Project Kissen APIs

v1.0.0

Commercial Bank APIs

Bank wallet detail record

Operation record list

url: `/v1/wallet/detail/operation/listPage`

method: `POST`

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

consumes: `*/*`

Note:

Operation record list

Example:

```
{
  "data": {
    "bankAccountId": ""
  },
  "page": {
    "pageNum": 1,
    "pageSize": 10
  }
}
```

Params:

name	description	in	require	type	schema
dataTableOfWalletOperationPageReqVO	DataTableOfWalletOperationPageReqVO	body	true	DataTableOfWalletOperationPageReqVO	DataTableOfWalletOperationPageReqVO
data	Query Entity		true	WalletOperationPageReqVO	WalletOperationPageReqVO
bankAccountId	Bank wallet id		true	string	
page	Paging Information		true	PageInfo	PageInfo
pageNum	pageNum		true	integer	
pageSize	pageSize		true	integer	

Status:

code	description	schema
200	OK	ResultInfoOfResultDataOfWalletOperationPageRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)

name	description	type	schema
data	data	ResultDataOfWalletOperationPageRespVO	ResultDataOfWalletOperationPageRespVO
page	page	ResultPageInfo	ResultPageInfo
isFirstPage	isFirstPage	boolean	
isLastPage	isLastPage	boolean	
pageNum	pageNum	integer	
pageSize	pageSize	integer	
pages	pages	integer	
total	total	integer	
rows	return information	array	WalletOperationPageRespVO
bankAccountId	Bank wallet id	string	
state	Status 3: Pending (pending) 4: Commit in progress (in process) 5: Commit successful (valid) 6: Commit failed	integer	
txHash	Transaction hash	string	
txTime	Transaction time	integer	
type	Type. 0: Create 1: Freeze 2: Unfreeze	integer	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
            "pageNum": 0,
            "pageSize": 0,
            "pages": 0,
            "total": 0
        },
        "rows": [
            {
                "bankAccountId": "",
                "state": 0,
                "txHash": "",
                "txTime": 0,
                "type": 0
            }
        ]
    },
    "message": ""
}
```

Bank wallet management

Wallet address list

url: `/v1/wallet/address/list`

method: `POST`

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

consumes: `*/*`

Note:

Wallet address list

Example:

```
{
  "centralBankId": 0
}
```

Params:

name	description	in	require	type	schema
walletAddressListReqVO	WalletAddressListReqVO	body	true	WalletAddressListReqVO	WalletAddressListReqVO
centralBankId	Central bank id		false	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfListOfstring

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	
message	message	string	

Response Example:

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

Wallet top up-withdraw tx apply

url: `/v1/wallet/applyCbdcTx`

method: `POST`

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

consumes: `*/*`

Note:

Wallet top up/withdraw tx apply

Example:

```
{
  "amount": 0,
  "currency": "",
  "password": "",
  "txType": 0,
  "walletAddress": ""
}
```

Params:

name	description	in	require	type	schema
bankWalletTopUpReqVO	BankWalletTopUpReqVO	body	true	BankWalletTopUpReqVO	BankWalletTopUpReqVO
amount	Amount		true	number(bigdecimal)	
currency	Currency symbol		true	string	
password	Password		true	string	
txType	TxType 1: Top Up 2: Withdraw		true	integer(int32)	
walletAddress	Wallet address		true	string	

Status:

code	description	schema
200	OK	ResultInfo

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	object	
message	message	string	

Response Example:

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

Capital pool top up-withdraw tx apply

url: `/v1/wallet/capitalPool/applyCbdcTx`

method: `POST`

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

consumes: `*/*`

Note:

Wallet top up/withdraw tx apply

Example:

```
{
  "amount": 0,
  "currency": "",
  "password": "",
  "txType": 0,
  "walletAddress": ""
}
```

Params:

name	description	in	require	type	schema
bankWalletTopUpReqVO	BankWalletTopUpReqVO	body	true	BankWalletTopUpReqVO	BankWalletTopUpReqVO
amount	Amount		true	number(bigdecimal)	
currency	Currency symbol		true	string	
password	Password		true	string	
txType	TxType 1: Top Up 2: Withdraw		true	integer(int32)	
walletAddress	Wallet address		true	string	

Status:

code	description	schema
200	OK	ResultInfo

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)

name	description	type	schema
data	data	object	
message	message	string	

Response Example:

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

Central bank list

url: `/v1/wallet/centralBank/list`

method: `POST`

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

consumes: `*/*`

Note:

Central bank list

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfCentralBankRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	CentralBankRespVO
bankIdentifierCode	Bank identifier code	string	
centralBankId	Central bank id	integer(int64)	
centralBankName	Central bank name	string	
digitalSymbol	Digital symbol	string	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": [
        {
            "bankIdentifierCode": "",
            "centralBankId": 0,
            "centralBankName": "",
            "digitalSymbol": ""
        }
    ],
    "message": ""
}
```

List of opened central banks

url: `/v1/wallet/centralBank/opened/list`

method: `POST`

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

consumes: `*/*`

Note:

List of opened central banks

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfCentralBankRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	CentralBankRespVO
bankIdentifierCode	Bank identifier code	string	
centralBankId	Central bank id	integer(int64)	
centralBankName	Central bank name	string	
digitalSymbol	Digital symbol	string	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": [
        {
            "bankIdentifierCode": "",
            "centralBankId": 0,
            "centralBankName": "",
            "digitalSymbol": ""
        }
    ],
    "message": ""
}
```

Query basic wallet information

url: `/v1/wallet/detail/basic`

method: `POST`

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

consumes: `*/*`

Note:

Query basic wallet information

Example:

```json
{
  "bankAccountId": 0
}
```

Params:

name	description	in	require	type	schema
bankWalletDetailReqVO	BankWalletDetailReqVO	body	true	BankWalletDetailReqVO	BankWalletDetailReqVO
bankAccountId	Bank account id		true	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfBankWalletDetailBasicRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	BankWalletDetailBasicRespVO	BankWalletDetailBasicRespVO
bankAccountId	Central bank id	integer(int64)	

name	description	type	schema
bnCode	Bn code	string	
centralBankName	Central bank name	string	
chainAccountAddress	Wallet address	string	
createTime	Create time	integer(int64)	
createUser	Create user	string	
currency	Currency	string	
reviewerRemarks	Reviewer remarks	string	
reviewerTime	Reviewer time	integer(int64)	
reviewerUser	Reviewer user	string	
state	State 1：pending approval 2：rejected 5：active 3/4/7：processing 6:processing failed 8：inactive	integer(int32)	
txHash	Transaction hash	string	
txTime	Transaction time	integer(int64)	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "bankAccountId": 0,
        "bnCode": "",
        "centralBankName": "",
        "chainAccountAddress": "",
        "createTime": 0,
        "createUser": "",
        "currency": "",
        "reviewerRemarks": "",
        "reviewerTime": 0,
        "reviewerUser": "",
        "state": 0,
        "txHash": "",
        "txTime": 0
    },
    "message": ""
}
```

Query wallet transaction list page

url: `/v1/wallet/detail/transaction/listPage`

method: `POST`

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

consumes: `*/*`

Note:

Query wallet transaction list page

Example:

```
{
  "data": {
    "bankAccountId": 0
  },
  "page": {
    "pageNum": 1,
    "pageSize": 10
  }
}
```

