,
                "childNodes": [
                    {
                        "rsucId": 0,
                        "parentId": 0,
                        "rsucCode": "",
                        "rsucName": "",
                        "rsucUrl": "",
                        "priority": 0,
                        "createDate": 0,
                        "updateDate": 0,
                        "lockable": 0,
                        "rsucType": 0,
                        "ischecked": 0,
                        "bankPermissionType": 0,
                        "systemType": 0,
                        "childNodes": [
                            {}
                        ]
                    }
                ]
            }
        ]
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Role List

url: `/lpcp/v1/fxsp/sys/role/manage/all/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the role list.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoListOutRoleSearches
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutRoleSearches
roleId	Role ID	integer(int64)	
roleName	Role name	string	
roleCode	Role code	string	
description	Description	string	
lockable	Status 1: Inactive 2: Active 3: Deleted	integer(int32)	
superRole	Superadmin 1: Yes 2: No	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "roleId": 0,
            "roleName": "",
            "roleCode": "",
            "description": "",
            "lockable": 0,
            "superRole": 0
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Password Management Module

Verify the Password

url: `/lpcp/v1/fxsp/sys/password/verify`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to confirm the validity of a user's password.

Example:

```
{
  "encryptionStr": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inPasswordVerify	InPasswordVerify	body	true	InPasswordVerify	InPasswordVerify
encryptionStr	The encrypted string		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoBoolean
500	Internal Server Error	ResultInfo

code-200:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	

Name	Description	Type	Schema
data	Api return data	boolean	

Response Example:

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

code-500:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Dashoboard Module

Query the Transaction Volume in Last 7 Days

method:`POST`

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

consumes:`*/*`

Note:

This interface is called to query the transaction volume in the last 7 days.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoListOutHistoryTransactionVolumeDynamicsInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutHistoryTransactionVolumeDynamicsInfo
transactionDate	Transaction date	integer(int64)	
outTransactionVolumeInfoList	Transaction volume	array	OutTransactionVolumeInfo
transactionNumber	Transaction number	integer	
sourceCurrency	Source currency name	string	
sourcePlatform	Source platform name	string	
targetCurrency	Target currency name	string	
targetPlatform	Target platform name	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "transactionDate": 0,
            "outTransactionVolumeInfoList": [
                {
                    "transactionNumber": 0,
                    "sourceCurrency": "",
                    "sourcePlatform": "",
                    "targetCurrency": "",
                    "targetPlatform": ""
                }
            ]
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Check the exchange rate under review

url: `/lpcp/v1/fxsp/sys/home/pending/rate`

method: `POST`

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

consumes: `*/*`

Note:

Query the exchange rate under review

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutPendingRate
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutPendingRate	OutPendingRate
outPendingRateListList	From currency name	array	OutPendingRateList

Name	Description	Type	Schema
fromCurrency	From currency name	string	
fromPlatform	From platform name	string	
toCurrency	To currency name	string	
toPlatform	To platform name	string	
rateId	Rate ID	string	
count	Number	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "outPendingRateListList": [
            {
                "fromCurrency": "",
                "fromPlatform": "",
                "toCurrency": "",
                "toPlatform": "",
                "rateId": ""
            }
        ],
        "count": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query the currency under review

url: `/lpcp/v1/fxsp/sys/home/pending/currency`

method: `POST`

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

consumes: `*/*`

Note:

Query the currency under review

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutPendingCurrency
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutPendingCurrency	OutPendingCurrency
outPendingCurrencyList	Currency name	array	OutPendingCurrencyList
currency	Currency name	string	
platform	Platform name	string	
address	Capital pool address	string	
count	Number	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "outPendingCurrencyList": [
            {
                "currency": "",
                "platform": "",
                "address": ""
            }
        ],
        "count": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Get unopened currency pair

url: `/lpcp/v1/fxsp/sys/home/nonactivated/rate`

method: `POST`

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

consumes: `*/*`

Note:

Query platform support for currencies that have not yet been opened

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutUnopenedRate
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutUnopenedRate	OutUnopenedRate
unopenedRateList	Unopened rate list	array	string

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "unopenedRateList": "[w-THB-UDPN->w-EUR-UDPN,w-THB-UDPN->w-USD-UDPN]"
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query unopened currencies

url: `/lpcp/v1/fxsp/sys/home/nonactivated/currency`

method: `POST`

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

consumes: `*/*`

Note:

Query the currencies that can open a fund pool

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutUnopenedCurrenciesInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutUnopenedCurrenciesInfo	OutUnopenedCurrenciesInfo
platformCurrency	Platform currency	array	OutCurrenciesInfo
currency	Currency name	string	
platform	Platform name	string	
count	Count	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "platformCurrency": [
            {
                "currency": "",
                "platform": ""
            }
        ],
        "count": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	

Name	Description	Type	Schema
data	Api return data	object	

Response Example:

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

Obtain the Latest Exchange Rate

url: `/lpcp/v1/fxsp/sys/home/latest/exchange/rate/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to retrieve the most recent exchange rate for all currency pairs available.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutApproveExchangeRateInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutApproveExchangeRateInfo	OutApproveExchangeRateInfo
outLatestExchangeRateInfoList	Approve exchange rate	array	OutLatestExchangeRateInfo
sourceCurrency	Source currency name	string	
sourcePlatform	Source platform name	string	

Name	Description	Type	Schema
targetCurrency	Target currency name	string	
targetPlatform	Target platform name	string	
exchangeRate	Exchange rate	number	
date	Date	integer	
chainRateId	Rate ID	string	
settlementModelCode	Settlement model code	string	
count	Count	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "outLatestExchangeRateInfoList": [
            {
                "sourceCurrency": "",
                "sourcePlatform": "",
                "targetCurrency": "",
                "targetPlatform": "",
                "exchangeRate": 0,
                "date": 0,
                "chainRateId": "",
                "settlementModelCode": ""
            }
        ],
        "count": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query the Exchange Rate Dynamics in Last 7 Day

url: `/lpcp/v1/fxsp/sys/home/history/exchange/rate/dynamics/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the exchange rate dynamics in the last 7 days.

Example:

