#POC6 API

Description:POC6 interface document

Version:1.0.1

POC6:Meta transaction manage APIs

01.Search all possible Transaction Node info which could handle the transaction (pre-process query).

url:`/v1/udpn/processing/metatx/manage/searchs`

method:`POST`

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

consumes:`*/*`

Note:

Example:

```
{
  "beneficiaryName": "beneficiaryName",
  "currencyType": "USDT",
  "platformType": "ETH",
  "sourceAccountAddress": "0x78AB5",
  "targetAccountAddress": "0x98AB56",
  "transferAmount": 1
}
```

Params:

name	description	in	require	type	schema
inMetaTxManageSearchs	InMetaTxManageSearchs	body	true	InMetaTxManageSearchs	InMetaTxManageSearchs
beneficiaryName	Beneficiary name. Remark:max length is 100		false	string	
currencyType	Currency type. Remark:max length is 20		true	string	
platformType	Platform type. Remark:max length is 20		true	string	
sourceAccountAddress	Source account address. Remark:max length is 100		true	string	
targetAccountAddress	Target account address. Remark:max length is 100		true	string	
transferAmount	Transfer amount. Remark:integer max length is 32;fraction max length is 18		true	number(bigdecimal)	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«List«OutMetaTxManageSearchs»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	array	OutMetaTxManageSearchs
beneficiaryName	Beneficiary name	string	
currencyPrecision	Currency precision	integer(int32)	
currencyType	Currency type	string	
isKyc	Does it support KYC. Remark:1=True;2=False;999=Unknown	integer(int32)	
key	Meta transaction key	string	
nodeDescription	Transaction Node description	string	
originalTnCode	Original Transaction Node code	string	
originalTnName	Original Transaction Node name	string	
platformType	Platform type	string	
sourceAccountAddress	Source account address	string	
spliceCurrencyType	Splice currency type and platform type	string	
targetAccountAddress	Target account address	string	
transferAmount	Transfer amount	number(bigdecimal)	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "beneficiaryName": "beneficiaryName",
            "currencyPrecision": 6,
            "currencyType": "USDC",
            "isKyc": 1,
            "key":
"UDPN0001_UDPN000100013_3526a13061ae4ec0ad1fab8f92a34a851636104061598",
            "nodeDescription": "Test node description1",
            "originalTnCode": "UDPN0001",
            "originalTnName": "Test node name1",
            "platformType": "ETH",
            "sourceAccountAddress":
"0x78AB5678CDF8907654909089765432345906AC0D",
            "spliceCurrencyType": "USDC-ETH",
            "targetAccountAddress":
"0x98AB5678CDF8907654909089765432345906AD0C",
            "transferAmount": 100
        }
    ],
    "message": "ok"
}
```

02.Get the data that needs to be signed.

url: `/v1/udpn/processing/metatx/manage/permit/info/search`

method: `POST`

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

consumes: `*/*`

Note:

The user needs to use the private key of sourceAccountAddress to sign the permitWaitSignatureInfo in the returned data.The signature result is used to send the transaction. ETH uses ECDSA
This interface will occupy a nonce. After calling this interface, you need to call interface 03 immediately.

Example:

```
{
  "key": "UDPN0001_UDPN000100013_3526a13061ae4ec0ad1fab8f92a34a851636104061598",
  "originalTnCode": "VN000100331",
  "sourceAccountAddress": "0x78AB5678CDF8907654909089765432345906AAAC",
  "transAmount": 100
}
```

Params:

name	description	in	require	type	schema
inMetaTxManagePermitSignatureInfoSelect	InMetaTxManagePermitSignatureInfoSelect	body	true	InMetaTxManagePermitSignatureInfoSelect	InMetaTxManagePermitSignatureInfoSelect
key	Meta transaction key. Remark:max length is 100;Returned by the 'pre-process query' interface		true	string	
originalTnCode	Original Transaction Node code. Remark:max length is 50		true	string	
sourceAccountAddress	Source account address. Remark:max length is 100		true	string	
transAmount	Transaction amount. Remark:integer max length is 32;fraction max length is 18		true	number(bigdecimal)	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«OutMetaTxManagePermitWaitSignatureInfoSelect»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	OutMetaTxManagePermitWaitSignatureInfoSelect	OutMetaTxManagePermitWaitSignatureInfoSelect
key	Meta transaction key. Remark:max length is 100;Returned by the 'pre-process query' interface	string	
permitWaitSignatureInfo	Data that requires the user to sign using their private key	string	
permitWaitSignatureSummaryInfo	Summary of data to be signed	string	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "key":
"UDPN0001_UDPN000100013_3526a13061ae4ec0ad1fab8f92a34a851636104061598",
        "permitWaitSignatureInfo": "{'test1':'a','test2':'b'}",
        "permitWaitSignatureSummaryInfo": "testhash"
    },
    "message": "ok"
}
```