Params:

name	description	in	require	type	schema
dataTableOfBankWalletDetailReqVO	DataTableOfBankWalletDetailReqVO	body	true	DataTableOfBankWalletDetailReqVO	DataTableOfBankWalletDetailReqVO
data	Query Entity		true	BankWalletDetailReqVO	BankWalletDetailReqVO
bankAccountId	Bank account id		true	integer	
page	Paging Information		true	PageInfo	PageInfo
pageNum	pageNum		true	integer	
pageSize	pageSize		true	integer	

Status:

code	description	schema
200	OK	ResultInfoOfResultDataOfWalletDetailTransactionPageRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	ResultDataOfWalletDetailTransactionPageRespVO	ResultDataOfWalletDetailTransactionPageRespVO
page	page	ResultPageInfo	ResultPageInfo
isFirstPage	isFirstPage	boolean	
isLastPage	isLastPage	boolean	
pageNum	pageNum	integer	
pageSize	pageSize	integer	
pages	pages	integer	
total	total	integer	

name	description	type	schema
rows	return information	array	WalletDetailTransactionPageRespVO
cbdcCount	Cbdc Count	number	
fromAccountAddress	From Address	string	
state	Transaction state, 1:pending 2/4 processing 5:success 3/6/7 failed Top-up/Withdrawal state, 1: pending approval 2: rejected 5: active 3/4/7: processing 6:processing failed	integer	
toAccountAddress	To Address	string	
transactionNo	Transaction no	string	
txHash	Transaction hash	string	
txTime	Transaction Time	integer	
type	1: swap 2: exchange 3: transfer 4:Top-up 5:Withdrawal	integer	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
            "pageNum": 0,
            "pageSize": 0,
            "pages": 0,
            "total": 0
        },
        "rows": [
            {
                "cbdcCount": 0,
                "fromAccountAddress": "",
                "state": 0,
                "toAccountAddress": "",
                "transactionNo": "",
                "txHash": "",
                "txTime": 0,
                "type": 0
            }
        ]
    },
    "message": ""
}
```

Query wallet transaction summary

url: `/v1/wallet/detail/transaction/summary`

method: `POST`

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

consumes: `*/*`

Note:

Query wallet transaction summary

Example:

```
{
  "bankAccountId": 0
}
```

Params:

name	description	in	require	type	schema
bankWalletDetailReqVO	BankWalletDetailReqVO	body	true	BankWalletDetailReqVO	BankWalletDetailReqVO
bankAccountId	Bank account id		true	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfWalletDetailTransactionSummaryRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	WalletDetailTransactionSummaryRespVO	WalletDetailTransactionSummaryRespVO
inflowSum	Inflow total	number(bigdecimal)	
outflowSum		number(bigdecimal)	
topUpSum	Top up total	number(bigdecimal)	
withdrawSum	Withdraw total	number(bigdecimal)	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "inflowSum": 0,
        "outflowSum": 0,
        "topUpSum": 0,
        "withdrawSum": 0
    },
    "message": ""
}
```

Query the list of wallets by currency

url: `/v1/wallet/listByCurrency`

method: `POST`

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

consumes: `*/*`

Note:

Query the list of wallets by currency

Example:

```
{
    "currency": ""
}
```

Params:

name	description	in	require	type	schema
bankCurrencyWalletReqVO	BankCurrencyWalletReqVO	body	true	BankCurrencyWalletReqVO	BankCurrencyWalletReqVO
currency	Currency		true	string	

Status:

code	description	schema
200	OK	ResultInfoOfListOfBankCurrencyWalletRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	BankCurrencyWalletRespVO
centralBankName	Central Bank name	string	

name	description	type	schema
chainAccountAddress	Chain Account Address	string	
currency	Currency	string	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": [
        {
            "centralBankName": "",
            "chainAccountAddress": "",
            "currency": ""
        }
    ],
    "message": ""
}
```

Wallet list page

url: `/v1/wallet/listPage`

method: `POST`

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

consumes: `*/*`

Note:

Wallet list page

Example:

```json
{
  "data": {
    "centralBankId": 0,
    "chainAccountAddress": "",
    "createTimeBegin": 0,
    "createTimeEnd": 0,
    "currency": "",
    "region": 0,
    "state": 0
  },
  "page": {
    "pageNum": 1,
    "pageSize": 10
  }
}
```

Params:

name	description	in	require	type	schema
dataTableOfBankWalletSearchesReqVO	DataTableOfBankWalletSearchesReqVO	body	true	DataTableOfBankWalletSearchesReqVO	DataTableOfBankWalletSearchesReqVO
data	Query Entity		true	BankWalletSearchesReqVO	BankWalletSearchesReqVO
centralBankId	Central bank id		false	integer	
chainAccountAddress	Wallet address		false	string	
createTimeBegin	Create time begin		false	integer	
createTimeEnd	Create time end		false	integer	
currency	Currency		false	string	
region	Region 1：domestic 2：foreign		false	integer	
state	State 1：pending approval 2：rejected 5：active 6:processing failed 7：processing 8：inactive		false	integer	
page	Paging Information		true	PageInfo	PageInfo
pageNum	pageNum		true	integer	
pageSize	pageSize		true	integer	

Status:

code	description	schema
200	OK	ResultInfoOfResultDataOfBankWalletSearchesRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	ResultDataOfBankWalletSearchesRespVO	ResultDataOfBankWalletSearchesRespVO
page	page	ResultPageInfo	ResultPageInfo
isFirstPage	isFirstPage	boolean	
isLastPage	isLastPage	boolean	
pageNum	pageNum	integer	
pageSize	pageSize	integer	
pages	pages	integer	
total	total	integer	
rows	return information	array	BankWalletSearchesRespVO
balance	Balance	number	
bankAccountId	Central bank id	integer	
centralBankName	Central bank name	string	
chainAccountAddress	Wallet address	string	
createTime	Create time	integer	
currency	Currency	string	
region	Region 1：home country 2：Other countries	integer	
state	State 1：pending approval 2：rejected 5：active 3/4/7：processing 6:processing failed 8：inactive	integer	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
            "pageNum": 0,
            "pageSize": 0,
            "pages": 0,
            "total": 0
        },
        "rows": [
            {
                "balance": 0,
                "bankAccountId": 0,
                "centralBankName": "",
                "chainAccountAddress": "",
                "createTime": 0,
                "currency": "",
                "region": 0,
                "state": 0
            }
        ]
    },
    "message": ""
}
```

Wallet save

url: `/v1/wallet/save`

method: `POST`

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

consumes: `*/*`

Note:

Wallet save

Params:

name	description	in	require	type	schema
bankWalletAddReqVO	bankWalletAddReqVO	query	true		
file	file	query	false	file	

Status:

code	description	schema
200	OK	ResultInfo

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	object	
message	message	string	

Response Example:

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

CBDC Transaction

Transaction details

url: `/v1/CBDCTransaction/details`

method: `POST`

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

consumes: `*/*`

Note:

details

Example:

```
{
  "transferId": 0
}
```

Params:

name	description	in	require	type	schema
cBDCTransactionDetailsReqVO	CBDCTransactionDetailsReqVO	body	true	CBDCTransactionDetailsReqVO	CBDCTransactionDetailsReqVO
transferId	transfer_id		true	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfCBDCTransactionDetailsRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	CBDCTransactionDetailsRespVO	CBDCTransactionDetailsRespVO
amount	Amount	string	
createUser	Create User	string	
creationTime	Creation Time	string	
currency	Currency	string	
remarks	remarks	string	
reviewerRemarks	Reviewer Remarks	string	
reviewerTime	Reviewer Time	string	
reviewerUser	Reviewer User	string	
status	Status 1:Pending Approval 2:Rejected 3:Processing 5:Success 6:Failed	integer(int32)	
transactionNo	Transaction No	string	
transferId	transfer_id	integer(int64)	
txHash	Transaction Hash	string	
txTime	Transaction Time	string	
type	Type 1:Top-up 2:Withdrawal	integer(int32)	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "amount": "",
        "createUser": "",
        "creationTime": "",
        "currency": "",
        "remarks": "",
        "reviewerRemarks": "",
        "reviewerTime": "",
        "reviewerUser": "",
        "status": 0,
        "transactionNo": "",
        "transferId": 0,
```