```
{
  "sourceCurrency": "",
  "sourcePlatform": "",
  "targetCurrency": "",
  "targetPlatform": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inHistoryExchangeRateDynamicsInfo	inHistoryExchangeRateDynamicsInfo	body	true	InHistoryExchangeRateDynamicsInfo	InHistoryExchangeRateDynamicsInfo
sourceCurrency	Source currency name		true	string	
sourcePlatform	Source platform name		true	string	
targetCurrency	Target currency name		true	string	
targetPlatform	Target platform name		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoListOutHistoryExchangeRateDynamicsInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	

Name	Description	Type	Schema
data	Api return data	array	OutHistoryExchangeRateDynamicsInfo
sourceCurrency	Source currency name	string	
sourcePlatform	Source platform name	string	
targetCurrency	Target currency name	string	
targetPlatform	Target platform name	string	
historyExchangeRateInfoList	Exchange rate	array	OutHistoryExchangeRateInfo
exchangeRate	Exchange rate	number	
date	Date	integer	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "sourceCurrency": "",
            "sourcePlatform": "",
            "targetCurrency": "",
            "targetPlatform": "",
            "historyExchangeRateInfoList": [
                {
                    "exchangeRate": 0,
                    "date": 0
                }
            ]
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query the currency that has been opened

url: `/lpcp/v1/fxsp/sys/home/approve/currency`

method: `POST`

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

consumes: `*/*`

Note:

Query the currency that has been opened

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutApproveCapitalPool
500	Internal Server Error	ResultInfo

code-200:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutApproveCapitalPool	OutApproveCapitalPool
outCapitalPoolSearchesList	Currency name	array	OutCapitalPoolSearches
capitalPoolAddress	Capital Pool address	string	
capitalPoolPlatform	Capital Pool platform	string	
capitalPoolCurrency	Capital Pool currency	string	
capitalPoolBalance	Capital Pool balance	number	
createDate	Created on	integer	

Name	Description	Type	Schema
status	Status 0: Pending 5: Approved 10: Rejected	integer	
count	Count	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "outCapitalPoolSearchesList": [
            {
                "capitalPoolAddress": "",
                "capitalPoolPlatform": "",
                "capitalPoolCurrency": "",
                "capitalPoolBalance": 0,
                "createDate": 0,
                "status": 0
            }
        ],
        "count": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Historical rate manage

Select historical rate

url: `/lpcp/v1/fxsp/sys/history/rate/searches`

method: `POST`

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

consumes: `*/*`

Note:

Query historical exchange rate information for each exchange rate pair

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "formRatePlatform": "",
  "formRateCurrency": "",
  "toRatePlatform": "",
  "toRateCurrency": "",
  "startDate": 0,
  "endDate": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inHistoricalRateSearches	InHistoricalRateSearches	body	true	InHistoricalRateSearches	InHistoricalRateSearches
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
formRatePlatform	From rate platform		false	string	
formRateCurrency	From Rate currency		false	string	
toRatePlatform	To rate platform		false	string	
toRateCurrency	To rate currency		false	string	
startDate	Start date		false	integer(int64)	
endDate	End date		false	integer(int64)	

Status:

Code	Description	Schema
200	OK	ResultInfoListOutHistoricalRateSearches
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutHistoricalRateSearches
formRatePlatform	From rate platform	string	
formRateCurrency	From rate currency	string	
toRatePlatform	To rate platform	string	
toRateCurrency	To rate currency	string	
exchangeRate	Exchange rate	number	
rateDate	Rate date	integer(int64)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "formRatePlatform": "",
            "formRateCurrency": "",
            "toRatePlatform": "",
            "toRateCurrency": "",
            "exchangeRate": 0,
            "rateDate": 0
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

File manage

Upload File

url: `/lpcp/v1/fxsp/sys/file/upload`

method: `POST`

produces: `application/x-www-form-urlencoded,multipart/form-data`

consumes: `*/*`

Note:

This interface is called to upload a file. Upon a successful upload, it returns a string that serves as the file key for the uploaded file.