```
            "txHash": "",
            "txTime": "",
            "type": 0
        },
        "message": ""
    }
```

Transaction list

url: `/v1/CBDCTransaction/listPage`

method: `POST`

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

consumes: `*/*`

Note:

list

Example:

```
{
  "data": {
    "centralBankId": 0,
    "creationTimeEnd": "",
    "creationTimeStart": "",
    "from": "",
    "status": 0,
    "to": "",
    "transactionNo": "",
    "type": 0
  },
  "page": {
    "pageNum": 1,
    "pageSize": 10
  }
}
```

Params:

name	description	in	require	type	schema
dataTableOfCBDCTransactionListPageReqVO	DataTableOfCBDCTransactionListPageReqVO	body	true	DataTableOfCBDCTransactionListPageReqVO	DataTableOfCBDCTransactionListPageReqVO
data	Query Entity		true	CBDCTransactionListPageReqVO	CBDCTransactionListPageReqVO
centralBankId	Currency: /v1/common/centralBankList		false	integer	
creationTimeEnd	Creation Time End		false	string	
creationTimeStart	Creation Time Start		false	string	
from	From		false	string	
status	Status 1:Pending Approval 2:Rejected 3:Processing 6:Success 7:Failed		false	integer	
to	To		false	string	
transactionNo	Transaction No		false	string	
type	Type 1:Top-up 2:Withdrawal		false	integer	
page	Paging Information		true	PageInfo	PageInfo
pageNum	pageNum		true	integer	
pageSize	pageSize		true	integer	

Status:

code	description	schema
200	OK	ResultInfoOfResultDataOfCBDCTransactionListPageRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	ResultDataOfCBDCTransactionListPageRespVO	ResultDataOfCBDCTransactionListPageRespVO
page	page	ResultPageInfo	ResultPageInfo
isFirstPage	isFirstPage	boolean	
isLastPage	isLastPage	boolean	
pageNum	pageNum	integer	
pageSize	pageSize	integer	
pages	pages	integer	
total	total	integer	
rows	return information	array	CBDCTransactionListPageRespVO
amount	Amount	string	
creationTime	Creation Time	string	
currency	Currency	string	
from	From	string	
status	Status 1:Pending Approval 2:Rejected 3:Processing 5:Success 6:Failed	integer	
to	To	string	
transactionNo	Transaction No	string	
transferId	Transfer Id	integer	
type	Type	integer	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
            "pageNum": 0,
            "pageSize": 0,
            "pages": 0,
            "total": 0
        },
        "rows": [
```

```
        {
            "amount": "",
            "creationTime": "",
            "currency": "",
            "from": "",
            "status": 0,
            "to": "",
            "transactionNo": "",
            "transferId": 0,
            "type": 0
        }
    ]
},
"message": ""
}
```

Capital Pool authorize

Capital Pool authorize

url: `/v1/capitalpool/authorize`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "amount": 0,
  "chainAccountAddress": "",
  "currency": ""
}
```

Params:

name	description	in	require	type	schema
capitalPoolAuthorizeReqVO	CapitalPoolAuthorizeReqVO	body	true	CapitalPoolAuthorizeReqVO	CapitalPoolAuthorizeReqVO
amount	Amount		true	number(bigdecimal)	
chainAccountAddress	Wallet address		true	string	
currency	Currency		true	string	

Status:

code	description	schema
200	OK	ResultInfo

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	object	
message	message	string	

Response Example:

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

Commercial banks into the network

Commercial bank add and register

url: `/v1/commercial/bank/add`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "bankBic": "",
  "bnCode": "",
  "businessLicenseUrl": "",
  "contactName": "",
  "detailedAddress": "",
  "email": "",
  "fileName": "",
  "mobileNumber": "",
  "spBriefIntroduction": "",
  "spDescription": "",
  "spName": ""
}
```

Params:

name	description	in	require	type	schema
commercialBankAddReqVO	CommercialBankAddReqVO	body	true	CommercialBankAddReqVO	CommercialBankAddReqVO
bankBic	Bank bic		true	string	
bnCode	BN code		true	string	
businessLicenseUrl	Business license file hash		true	string	
contactName	Contact name		true	string	
detailedAddress	Detailed address		true	string	

name	description	in	require	type	schema
email	Email		true	string	
fileName	Business license file name		true	string	
mobileNumber	Mobile number		false	string	
spBriefIntroduction	Bank brief introduction		true	string	
spDescription	Bank description		true	string	
spName	Bank name		true	string	

Status:

code	description	schema
200	OK	ResultInfo

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	object	
message	message	string	

Response Example:

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

Commercial bank associated central bank

url: `/v1/commercial/bank/belongCentralBank`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfCentralBankRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	CentralBankRespVO	CentralBankRespVO
bankIdentifierCode	Bank identifier code	string	
centralBankId	Central bank id	integer(int64)	
centralBankName	Central bank name	string	
digitalSymbol	Digital symbol	string	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "bankIdentifierCode": "",
        "centralBankId": 0,
        "centralBankName": "",
        "digitalSymbol": ""
    },
    "message": ""
}
```

Query the bn code bound by the commercial bank

url: `/v1/commercial/bank/bnCodeSelect`

method: `POST`

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

consumes: `*/*`

Note:

Query the bn code bound by the commercial bank

Params:

Status:

code	description	schema
200	OK	ResultInfoOfstring

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	string	
message	message	string	

Response Example:

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

Commercial bank details

url: `/v1/commercial/bank/detail`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfCommercialBankInfoRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	CommercialBankInfoRespVO	CommercialBankInfoRespVO
bankBic	Bank bic	string	
bankType	Bank type:1=Commercial Bank 2=Settlement Bank	integer(int32)	
bnCode	BN code	string	
businessLicenseUrl	Business license file hash	string	
centralBankId	Central bank id	string	
centralBankName	Central bank name	string	
contactName	Contact name	string	
detailedAddress	Detailed address	string	

name	description	type	schema
email	Email	string	
fileName	Business license file name	string	
mobileNumber	Mobile number	string	
remark	Remark	string	
spBesuWalletAddress	Bank besu wallet address	string	
spBriefIntroduction	Bank brief introduction	string	
spCode	Bank code	string	
spDescription	Bank description	string	
spName	Bank name	string	
spStatus	SP status. 5=Pending 10=Approved 15=Rejected	integer(int32)	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "bankBic": "",
        "bankType": 0,
        "bnCode": "",
        "businessLicenseUrl": "",
        "centralBankId": "",
        "centralBankName": "",
        "contactName": "",
        "detailedAddress": "",
        "email": "",
        "fileName": "",
        "mobileNumber": "",
        "remark": "",
        "spBesuWalletAddress": "",
        "spBriefIntroduction": "",
        "spCode": "",
        "spDescription": "",
        "spName": "",
        "spStatus": 0
    },
    "message": ""
}
```