Params:

Name	Description	In	Required	Type	Schema
file		query	true	file	

Status:

Code	Description	Schema
200	OK	ResultInfoString
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	string	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Download File with File Information

url: `/lpcp/v1/fxsp/sys/file/extension/download`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to download a file by its key. It returns the base64 encoding of the file, along with its suffix and other relevant information.

Example:

```
{
  "hash": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inFileDownload	InFileDownload	body	true	InFileDownload	InFileDownload
hash	File key		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutFileDownload
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutFileDownload	OutFileDownload
fileName	File name	string	
fileExtension	Extension	string	
data	Data	string	

Response Example:

```json
{
    "code": "",
    "message": "",
    "data": {
        "fileName": "",
        "fileExtension": "",
        "data": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Download File

url: `/lpcp/v1/fxsp/sys/file/download`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to download a file by its key. It only returns the base64 encoding of the file.

Example:

```
{
  "hash": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inFileDownload	InFileDownload	body	true	InFileDownload	InFileDownload
hash	File key		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

FX Pair Management Module

Query sp basic information in exchange application detail

url: `/lpcp/v1/fxsp/sys/exchange/spinfo/search`

method: `POST`

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

consumes: `*/*`

Note:

Query sp basic information in exchange application detail.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutSelectExchange
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutSelectExchange	OutSelectExchange
spName	Sp name	string	
country	Country of region	string	
introduction	Brief introduction	string	
description	Description	string	
applicationTime	Application Time	integer(int64)	
status	Status:0=Pending 1=Approved 2=Rejected 3=Repealed	integer(int32)	
approvalResult	Approval result:0=Pending 1=Approved 2=Rejected 3=Repealed	integer(int32)	
comments	Comments	string	
spBesuWalletAddress	Sp besu wallet address	string	
bnCode	Bn code	string	
creationTime	Creation time	integer(int64)	
approvalTime	Approval time	integer(int64)	
businessApplicationCode	Applicaiton no	string	
outSettlementInformations	Settlement information	array	OutSettlementInformation
currency	Currency	string	
capitalPoolAddress	Capital pool address	string	
outExchangeInformations	Foreign Exchange Information	array	OutExchangeInformation
sourceCurrency	Source currency	string	
targetCurrency	Target currency	string	
applicationType	Application type:0=Activate 1=Reduce	integer(int32)	
bicName	Bic	string	
centralBankName	Central bank	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "spName": "",
        "country": "",
        "introduction": "",
        "description": "",
        "applicationTime": 0,
        "status": 0,
        "approvalResult": 0,
        "comments": "",
        "spBesuWalletAddress": "",
        "bnCode": "",
        "creationTime": 0,
        "approvalTime": 0,
        "businessApplicationCode": "",
        "outSettlementInformations": [
            {
                "currency": "",
                "capitalPoolAddress": ""
            }
        ],
        "outExchangeInformations": [
            {
                "sourceCurrency": "",
                "targetCurrency": ""
            }
        ],
        "applicationType": 0,
        "bicName": "",
        "centralBankName": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query foreign exchange application list

url: `/lpcp/v1/fxsp/sys/exchange/searches`

method: `POST`

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

consumes: `*/*`

Note:

Query foreign exchange application list

Example:

```json
{
  "pageNum": 0,
  "pageSize": 0,
  "businessApplicationCode": "",
  "businessApplicationStatus": 0,
  "startApplicationDate": 0,
  "endApplicationDate": 0,
  "startApprovalDate": 0,
  "endApprovalDate": 0,
  "businessType": 0,
  "applicationType": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inSelectExchangeList	InSelectExchangeList	body	true	InSelectExchangeList	InSelectExchangeList
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
businessApplicationCode	Business application code		false	string	
businessApplicationStatus	Business Application Status 0: Pending 1: Approved 2: Rejected 3: Repealed		false	integer(int32)	
startApplicationDate	Application start date		false	integer(int64)	
endApplicationDate	Application end date		false	integer(int64)	
startApprovalDate	Approval start date		false	integer(int64)	
endApprovalDate	Approval end date		false	integer(int64)	
businessType	Business Type 0: Rate 1: Capital Pool		false	integer(int32)	
applicationType	Application Type 0: Activate 1: Reduce		false	integer(int32)	

Status:

Code	Description	Schema
200	OK	ResultInfoListOutSelectExchangeList
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutSelectExchangeList
applicationNo	Application no.	string	
currency	Currency	array	string
applicationTime	Application time	integer(int64)	
approvalTime	Approval time	integer(int64)	
status	Status 0: Pending 1: Approved 2: Rejected 3: Repealed	integer(int32)	
applicationType	Application Type 0: Activate 1: Reduce	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "applicationNo": "",
            "currency": [],
            "applicationTime": 0,
            "approvalTime": 0,
            "status": 0,
            "applicationType": 0
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query exchange application detail

url: `/lpcp/v1/fxsp/sys/exchange/search`

method: `POST`

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

consumes: `*/*`

Note:

Query exchange application detail

Example:

```
{
  "businessApplicationCode": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inSelectExchange	InSelectExchange	body	true	InSelectExchange	InSelectExchange
businessApplicationCode	Business application code		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutSelectExchange
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutSelectExchange	OutSelectExchange
spName	SP name	string	
country	Region	string	
introduction	Brief introduction	string	
description	Description	string	
applicationTime	Application time	integer(int64)	
status	Status 0: Pending 1: Approved 2: Rejected 3: Repealed	integer(int32)	
approvalResult	Approval result:0=Pending 1=Approved 2=Rejected 3=Repealed	integer(int32)	
comments	Comments	string	
spBesuWalletAddress	SP wallet address	string	
bnCode	BN code	string	
creationTime	Creation time	integer(int64)	
approvalTime	Approval time	integer(int64)	
businessApplicationCode	Applicaiton no.	string	
outSettlementInformations	Settlement information	array	OutSettlementInformation
currency	Currency	string	
capitalPoolAddress	Capital Pool address	string	
outExchangeInformations	Foreign Exchange Information	array	OutExchangeInformation
sourceCurrency	Source currency	string	
targetCurrency	Target currency	string	
applicationType	Application type:0=Activate 1=Reduce	integer(int32)	
bicName	BIc name	string	
centralBankName	Central Bank name	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "spName": "",
        "country": "",
        "introduction": "",
        "description": "",
        "applicationTime": 0,
        "status": 0,
        "approvalResult": 0,
        "comments": "",
        "spBesuWalletAddress": "",
        "bnCode": "",
        "creationTime": 0,
        "approvalTime": 0,
        "businessApplicationCode": "",
        "outSettlementInformations": [
            {
                "currency": "",
                "capitalPoolAddress": ""
            }
        ],
        "outExchangeInformations": [
            {
                "sourceCurrency": "",
                "targetCurrency": ""
            }
        ],
        "applicationType": 0,
        "bicName": "",
        "centralBankName": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Submit an application for opening foreign exchange business

url: `/lpcp/v1/fxsp/sys/exchange/save`

method: `POST`

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

consumes: `*/*`

Note:

Submit an application for opening foreign exchange business

Example:

```
{
  "businessType": 0,
  "settlementInformations": [
    {
      "capitalPoolPlatform": "",
      "capitalPoolCurrency": "",
      "capitalPoolAddress": ""
    }
  ],
  "exchangeInformations": [
    {
      "fromPlatform": "",
      "fromCurrency": "",
      "toPlatform": "",
      "toCurrency": ""
    }
  ]
}
```

Params:

Name	Description	In	Required	Type	Schema
inSaveExchange	InSaveExchange	body	true	InSaveExchange	InSaveExchange
businessType	Business Type 0: Rate 1: Capital Pool		true	integer(int32)	
settlementInformations	Settlement information		false	array	InSettlementInformation
capitalPoolPlatform	Capital Pool platform		false	string	
capitalPoolCurrency	Capital Pool currency		false	string	
capitalPoolAddress	Capital Pool address		false	string	
exchangeInformations	Foreign exchange information		false	array	InExchangeInformation
fromPlatform	From platform		false	string	

Name	Description	In	Required	Type	Schema
fromCurrency	From currency		false	string	
toPlatform	To platform		false	string	
toCurrency	To currency		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutSaveExchange
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutSaveExchange	OutSaveExchange
saveStatus	Status 0: Success 1: Failed	integer(int32)	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Delete exchange rate

url: `/lpcp/v1/fxsp/sys/exchange/delete/exchange/pair`

method: `POST`

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

consumes: `*/*`

Note:

Delete exchange rate

Example:

```
{
  "exchangeInformations": [
    {
      "fromPlatform": "",
      "fromCurrency": "",
      "toPlatform": "",
      "toCurrency": ""
    }
  ]
}
```

Params:

Name	Description	In	Required	Type	Schema
inDeleteExchange	InDeleteExchange	body	true	InDeleteExchange	InDeleteExchange
exchangeInformations	Foreign exchange information		false	array	InExchangeInformation
fromPlatform	From platform		false	string	
fromCurrency	From currency		false	string	
toPlatform	To platform		false	string	
toCurrency	To currency		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoObject
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query the exchange rate business that sp has opened

url: `/lpcp/v1/fxsp/sys/exchange/approved/sp/info/search`

method: `POST`

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

consumes: `*/*`

Note:

Query the exchange rate business that sp has opened

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutApprovedSpInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutApprovedSpInfo	OutApprovedSpInfo
spName	SP name	string	
country	Region	string	
introduction	Brief introduction	string	
description	Description	string	
spBesuWalletAddress	SP wallet address	string	
bnCode	BN code	string	
outCapitalPool	Settlement information	array	OutCapitalPool
capitalPoolPlatform	Capital Pool platform	string	
capitalPoolCurrency	Capital Pool currency	string	
capitalPoolAddress	Capital Pool address	string	
outExchangeRate	Foreign exchange information	array	OutExchangeRate
formPlatform	From platform	string	
formCurrency	From currency	string	
toPlatform	To Platform	string	
toCurrency	To Currency	string	
bicName	BIC name	string	

Name	Description	Type	Schema
centralBankName	Central Bank name	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "spName": "",
        "country": "",
        "introduction": "",
        "description": "",
        "spBesuWalletAddress": "",
        "bnCode": "",
        "outCapitalPool": [
            {
                "capitalPoolPlatform": "",
                "capitalPoolCurrency": "",
                "capitalPoolAddress": ""
            }
        ],
        "outExchangeRate": [
            {
                "formPlatform": "",
                "formCurrency": "",
                "toPlatform": "",
                "toCurrency": ""
            }
        ],
        "bicName": "",
        "centralBankName": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Check if there are approved business applications

url: `/lpcp/v1/fxsp/sys/exchange/approved/search`

method: `POST`

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

consumes: `*/*`

Note:

Check if there are approved business applications

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutExamineStatus
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutExamineStatus	OutExamineStatus
examineStatus	Status 0: Success 1: Failed	integer(int32)	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Capital Pool Management Module

Query Capital Pool List

url: `/lpcp/v1/fxsp/sys/capital/pool/searches`

method: `POST`

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

consumes: `*/*`

Note:

Find all capital pools

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoListOutCapitalPoolInfoSearches
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutCapitalPoolInfoSearches
capitalPoolAddress	Capital Pool address	string	
capitalPoolPlatform	Capital Pool platform	string	
capitalPoolCurrency	Capital Pool currency	string	
capitalPoolBalance	Capital Pool authorized balance	string	
capitalPoolRealBalance	Capital Pool actual balance	string	
createDate	Created on	integer(int64)	
currencyPrecision	Currency precision	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "capitalPoolAddress": "",
            "capitalPoolPlatform": "",
            "capitalPoolCurrency": "",
            "capitalPoolBalance": "",
            "capitalPoolRealBalance": "",
            "createDate": 0,
            "currencyPrecision": 0
        }
    ]
}
```

code-500

Response Params:

name	description	type	schema
code	Code	string	

name	description	type	schema
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Unused Capital Pool

url: `/lpcp/v1/fxsp/sys/capital/pool/delete/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to retrieve the Capital Pool list that are not in use.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoListOutCanDeleteCapitalPoolInfoSearches
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutCanDeleteCapitalPoolInfoSearches
capitalPoolAddress	Capital Pool address	string	

Name	Description	Type	Schema
capitalPoolPlatform	Capital Pool platform	string	
capitalPoolCurrency	Capital Pool currency	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "capitalPoolAddress": "",
            "capitalPoolPlatform": "",
            "capitalPoolCurrency": ""
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Capital Pool List by Pagination

url: `/lpcp/v1/fxsp/sys/capital/pool/application/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the Capital Pool list by pagination.

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "businessApplicationCode": "",
  "businessApplicationStatus": 0,
  "startApplicationDate": 0,
  "endApplicationDate": 0,
  "startApprovalDate": 0,
  "endApprovalDate": 0,
  "applicationType": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inSelectCapitalPoolApplicationList	InSelectCapitalPoolApplicationList	body	true	InSelectCapitalPoolApplicationList	InSelectCapitalPoolApplicationList
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
businessApplicationCode	Business application code		false	string	
businessApplicationStatus	Business Application Status 0: Pending 1: Approved 2: Rejected 3: Repealed		false	integer(int32)	
startApplicationDate	Application start date		false	integer(int64)	
endApplicationDate	Application end date		false	integer(int64)	
startApprovalDate	Approval start date		false	integer(int64)	
endApprovalDate	Approval end date		false	integer(int64)	
applicationType	Application type:0=Activate 1=Reduce		true	integer(int32)	

Status:

Code	Description	Schema
200	OK	ResultInfoListOutSelectCapitalPoolApplicationList
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	OutSelectCapitalPoolApplicationList
applicationNo	Application no.	string	
currency	Currency	array	string

Name	Description	Type	Schema
applicationTime	Application time	integer(int64)	
approvalTime	Approval time	integer(int64)	
applicationType	Application type:0=Activate 1=Reduce	integer(int32)	
status	Status 0: Pending 1: Approved 2: Rejected 3: Repealed	integer(int32)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "applicationNo": "",
            "currency": [],
            "applicationTime": 0,
            "approvalTime": 0,
            "applicationType": 0,
            "status": 0
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Details of Capital Pool Application

url: `/lpcp/v1/fxsp/sys/capital/pool/application/detail`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the detailed information about a specific capital pool application.