Commercial bank modification

url: `/v1/commercial/bank/edit`

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
commercialBankEditReqVO	CommercialBankEditReqVO	body	true	CommercialBankEditReqVO	CommercialBankEditReqVO
businessLicenseUrl	Business license file hash		true	string	
contactName	Contact name		true	string	
detailedAddress	Detailed address		true	string	
email	Email		true	string	
fileName	Business license file name		true	string	
mobileNumber	Mobile number		false	string	
spBriefIntroduction	Bank brief introduction		true	string	
spCode	Bank code		false	string	
spDescription	Bank description		true	string	

Status:

code	description	schema
200	OK	ResultInfo

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	object	
message	message	string	

Response Example:

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

Common Controller

List of all wallets (aggregated).

url: `/v1/common/allWalletList`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfAllWalletListRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	AllWalletListRespVO
chainAccountAddress	Chain Account Address	string	
walletExtendInfo	Chain Account Address	array	WalletExtendInfo
bankAccountId	Bank Account Id	integer	
bankIdentifierCode	Bank code	string	
bankName	Remitter's Bank Name	string	
cbdcCount	Available Balance	number	
centralBankId	Bank id	integer	
digitalCurrencyName	Digital Currency Name	string	
```

| name | description | type | schema |
|------|-------------|------|--------|
| digitalSymbol | Digital Symbol | string | |
| legalCurrencyName | Legal Currency Name | string | |
| message | message | string | |

**Response Example**:

```json
{
    "code": 0,
    "data": [
        {
            "chainAccountAddress": "",
            "walletExtendInfo": [
                {
                    "bankAccountId": 0,
                    "bankIdentifierCode": "",
                    "bankName": "",
                    "cbdcCount": 0,
                    "centralBankId": 0,
                    "digitalCurrencyName": "",
                    "digitalSymbol": "",
                    "legalCurrencyName": ""
                }
            ]
        }
    ],
    "message": ""
}
```

Bank information

url: `/v1/common/bankInfo`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfBankInfoRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	BankInfoRespVO	BankInfoRespVO
bankName	Bank Name	string	
bnCode	BN Code	string	
message	message	string	

Response Example:

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

List of central banks

url: `/v1/common/centralBankList`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfCentralBankListRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	CentralBankListRespVO
bankIdentifierCode	Bank Identifier Code	string	
bankName	Bank Name	string	
bankType	Bank Type	integer(int32)	

name	description	type	schema
centralBankId	Central Bank Id	integer(int64)	
contractBankCode	Contract Bank Code	string	
digitalCurrencyName	Currency Name	string	
digitalSymbol	Digital Symbol	string	
legalCurrencyName	Legal Currency Name	string	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "bankIdentifierCode": "",
            "bankName": "",
            "bankType": 0,
            "centralBankId": 0,
            "contractBankCode": "",
            "digitalCurrencyName": "",
            "digitalSymbol": "",
            "legalCurrencyName": ""
        }
    ],
    "message": ""
}
```

Exchange rate information

url: `/v1/common/fxRates`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "from": "",
  "to": ""
}
```

Params:

name	description	in	require	type	schema
fxRatesReqVO	FxRatesReqVO	body	true	FxRatesReqVO	FxRatesReqVO
from	from		true	string	
to	to		true	string	

Status:

code	description	schema
200	OK	ResultInfoOfListOfFxRatesRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	FxRatesRespVO
date	Date	number(biginteger)	
from	From	string	
provider	Provider	string	
rate	Rate	number(bigdecimal)	
rateId	Rate Id	string	
smChargeModel	SM ChargeModel 0:proportional 1: fixed	number(biginteger)	
smMaxFee	SM MaxFee	number(bigdecimal)	
smValue	SM Value	number(bigdecimal)	
to	To	string	
volume	Volume	number(biginteger)	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": [
        {
            "date": 0,
            "from": "",
            "provider": "",
            "rate": 0,
            "rateId": "",
            "smChargeModel": 0,
```

```
            "smMaxFee": 0,
            "smValue": 0,
            "to": "",
            "volume": 0
        }
    ],
    "message": ""
}
```

Information from the country's central bank

url: `/v1/common/nationalCentralBank`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfCentralBankListRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	CentralBankListRespVO	CentralBankListRespVO
bankIdentifierCode	Bank Identifier Code	string	
bankName	Bank Name	string	
bankType	Bank Type	integer(int32)	
centralBankId	Central Bank Id	integer(int64)	
contractBankCode	Contract Bank Code	string	
digitalCurrencyName	Currency Name	string	
digitalSymbol	Digital Symbol	string	
legalCurrencyName	Legal Currency Name	string	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": {
        "bankIdentifierCode": "",
        "bankName": "",
        "bankType": 0,
        "centralBankId": 0,
        "contractBankCode": "",
        "digitalCurrencyName": "",
        "digitalSymbol": "",
        "legalCurrencyName": ""
    },
    "message": ""
}
```

List of wallets in your home country

url: `/v1/common/nationalWalletList`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfNationalWalletListRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	NationalWalletListRespVO
accountName	Account Name	string	
bankAccountId	Bank Account Id	integer(int64)	
cbdcCount	Available Balance	number(bigdecimal)	
chainAccountAddress	Chain Account Address	string	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": [
        {
            "accountName": "",
            "bankAccountId": 0,
            "cbdcCount": 0,
            "chainAccountAddress": ""
        }
    ],
    "message": ""
}
```

Management cockpit(Dashboard)

List of valid currencies

url: `/v1/cockpit/getActiveCurrency`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfstring

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": [],
    "message": ""
}
```

CBDC Movements In the Last N Days

url: `/v1/cockpit/transaction/stat`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "currency": "",
  "endTime": 0,
  "startTime": 0
}
```

Params:

name	description	in	require	type	schema
cockpitTransactionReqVO	CockpitTransactionReqVO	body	true	CockpitTransactionReqVO	CockpitTransactionReqVO
currency	Currency		true	string	
endTime	End time		false	integer(int64)	
startTime	Start time		false	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfCockpitTransactionRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	CockpitTransactionRespVO	CockpitTransactionRespVO
daysList	days	array	integer
topUpAmountList	Top-up amount	array	number
transferInAmountList	Transfer in amount	array	number
transferOutAmountList	Transfer out amount	array	number
withdrawAmountList	Withdraw amount	array	number
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "daysList": [],
        "topUpAmountList": [],
        "transferInAmountList": [],
        "transferOutAmountList": [],
        "withdrawAmountList": []
    },
    "message": ""
}
```

CBDC Wallet Balance

url: `/v1/cockpit/walletInfo`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "chainAccountAddress": "",
  "currency": ""
}
```

Params:

name	description	in	require	type	schema
cockpitWalletReqVO	CockpitWalletReqVO	body	true	CockpitWalletReqVO	CockpitWalletReqVO
chainAccountAddress	Wallet address		false	string	
currency	Currency		false	string	

Status:

code	description	schema
200	OK	ResultInfoOfListOfCockpitWalletRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	CockpitWalletRespVO
balance	balance	number(bigdecimal)	

name	description	type	schema
chainAccountAddress	Wallet address	string	
currency	Currency	string	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "balance": 0,
            "chainAccountAddress": "",
            "currency": ""
        }
    ],
    "message": ""
}
```

Remittance Management

Transfer -> Beneficiary Information

url: `/v1/remittanceManagement/beneficiaryInformation`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "bankName": "",
  "chainAccountAddress": ""
}
```

Params:

name	description	in	require	type	schema
beneficiaryInformationReqVO	BeneficiaryInformationReqVO	body	true	BeneficiaryInformationReqVO	BeneficiaryInformationReqVO
bankName	Beneficiary's Name		false	string	
chainAccountAddress	Beneficiary's Wallet Address		false	string	

Status:

code	description	schema
200	OK	ResultInfoOfListOfBeneficiaryInformationRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	BeneficiaryInformationRespVO
bankId	Beneficiary's Id	integer(int64)	
bankName	Beneficiary's Name	string	
beneficiaryCurrencyRespVOs	Beneficiary's Currency List	array	BeneficiaryCurrencyRespVO
beneficiaryWalletExtendedRespVOs	Currency List	array	BeneficiaryWalletExtendedRespVO
bankWalletId	Beneficiary's Wallet Id	integer	
chainAccountAddress	Beneficiary's Wallet Address	string	
centralBankId	Beneficiary's Bank Id	integer	
centralBankName	Beneficiary's Bank Name	string	
digitalCurrencyName	Digital Currency Name	string	
digitalSymbol	Currency	string	
legalCurrencyName	Legal Currency Name	string	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "bankId": 0,
            "bankName": "",
            "beneficiaryCurrencyRespVOs": [
                {
                    "beneficiaryWalletExtendedRespVOs": [
                        {
                            "bankWalletId": 0,
                            "chainAccountAddress": ""
                        }
                    ],
                    "centralBankId": 0,
                    "centralBankName": "",
```

```
                    "digitalCurrencyName": "",
                    "digitalSymbol": "",
                    "legalCurrencyName": ""
                }
            ]
        }
    ],
    "message": ""
}
```

FX Purchasing

url: `/v1/remittanceManagement/fxPurchasing`

method: `POST`

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

consumes: `*/*`

Note:

FX Purchasing

Example:

```
{
  "fxPurchaseAmount": 0,
  "fxPurchasingInformation": "",
  "fxReceivingWalletId": 0,
  "passWord": "",
  "rateId": "",
  "transactionWalletId": 0
}
```

Params:

name	description	in	require	type	schema
fXPurchasingReqVO	FXPurchasingReqVO	body	true	FXPurchasingReqVO	FXPurchasingReqVO
fxPurchaseAmount	FX Purchase Amount		true	number(bigdecimal)	
fxPurchasingInformation	FX Purchasing Information		false	string	
fxReceivingWalletId	FX Receiving Wallet Address		true	integer(int64)	
passWord	Password		true	string	
rateId	Rate Id		true	string	
transactionWalletId	Transaction Wallet Address		true	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfboolean

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	boolean	
message	message	string	

Response Example:

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

FX Purchasing -> Receiving Information

url: `/v1/remittanceManagement/fxReceivingWalletList`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfFXReceivingWalletListRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	FXReceivingWalletListRespVO
bankId	Receiving Bank Id	integer(int64)	
bankName	Receiving Bank Name	string	
digitalCurrencyName	Currency Name	string	

name	description	type	schema
digitalSymbol	Digital Symbol	string	
legalCurrencyName	Legal Currency Name	string	
wallets	FX Receiving Wallet Address	array	FXReceivingWalletListRespVOExtendInfo
bankAccountId	Bank Account Id	integer	
chainAccountAddress	Chain Account Address	string	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "bankId": 0,
            "bankName": "",
            "digitalCurrencyName": "",
            "digitalSymbol": "",
            "legalCurrencyName": "",
            "wallets": [
                {
                    "bankAccountId": 0,
                    "chainAccountAddress": ""
                }
            ]
        }
    ],
    "message": ""
}
```

(Transfer -> Remitter Information) And (FX Purchasing -> Original Information)

url: `/v1/remittanceManagement/remitterInformation`

method: `POST`

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

consumes: `*/*`

Note:

Params:

Status:

code	description	schema
200	OK	ResultInfoOfListOfRemitterInformationRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	array	RemitterInformationRespVO
bankName	Central Bank Name	string	
centralBankId	Bank id	integer(int64)	
digitalCurrencyName	Digital Currency Name	string	
digitalSymbol	Digital Symbol	string	
legalCurrencyName	Legal Currency Name	string	
remitterInformationExtendInfoList	Wallet Address	array	RemitterInformationExtendInfo
bankAccountId	Bank Account Id	integer	
cbdcCount	Available Balance	number	
chainAccountAddress	Chain Account Address	string	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "bankName": "",
            "centralBankId": 0,
            "digitalCurrencyName": "",
            "digitalSymbol": "",
            "legalCurrencyName": "",
            "remitterInformationExtendInfoList": [
                {
                    "bankAccountId": 0,
                    "cbdcCount": 0,
                    "chainAccountAddress": ""
                }
            ]
        }
```

```
    ],
    "message": ""
}
```

Sign Test

url: `/v1/remittanceManagement/signTest`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "account": ""
}
```

Params:

name	description	in	require	type	schema
signTestReqVO	SignTestReqVO	body	true	SignTestReqVO	SignTestReqVO
account	Account		true	string	

Status:

code	description	schema
200	OK	ResultInfoOfSignTestRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	SignTestRespVO	SignTestRespVO
signData	sign data	string	
message	message	string	

Response Example:

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

Transaction details

url: `/v1/remittanceManagement/transaction/details`

method: `POST`

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

consumes: `*/*`

Note:

Transaction details

Example:

```
{
  "transferId": 0
}
```

Params:

name	description	in	require	type	schema
transactionDetailsReqVO	TransactionDetailsReqVO	body	true	TransactionDetailsReqVO	TransactionDetailsReqVO
transferId	Transfer Id		false	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfTransactionDetailsRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	TransactionDetailsRespVO	TransactionDetailsRespVO
cbdcCount	Cbdc Count	string	
commission	Commission	number(bigdecimal)	
createTime	Create Time	integer(int64)	
createUser	Create User	string	
exchangeRate	Exchange Rate	number(bigdecimal)	

name	description	type	schema
failureReason	Failure Reason	string	
fromAccountAddress	From Address	string	
fromBankName	From Central Bank	string	
fromCbdcCount	From Cbdc Count	string	
fromCommercialBankName	From Commercial Bank	string	
fromCurrency	From Currency	string	
remarks	Remarks	string	
serialNumber	Serial Number	string	
settlementBank	Settlement Bank	string	
state	State, 1: Pending authorization 2: Authorization is successful 3: Authorization failed 4: Transferring 5: The transfer was successful 6: The transfer failed 7: The transfer failed, and the authorization was cancelled successfully	integer(int32)	
toAccountAddress	To Address	string	
toBankName	To Central Bank	string	
toCbdcCount	To Cbdc Count	string	
toCommercialBankName	To Commercial Bank	string	
toCurrency	To Currency	string	
transactionLogTrees	Transaction Log Tree	array	TransactionLogTree
state	State 1:processing 2:success 3: fail -1: Not shown	integer	
txHash	Tx Hash	string	
txTime	Tx Time	integer	
transferId	Transfer Id	integer(int64)	
txHash	Tx Hash	string	
txTime	Tx Time	integer(int64)	

name	description	type	schema
type	Transfer Type, 1: Cross-border transfers 2: Cross-border foreign exchange purchases 3: Ordinary transfers	integer(int32)	
message	message	string	