Example:

```
{
  "businessApplicationCode": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inSelectExchange	InSelectExchange	body	true	InSelectExchange	InSelectExchange
businessApplicationCode	Business application code		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutCapitalPoolApplicationDetail
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutCapitalPoolApplicationDetail	OutCapitalPoolApplicationDetail
spName	SP name	string	
country	Region	string	
introduction	Brief introduction	string	
description	Description	string	
applicationTime	Application time	integer(int64)	
approvalTime	Approval time	integer(int64)	

Name	Description	Type	Schema
status	Status 0: Pending 1: Approved 2: Rejected 3: Repealed	integer(int32)	
approvalResult	Approval Result 0: Pending 1: Approved 2: Rejected 3: Repealed	integer(int32)	
comments	Comments	string	
spBesuWalletAddress	SP wallet address	string	
bnCode	BN code	string	
creationTime	Created on	integer(int64)	
businessApplicationCode	Applicaiton no.	string	
outPlatformCurrencyInformations	Platform currency information	array	OutPlatformCurrencyInformation
platformCurrency	Platform currency	string	
capitalPoolAddress	Capital Pool address	string	
applicationType	Application type:0=Activate 1=Reduce	integer(int32)	
bicName	BIC nane	string	
centralBankName	Central Bank name	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "spName": "",
        "country": "",
        "introduction": "",
        "description": "",
        "applicationTime": 0,
        "approvalTime": 0,
        "status": 0,
        "approvalResult": 0,
        "comments": "",
        "spBesuWalletAddress": "",
        "bnCode": "",
        "creationTime": 0,
        "businessApplicationCode": "",
        "outPlatformCurrencyInformations": [
            {
                "platformCurrency": "",
                "capitalPoolAddress": ""
            }
        ],
        "applicationType": 0,
        "bicName": "",
        "centralBankName": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Delete Capital Pool

url: `/lpcp/v1/fxsp/sys/capital/pool/application/delete`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to delete existing Capital Pool.

Example:

```
{
  "settlementInformations": [
    {
      "capitalPoolPlatform": "",
      "capitalPoolCurrency": "",
      "capitalPoolAddress": ""
    }
  ]
}
```

Params:

Name	Description	In	Required	Type	Schema
inDeleteCapitalPool	InDeleteCapitalPool	body	true	InDeleteCapitalPool	InDeleteCapitalPool
settlementInformations	Settlement information		false	array	InSettlementInformation
capitalPoolPlatform	Capital Pool platform		false	string	

Name	Description	In	Required	Type	Schema
capitalPoolCurrency	Capital Pool currency		false	string	
capitalPoolAddress	Capital Pool address		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutDeleteCapitalPool
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutDeleteCapitalPool	OutDeleteCapitalPool
deleteStatus	Status 0: Success 1: Failed	integer(int32)	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Settlement Management Module

This module is exclusively available to a Commercial Bank that has registered as a Liquidity Provider.

Query FX Transaction List by Bill Number

url: `/lpcp/v1/fxsp/sys/bill/transaction/detail`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the FX transaction list using a specific bill number.

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "billNo": "",
  "transactionNo": "",
  "fromBnId": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inBillingTransactionDetailIPage	InBillingTransactionDetailIPage	body	true	InBillingTransactionDetailIPage	InBillingTransactionDetailIPage
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
billNo	Bill no.		true	string	
transactionNo	Transaction no.		false	string	
fromBnId	From BN ID		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoPage
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	
resultPageInfo		PageInfo	PageInfo
total	Total	integer(int64)	
pages	Pages	integer(int32)	
pageSize	Page size	integer(int32)	
pageNum	Page num	integer(int32)	
lastPage		boolean	
firstPage		boolean	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {},
    "resultPageInfo": {
        "total": 0,
        "pages": 0,
        "pageSize": 0,
        "pageNum": 0,
        "lastPage": true,
        "firstPage": true
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query Monthly Income Statement List by Pagination

url: `/lpcp/v1/fxsp/sys/bill/page/list`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to query the monthly income statement list based on specified parameters by pagination.

Example:

```
{
  "pageNum": 0,
  "pageSize": 0,
  "billNo": "",
  "billCycle": 0,
  "bankId": "",
  "bankName": "",
  "formPlatform": "",
  "formCurrency": "",
  "toPlatform": "",
  "toCurrency": "",
  "startDate": 0,
  "endDate": 0
}
```

Params:

Name	Description	In	Required	Type	Schema
inBillingIPage	InBillingIPage	body	true	InBillingIPage	InBillingIPage
pageNum	Page number		true	integer(int32)	
pageSize	Page size		true	integer(int32)	
billNo	Bill no.		false	string	
billCycle	Bill cycle		false	integer(int64)	
bankId	Bank ID		false	string	
bankName	Bank name		false	string	

Name	Description	In	Required	Type	Schema
formPlatform	From platform		false	string	
formCurrency	From currency		false	string	
toPlatform	To platform		false	string	
toCurrency	To currency		false	string	
startDate	Start date		false	integer(int64)	
endDate	End date		false	integer(int64)	

Status:

Code	Description	Schema
200	OK	ResultInfoPage
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	
resultPageInfo		PageInfo	PageInfo
total	Total	integer(int64)	
pages	Pages	integer(int32)	
pageSize	Page size	integer(int32)	
pageNum	Page num	integer(int32)	
lastPage		boolean	
firstPage		boolean	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {},
    "resultPageInfo": {
        "total": 0,
        "pages": 0,
        "pageSize": 0,
        "pageNum": 0,
        "lastPage": true,
        "firstPage": true
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query the Invoice Information by Bill Number

url: `/lpcp/v1/fxsp/sys/bill/invoice/select`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to get the monthly invoice using a specific bill number.

Example:

```
{
  "billNo": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inBillingInvoiceSelect	InBillingInvoiceSelect	body	true	InBillingInvoiceSelect	InBillingInvoiceSelect
billNo	Bill no		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutBillingDetail
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutBillingDetail	OutBillingDetail
billInfoId	Bill info ID	integer(int64)	
billNo	Bill no.	string	
billCycle	Bill cycle	integer(int64)	
bankId	Bank ID	string	
bankName	Bank name	string	
formPlatform	From platform	string	
formCurrency	From currency	string	
toPlatform	To platform	string	
toCurrency	To currency	string	
billAmount	Bill amount	number	
createDate	Created on	integer(int64)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "billInfoId": 0,
        "billNo": "",
        "billCycle": 0,
        "bankId": "",
        "bankName": "",
        "formPlatform": "",
        "formCurrency": "",
        "toPlatform": "",
        "toCurrency": "",
        "billAmount": 0,
        "createDate": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Export Invoice as PDF

url: `/lpcp/v1/fxsp/sys/bill/invoice/export`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to export the invoice as PDF.

Example:

```
{
  "billNo": "",
  "recipientList": []
}
```

Params:

Name	Description	In	Required	Type	Schema
inBillingInvoiceExport	InBillingInvoiceExport	body	true	InBillingInvoiceExport	InBillingInvoiceExport
billNo	Bill no.		true	string	
recipientList	Recipient list		false	array	string

Status:

Code	Description	Schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500:

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Export FX Transaction List as Excel

url: `/lpcp/v1/fxsp/sys/bill/export`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to export the list of FX transactions in excel format.

Example:

```json
{
  "minId": 0,
  "maxId": 0,
  "billNo": "",
  "transactionNo": "",
  "bnId": "",
  "recipientList": []
}
```

Params:

Name	Description	In	Required	Type	Schema
inBillingExport	InBillingExport	body	true	InBillingExport	InBillingExport
minId			false	integer(int64)	
maxId			false	integer(int64)	
billNo	Bill no.		true	string	
transactionNo	Transaction no.		false	string	
bnId	BN ID		false	string	
recipientList	Recipient list		false	array	string

Status:

Code	Description	Schema
200	OK	ResultInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Query billing details based on bill No

url: `/lpcp/v1/fxsp/sys/bill/detail`

method: `POST`

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

consumes: `*/*`

Note:

Query billing details based on bill No

Example:

```
{
  "billNo": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inBillingDetail	InBillingDetail	body	true	InBillingDetail	InBillingDetail
billNo	Bill no.		true	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutBillingDetail
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutBillingDetail	OutBillingDetail
billInfoId	Bill info ID	integer(int64)	
billNo	Bill no.	string	
billCycle	Bill cycle	integer(int64)	
bankId	Bank ID	string	
bankName	Bank name	string	
formPlatform	From platform	string	
formCurrency	From currency	string	
toPlatform	To platform	string	
toCurrency	To currency	string	
billAmount	Bill amount	number	
createDate	Created on	integer(int64)	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "billInfoId": 0,
        "billNo": "",
        "billCycle": 0,
        "bankId": "",
        "bankName": "",
        "formPlatform": "",
        "formCurrency": "",
        "toPlatform": "",
        "toCurrency": "",
        "billAmount": 0,
        "createDate": 0
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Resource manage

Fetch Permission List

url: `/lpcp/v1/fxsp/resource/manage/all/searches`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to retrieve the permission list for role configurations.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoListSysResourceTreeExtended
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	array	SysResourceTreeExtended
rsucId		integer(int64)	
parentId		integer(int32)	
rsucCode		string	
rsucName		string	
rsucUrl		string	
priority		integer(int32)	
createDate		integer(int64)	
updateDate		integer(int64)	
lockable		integer(int32)	
rsucType		integer(int32)	
ischecked		integer(int32)	
bankPermissionType		integer(int32)	
systemType		integer(int32)	
childNodes	Child node	array	SysResourceTreeExtended

Response Example:

```
{
    "code": "",
    "message": "",
    "data": [
        {
            "rsucId": 0,
            "parentId": 0,
            "rsucCode": "",
            "rsucName": "",
            "rsucUrl": "",
            "priority": 0,
            "createDate": 0,
            "updateDate": 0,
            "lockable": 0,
            "rsucType": 0,
            "ischecked": 0,
            "bankPermissionType": 0,
            "systemType": 0,
            "childNodes": [
                {
                    "rsucId": 0,
                    "parentId": 0,
                    "rsucCode": "",
                    "rsucName": "",
                    "rsucUrl": "",
                    "priority": 0,
                    "createDate": 0,
                    "updateDate": 0,
                    "lockable": 0,
                    "rsucType": 0,
                    "ischecked": 0,
                    "bankPermissionType": 0,
                    "systemType": 0,
                    "childNodes": [
                        {}
                    ]
                }
            ]
        }
    ]
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Common manage

Fetch System UI Configurations

url: `/lpcp/v1/fxsp/anon/common/system/style/search`

method: `POST`

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

consumes: `*/*`

Note:

This interface is called to retrieve the configuration settings for the System's user interface, including the logo, System name and the color theme.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutSystemStyleSearch
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutSystemStyleSearch	OutSystemStyleSearch
systemName	System name	string	
logoFileHash	Logo file hash	string	
themeColor	Theme color	string	

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "systemName": "",
        "logoFileHash": "",
        "themeColor": ""
    }
}
```

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Fetch Dropdown Content by TypeCode

url: `/lpcp/v1/fxsp/anon/common/drop/down/box/searches`

method: `POST`

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

consumes: `*/*`

Note:

Param detailed description

- 1. dropDownTypeCode value is drop_down_country_info can query information for all countries
- 2. dropDownTypeCode value is drop_down_platform_currency_info can query all platform and currency information,csePCode can be null,If the value of csePCode is FXSP_PLATFORM_CURRENCY_VAILD only queries valid platform currencies
- 3. dropDownTypeCode value is drop_down_exchange_rate_info can query all exchange rate information,csePCode can be null,If the value of csePCode is FXSP_EXCHANGE_RATE_VAILD only queries valid exchange rate
- 4. dropDownTypeCode value is drop_down_business_status_info can query business status,If the value of csePCode is FXSP_FOREIGN_EXCHANGE_APPLICATION the query is for the status

of the exchange rate application form,if the value of csePCode is FXSP_CHARGING_MODEL Query the settlement type of the settlement model

- 5. dropDownTypeCode value is drop_down_business_status_info The value of csePCode must be filled in

Example:

```
{
  "dropDownTypeCode": "",
  "csePCode": ""
}
```

Params:

Name	Description	In	Required	Type	Schema
inStatusSearches	InStatusSearches	body	true	InStatusSearches	InStatusSearches
dropDownTypeCode	**drop-down typeCode** • Country: typeCode= drop_down_country_info • PlatformCurrency: typeCode= drop_down_platform_currency_info • ExchangeRatePair: typeCode= drop_down_exchange_rate_info • BusinessStatus: typeCode= drop_down_business_status_info		true	string	
csePCode	**Page search criteria - drop-down list data query** • PlatformCurrency-status: csePCode= FXSP_PLATFORM_CURRENCY_VAILD • ExchangeRatePair-status: csePCode= FXSP_EXCHANGE_RATE_VAILD • Profile-Foreign Exchange Application-status: csePCode= FXSP_FOREIGN_EXCHANGE_APPLICATION • Settlement Management-Charging Model-status: csePCode= FXSP_CHARGING_MODEL		false	string	

Status:

Code	Description	Schema
200	OK	ResultInfoOutDropDownInfo
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutDropDownInfo	OutDropDownInfo
dataInfo	Drop-down value	object	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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

Login APIs

Generate CAPTCHA image

url: `/lpcp/v1/fxsp/anon/generate/captcha`

method: `GET`

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

consumes: `application/json,*/*`

Note:

This interface is called to generate CAPTCHA image.

Params:

None

Status:

Code	Description	Schema
200	OK	ResultInfoOutGenerateCaptcha

Code	Description	Schema
500	Internal Server Error	ResultInfo

code-200

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data		OutGenerateCaptcha	OutGenerateCaptcha
baseStr		string	

Response Example:

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

code-500

Response Params:

Name	Description	Type	Schema
code	Code	string	
message	Message	string	
data	Api return data	object	

Response Example:

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