Response Example:

```json
{
    "code": 0,
    "data": {
        "cbdcCount": "",
        "commission": 0,
        "createTime": 0,
        "createUser": "",
        "exchangeRate": 0,
        "failureReason": "",
        "fromAccountAddress": "",
        "fromBankName": "",
        "fromCbdcCount": "",
        "fromCommercialBankName": "",
        "fromCurrency": "",
        "remarks": "",
        "serialNumber": "",
        "settlementBank": "",
        "state": 0,
        "toAccountAddress": "",
        "toBankName": "",
        "toCbdcCount": "",
        "toCommercialBankName": "",
        "toCurrency": "",
        "transactionLogTrees": [
            {
                "state": 0,
                "txHash": "",
                "txTime": 0
            }
        ],
        "transferId": 0,
        "txHash": "",
        "txTime": 0,
        "type": 0
    },
    "message": ""
}
```

Transaction list

url: `/v1/remittanceManagement/transaction/listPage`

method: `POST`

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

consumes: `*/*`

Note:

Transaction list

Example:

```
{
  "data": {
    "centralBankId": 0,
    "endCreateTime": 0,
    "fromAccountAddress": "",
    "serialNumber": "",
    "startCreateTime": 0,
    "state": 0,
    "toAccountAddress": "",
    "type": 0
  },
  "page": {
    "pageNum": 1,
    "pageSize": 10
  }
}
```

Params:

name	description	in	require	type	schema
dataTableOfTransactionSearchesReqVO	DataTableOfTransactionSearchesReqVO	body	true	DataTableOfTransactionSearchesReqVO	DataTableOfTransactionSearchesReqVO
data	Query Entity		true	TransactionSearchesReqVO	TransactionSearchesReqVO
centralBankId	Central Bank Id		false	integer	
endCreateTime	End Create Time		false	integer	
fromAccountAddress	From Address		false	string	
serialNumber	Serial Number		false	string	
startCreateTime	Start Create Time		false	integer	
state	State, 1：Pending authorization 2：Authorization is successful 3：Authorization failed 4：Transferring 5：The transfer was successful 6：The transfer failed 7：The transfer failed, and the authorization was cancelled successfully		false	integer	
toAccountAddress	To Address		false	string	
type	Transfer Type, 1：Cross-border transfers 2：Cross-border foreign exchange purchases 3：Ordinary transfers		false	integer	
page	Paging Information		true	PageInfo	PageInfo
pageNum	pageNum		true	integer	
pageSize	pageSize		true	integer	

Status:

code	description	schema
200	OK	ResultInfoOfResultDataOfTransactionSearchesRespVO

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	ResultDataOfTransactionSearchesRespVO	ResultDataOfTransactionSearchesRespVO
page	page	ResultPageInfo	ResultPageInfo
isFirstPage	isFirstPage	boolean	
isLastPage	isLastPage	boolean	
pageNum	pageNum	integer	
pageSize	pageSize	integer	
pages	pages	integer	
total	total	integer	
rows	return information	array	TransactionSearchesRespVO
cbdcCount	Cbdc Count	string	
createTime		integer	
currency	Currency	string	
fromAccountAddress	From Address	string	
serialNumber	Serial Number	string	
state	State, 1: Pending authorization 2: Authorization is successful 3: Authorization failed 4: Transferring 5: The transfer was successful 6: The transfer failed 7: The transfer failed, and the authorization was cancelled successfully	integer	
toAccountAddress	To Address	string	
transferId	Transfer Id	integer	
type	Transfer Type, 1: Cross-border transfers 2: Cross-border foreign exchange purchases 3: Ordinary transfers	integer	
message	message	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "page": {
            "isFirstPage": false,
            "isLastPage": false,
```

```
                "pageNum": 0,
                "pageSize": 0,
                "pages": 0,
                "total": 0
            },
            "rows": [
                {
                    "cbdcCount": "",
                    "createTime": 0,
                    "currency": "",
                    "fromAccountAddress": "",
                    "serialNumber": "",
                    "state": 0,
                    "toAccountAddress": "",
                    "transferId": 0,
                    "type": 0
                }
            ]
        },
        "message": ""
    }
```

Transfer

url: `/v1/remittanceManagement/transfer`

method: `POST`

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

consumes: `*/*`

Note:

Transfer

Example:

```
{
    "beneficiaryBankId": 0,
    "beneficiaryWalletAddress": "",
    "interbankSettlementAmount": 0,
    "passWord": "",
    "rateId": "",
    "remittanceInformation": "",
    "remitterWalletId": 0,
    "toCommercialBankId": 0
}
```

Params:

name	description	in	require	type	schema
transferReqVO	TransferReqVO	body	true	TransferReqVO	TransferReqVO

name	description	in	require	type	schema
beneficiaryBankId	Beneficiary's Bank Name		true	integer(int64)	
beneficiaryWalletAddress	Beneficiary's Wallet Address		true	string	
interbankSettlementAmount	Interbank Settlement Amount		true	number(bigdecimal)	
passWord	Password		true	string	
rateId	Rate Id		false	string	
remittanceInformation	Remittance Information		true	string	
remitterWalletId	Remitter's Wallet Address		true	integer(int64)	
toCommercialBankId	Beneficiary's Id		true	integer(int64)	

Status:

code	description	schema
200	OK	ResultInfoOfboolean

Response Params:

name	description	type	schema
code	code	integer(int32)	integer(int32)
data	data	boolean	
message	message	string	

Response Example:

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

Login

Login

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

method: `POST`

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

consumes: `*/*`

Note:

Login

Params:

name	description	in	require	type	schema
code	User info	query	false	string	

Response Params:

None

Response Example:

Login out

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

method: `POST`

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

consumes: `*/*`

Note:

Login out

Headers:

name	description	in	require	type	schema
Cookie	Session id		true	string	

Params:

None

Response Params:

None

Response Example:

Service Provider APIs

User manage

User status update

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inUserStatusUpdate	InUserStatusUpdate	body	true	InUserStatusUpdate	InUserStatusUpdate
userId	user id		true	integer(int64)	
lockable	locked state: 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": {}
}
```

User searches

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inUserSearches	InUserSearches	body	true	InUserSearches	InUserSearches
pageNum	page number		true	integer(int32)	
pageSize	page size		true	integer(int32)	
```

name	description	in	require	type	schema
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
userId	user id	integer(int64)	
userName	user name	string	
realName	real name	string	
telephone	telephone	string	
email	email	string	
lockable	lockable: 1= Disable 2= Enable 3=Delete	integer(int32)	
superUser	superUser: 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": {}
}
```

Save user

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	lockable		true	integer(int32)	
roleIdList	role id list		true	array	integer(int64)
pwd	password		false	string	
superUser	superUser		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	

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

Response Example:

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

Password modification of the current user

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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": {}
}
```

Password reset of the current user(Password after reset:Abcd1234)

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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": {}
}
```

Administrator login for the first time

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

method: `POST`

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

consumes: `*/*`

Note:

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 login first	string	

Response Example:

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

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": {}
}
```

User edit

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	lockable		true	integer(int32)	

name	description	in	require	type	schema
roleIdList	role 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": {}
}
```

User edit detail search

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	lockable: 1= Disable 2= Enable 3=Delete	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 clearing bank

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	require	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": {}
}
```

Upgrade details

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

method: `POST`

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

consumes: `*/*`

Note:

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 10=Approved 15=Rejected	integer(int32)	
remark	remark	string	
fileList	file list	array	UpgradeFile
fileUrl	file url	string	
fileCode	file hash	string	
capitalPoolList	capital pool list	array	UpgradeCapitalPool
capitalPoolAddress	capital pool address	string	

name	description	type	schema
capitalPoolPlatform	capital pool platform	string	
capitalPoolCurrency	capital pool currency	string	

Response Example:

```json
{
    "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:

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

Transaction manage

Transaction list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inTransactionPage	InTransactionPage	body	true	InTransactionPage	InTransactionPage
pageNum	page number		true	integer(int32)	
pageSize	page size		true	integer(int32)	
transactionNo	transaction No		false	string	
transactionHash	transaction hash		false	string	
fromBnId	from Bn Id		false	string	
formPlatform	form platform		false	string	
formCurrency	form 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	

name	description	in	require	type	schema
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		integer(int64)	
pages		integer(int32)	
pageSize		integer(int32)	
pageNum		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": {}
}
```

Transaction details

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	form platform	string	
formCurrency	form currency	string	
fromAccount	from account	string	
fromBankId	from commercial bank id	string	
fromBankName	from commercial bank name	string	
fromBnId	from Bn Id	string	
fromTxnNo		string	
fromAmount	from Amount	number	
toPlatform	to platform	string	
toCurrency	to currency	string	
toAccount	to account	string	
toBankId		string	
toBankName		string	
toBnId		string	
toTxnNo		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)	

name	description	type	schema
regularCommission	regular commission. Charging Model equal 2 not empty	number	
ratioCommission	ratio commission. Charging Model equal 1 not empty	number	
maxCommission	max commission. Charging Model equal 1 not empty	number	
commissionPlatform		string	
commissionCurrency		string	
settlementModelName	settlement model name	string	
spCode		string	
exchangeRate	exchange rate	number	
status	status. 1 success. 0 fail	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": 100,
        "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 verify

Token verifies and validates menu permissions

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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": {}
}
```

Style manage

Submission system style

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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

code	description	schema
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": {}
}
```

SP Manage

Get sp wallet address(Besu)

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

method: `POST`

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

consumes: `*/*`

Note:

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	
message	Message	string	
data	Api return data	object	

Response Example:

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

Sp info modification

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

method: `POST`

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

consumes: `*/*`

Note:

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	
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

code	description	schema
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": {}
}
```

Sp details

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

method: `POST`

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

consumes: `*/*`

Note:

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	
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:

```json
{
    "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:

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

Sp add and register

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

```json
{
  "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	
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	
message	Message	string	
data	Api return data	object	

Response Example:

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

Settlement model

Settlement model list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	
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)	

name	description	type	schema
pages	pages	integer(int32)	
pageSize	page size	integer(int32)	
pageNum	page num	integer(int32)	
lastPage	last page	boolean	
firstPage	first page	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": {}
}
```

Single settlement model history list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	

name	description	type	schema
chargingModel	Charging Model:1=Proportional Charges 2=Fixed Charges	integer(int32)	
regularCommission	regular commission. Charging Model equal 2 not empty	number	
ratioCommission	ratio commission. Charging Model equal 1 not empty	number	
maxCommission	max commission. Charging Model equal 1 not empty	number	
createDate	create date	integer(int64)	
updateDate	update date	integer(int64)	
modifyUser	modify user	string	

Response Example:

```json
{
    "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	

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

Response Example:

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

Edit settlement model

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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)	

name	description	in	require	type	schema
regularCommission	regular commission. Charging Model equal 2 not empty		false	number	
ratioCommission	ratio commission. Charging Model equal 1 not empty		false	number	
maxCommission	max commission. Charging Model equal 1 not empty		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": {}
}
```

Settlement model details

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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

name	description	type	schema
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	regular commission. Charging Model equal 2 not empty	number	
ratioCommission	ratio commission. Charging Model equal 1 not empty	number	
maxCommission	max commission. Charging Model equal 1 not empty	number	
createDate	create date	integer(int64)	
updateDate	update date	integer(int64)	
modifyUser	modify user	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": {}
}
```

Add settlement model

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	

name	description	in	require	type	schema
settlementModelName	settlement model name		true	string	
chargingModel	Charging Model:1=Proportional Charges 2=Fixed Charges		true	integer(int32)	
regularCommission	regular commission. Charging Model equal 2 not empty		false	number	
ratioCommission	ratio commission. Charging Model equal 1 not empty		false	number	
maxCommission	max commission. Charging Model equal 1 not empty		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	

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

Role manage

Role status update

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inRoleStatusUpdate	InRoleStatusUpdate	body	true	InRoleStatusUpdate	InRoleStatusUpdate
roleCode	role code		true	string	
lockable	locked state		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	

name	description	type	schema
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 searches

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	lockable: 1= Disable 2= Enable 3=Delete	integer(int32)	
superRole	super role	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": {}
}
```

Save role

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inRoleSave	InRoleSave	body	true	InRoleSave	InRoleSave
roleName	role name		true	string	
description	role description		false	string	
lockable	locked state		true	integer(int32)	

name	description	in	require	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": {}
}
```

Role edit

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inRoleEdit	InRoleEdit	body	true	InRoleEdit	InRoleEdit
roleCode	role code		true	string	
roleName	role name		false	string	
description	role description		false	string	
lockable	locked state		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": {}
}
```

Role edit detail search

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	lockable	integer(int32)	
myResourceList	my resource list	array	SysResourceTreeExtended
rsucId	Resource id	integer	
parentId	parent id	integer	
rsucCode	Resource code	string	
rsucName	Resource name	string	
rsucUrl	Resource url	string	
priority	priority	integer	
createDate	create date	integer	
updateDate	update date	integer	
lockable	lockable	integer	
rsucType	Resource type	integer	
ischecked	ischecked	integer	
bankPermissionType	bank permission type	integer	
childNodes	Child node	array	SysResourceTreeExtended

Response Example:

```
{
    "code": "",
    "message": "",
```

```json
    "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,
                "childNodes": [
                    {
                        "rsucId": 0,
                        "parentId": 0,
                        "rsucCode": "",
                        "rsucName": "",
                        "rsucUrl": "",
                        "priority": 0,
                        "createDate": 0,
                        "updateDate": 0,
                        "lockable": 0,
                        "rsucType": 0,
                        "ischecked": 0,
                        "bankPermissionType": 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": {}
}
```

Catch resource to redis

url: `/fxsp/v1/fxsp/sys/role/manage/catch/resource/searches`

method: `POST`

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

consumes: `*/*`

Note:

Params:

None

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": {}
}
```

Role searches all

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

method: `POST`

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

consumes: `*/*`

Note:

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	

name	description	type	schema
description	description	string	
lockable	lockable: 1= Disable 2= Enable 3=Delete	integer(int32)	
superRole	super role	integer(int32)	

Response Example:

```json
{
    "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:

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

Password manage

Verify password

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inPasswordVerify	InPasswordVerify	body	true	InPasswordVerify	InPasswordVerify
encryptionStr	The encrypted string, the content before encryption is: {"pwd":"111111111"}		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": {}
}
```

Home manage

Transaction volume in the last 7 Days

url: `/fxsp/v1/fxsp/sys/home/transaction/volume/searches`

method: `POST`

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

consumes: `*/*`

Note:

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:

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

Check the exchange rate under review

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

method: `POST`

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

consumes: `*/*`

Note:

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
fromCurrency	From currency name	string	
fromPlatform	From platform name	string	
toCurrency	To currency name	string	
toPlatform	To platform name	string	
rateId	Rate id	string	

name	description	type	schema
count	Number	integer(int32)	

Response Example:

```json
{
    "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:

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

Query the currency under review

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

method: `POST`

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

consumes: `*/*`

Note:

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

method: `POST`

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

consumes: `*/*`

Note:

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		array	string

Response Example:

```
{
    "code": "",
    "message": "",
    "data": {
        "unopenedRateList": []
    }
}
```

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

method: `POST`

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

consumes: `*/*`

Note:

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	
data	Api return data	object	

Response Example:

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

Search latest exchange rate

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

method: `POST`

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

consumes: `*/*`

Note:

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	
targetCurrency	target currency name	string	
targetPlatform	target platform name	string	
exchangeRate	Exchange Rate	number	
date	Date	integer	
chainRateId	rateId	string	
settlementModelCode	settlementModelCode	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": {}
}
```

Exchange rate dynamics in the last 7 days

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inHistoryExchangeRateDynamicsInfo	InHistoryExchangeRateDynamicsInfo	body	true	InHistoryExchangeRateDynamicsInfo	InHistoryExchangeRateDynamicsInfo

name	description	in	require	type	schema
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	
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:

```json
{
    "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: `/fxsp/v1/fxsp/sys/home/approve/currency`

method: `POST`

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

consumes: `*/*`

Note:

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	

name	description	type	schema
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	create date	integer	
status	0=Pending 5=Approved 10=Rejected	integer	
count	Number	integer(int32)	

Response Example:

```json
{
    "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: `/fxsp/v1/fxsp/sys/history/rate/searches`

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inHistoricalRateSearches	InHistoricalRateSearches	body	true	InHistoricalRateSearches	InHistoricalRateSearches
pageNum	page number		true	integer(int32)	
pageSize	page size		true	integer(int32)	
formRatePlatform	form Rate Platform		false	string	
formRateCurrency	form 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

code	description	schema
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	formRatePlatform	string	
formRateCurrency	formRateCurrency	string	
toRatePlatform	toRatePlatform	string	
toRateCurrency	toRateCurrency	string	
exchangeRate	exchangeRate	number	
rateDate	rateDate	integer(int64)	

Response Example:

```json
{
    "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	

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

File manage

File upload

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

method: `POST`

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

consumes: `*/*`

Note:

Params:

name	description	in	require	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": {}
}
```

File extension download

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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:

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

File download

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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:

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

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": {}
}
```

Foreign exchange application

Search exchange application detail

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

method: `POST`

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

consumes: `*/*`

Note:

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)	

name	description	type	schema
status	Status	integer(int32)	
approvalResult	Approval Result	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": {}
}
```

Select foreign exchange application list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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		false	integer(int32)	
startApplicationDate	start application date		false	integer(int64)	
endApplicationDate	end application date		false	integer(int64)	
startApprovalDate	start approval date		false	integer(int64)	
endApprovalDate	end approval date		false	integer(int64)	
businessType	business type 0:rate 1:capitalPool		false	integer(int32)	
applicationType	business 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

name	description	type	schema
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	business type:0=Activate 1=Reduce	integer(int32)	

Response Example:

```json
{
    "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": {}
}
```

Search exchange application detail

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	Country of Region	string	
introduction	Brief Introduction	string	

name	description	type	schema
description	Description	string	
applicationTime	Application Time	integer(int64)	
status	Status	integer(int32)	
approvalResult	Approval Result	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": {}
}
```

Save exchange application

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	
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	Save Status:0=success 1=fail	integer(int32)	

Response Example:

```json
{
    "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:

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

Delete exchange pair

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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 opened services

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

method: `POST`

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

consumes: `*/*`

Note:

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	

name	description	type	schema
country	Country of Region	string	
introduction	Brief Introduction	string	
description	Description	string	
spBesuWalletAddress	sp besu 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	Form Platform	string	
formCurrency	Form Currency	string	
toPlatform	To Platform	string	
toCurrency	To Currency	string	
bicName	Bic	string	
centralBankName	Central Bank	string	

Response Example:

```json
{
    "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: `/fxsp/v1/fxsp/sys/exchange/approved/search`

method: `POST`

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

consumes: `*/*`

Note:

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	Save Status:0=success 1=fail	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 manage

Select capital pool

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

method: `POST`

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

consumes: `*/*`

Note:

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	capitalPool authorized balance	string	
capitalPoolRealBalance	capitalPool real balance	string	
createDate	create date	integer(int64)	
currencyPrecision	currency precision	integer(int32)	

Response Example:

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

code-500:

Response Params:

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

Response Example:

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

Searches the funds pool that can be deleted

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

method: `POST`

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

consumes: `*/*`

Note:

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	
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": {}
}
```

Select capital pool application list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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		false	integer(int32)	
startApplicationDate	start application date		false	integer(int64)	
endApplicationDate	end application date		false	integer(int64)	
startApprovalDate	start approval date		false	integer(int64)	
endApprovalDate	end approval 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
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	

name	description	type	schema
data	Api return data	object	

Response Example:

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

Search capital pool Application Detail

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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

name	description	type	schema
spName	SP Name	string	
country	Country of Region	string	
introduction	Brief Introduction	string	
description	Description	string	
applicationTime	Application Time	integer(int64)	
approvalTime	approval 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)	
businessApplicationCode	Applicaiton No	string	
outPlatformCurrencyInformations	Platform Currency Information	array	OutPlatformCurrencyInformation
platformCurrency	Platform Currency	string	
capitalPoolAddress	capitalPoolAddress	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,
        "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 Application

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inDeleteCapitalPool	InDeleteCapitalPool	body	true	InDeleteCapitalPool	InDeleteCapitalPool

name	description	in	require	type	schema
settlementInformations	Settlement Information		false	array	InSettlementInformation
capitalPoolPlatform	Capital Pool Platform		false	string	
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	Save Status:0=success 1=fail	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": {}
}
```

Billing

Billing transaction detail list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	last page	boolean	
firstPage	first page	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": {}
}
```

Billing list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	
formPlatform	From Platform		false	string	

name	description	in	require	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)	

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	last page	boolean	
firstPage	first page	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": {}
}
```

Billing invoice select

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	create date	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": {}
}
```

Billing invoice export

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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": {}
}
```

Billing transaction export

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	type	schema
inBillingExport	InBillingExport	body	true	InBillingExport	InBillingExport
minId	min id		false	integer(int64)	
maxId	max id		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": {}
}
```

Billing details

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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	create date	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

Resource tree searches

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

method: `POST`

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

consumes: `*/*`

Note:

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	

name	description	type	schema
message	Message	string	
data	Api return data	array	SysResourceTreeExtended
rsucId	rsuc id	integer(int64)	
parentId	parent id	integer(int32)	
rsucCode	rsuc code	string	
rsucName	rsuc name	string	
rsucUrl	rsuc url	string	
priority	priority	integer(int32)	
createDate	create date	integer(int64)	
updateDate	update date	integer(int64)	
lockable	lockable	integer(int32)	
rsucType	rsuc type	integer(int32)	
ischecked	ischecked	integer(int32)	
bankPermissionType	bank permission type	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,
            "childNodes": [
                {
                    "rsucId": 0,
                    "parentId": 0,
                    "rsucCode": "",
```

```
                    "rsucName": "",
                    "rsucUrl": "",
                    "priority": 0,
                    "createDate": 0,
                    "updateDate": 0,
                    "lockable": 0,
                    "rsucType": 0,
                    "ischecked": 0,
                    "bankPermissionType": 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

Querying system configuration

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

method: `POST`

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

consumes: `*/*`

Note:

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 subject 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": {}
}
```

Search drop-down list

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

method: `POST`

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

consumes: `*/*`

Note:

Example:

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

Params:

name	description	in	require	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: `/fxsp/v1/fxsp/anon/generate/captcha`

method: `GET`

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

consumes: `*/*`

Note:

Params:

None

Status:

code	description	schema
200	OK	
500	Internal Server Error	ResultInfo

Response Params:

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

Response Example:

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