03.Submit meta transaction.

url: `/v1/udpn/processing/metatx/manage/save`

method: `POST`

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

consumes: `*/*`

Note:

The parameter userSignatureData is the data used by the user to sign the results returned by the 02 interface.

Example:

```
{
  "beneficiaryName": "beneficiaryName",
  "currencyType": "USDC",
  "key": "UDPN0001_UDPN000100013_3526a13061ae4ec0ad1fab8f92a34a851636104061598",
  "originalTnCode": "UDPN000100003",
  "platformType": "ETH",
  "sourceAccountAddress": "0x78AB5678CDF8907654909089765432345906AC0D",
  "targetAccountAddress": "0x98AB5678CDF8907654909089765432345906AD0C",
  "transAmount": 100,
  "userSignatureData": {
    "r": "5124c8477159844f385258b38f9d39e4683c1967ae365e0d272c01e1529be4a2",
    "s": "1b4a467b8bf03142e4c6b0a880121c4b7cd316b61c36c1c5d05b0047c8492b53",
    "v": "1b"
  }
}
```

Params:

name	description	in	require	type	schema
inMetaTxManageSave	InMetaTxManageSave	body	true	InMetaTxManageSave	InMetaTxManageSave
beneficiaryName	Beneficiary name. Remark:max length is 100		false	string	
currencyType	Currency type. Remark:max length is 20		true	string	
key	Meta transaction key. Remark:max length is 100;Returned by the 'pre-process query' interface		true	string	
originalTnCode	Original Transaction Node code. Remark:max length is 50		true	string	
platformType	Platform type. Remark:max length is 20		true	string	
sourceAccountAddress	Source account address. Remark:max length is 100		true	string	
targetAccountAddress	Target account address. Remark:max length is 100		true	string	
transAmount	Transaction amount. Remark:integer max length is 32;fraction max length is 18		true	number(bigdecimal)	
userSignatureData	User signed data		false	InUserSignatureData	InUserSignatureData
r	ECDSA signature result: R		true	string	
s	ECDSA signature result: S		true	string	
v	ECDSA signature result: V		true	string	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«OutMetaTxManageSave»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	OutMetaTxManageSave	OutMetaTxManageSave
groupCode	The unique identifier of the transaction	string	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "groupCode": "Metad97fc1f58"
    },
    "message": "ok"
}
```

04.Get Business Node wallet address and estimated gas fee.

url: `/v1/udpn/processing/metatx/manage/bnAddressAndEstimateGasFee/select`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "tnCode": "TN00001"
}
```

Params:

name	description	in	require	type	schema
inGetBnAddressAndEstimateGasFee	InGetBnAddressAndEstimateGasFee	body	true	InGetBnAddressAndEstimateGasFee	InGetBnAddressAndEstimateGasFee
tnCode	Transaction Node code		true	string	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«OutGetBnAddressAndEstimateGasFee»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	OutGetBnAddressAndEstimateGasFee	OutGetBnAddressAndEstimateGasFee
bnWalletAddress	Business Node wallet address	string	
estimateGasFee	Estimate gas fee	string	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "bnWalletAddress": "0xFc8d04578B0fF6441ACfd2183525b0C783C2C4C1",
        "estimateGasFee": "0.02"
    },
    "message": "ok"
}
```

05.Get drop-down box data of meta-transaction status.

url: `/v1/udpn/processing/metatx/manage/drop/down/status/searchs`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "csePCode": "META_TRANSACTION_STATUS"
}
```

Params:

name	description	in	require	type	schema
inStatusSearches	InStatusSearches	body	true	InStatusSearches	InStatusSearches
csePCode	**Page search criteria - drop-down list data query**: • Meta transaction status： csePCode= META_TRANSACTION_STATUS		true	string	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«List«OutStatusSearches»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	array	OutStatusSearches
cseDesc	Drop-down description	string	
csePCode	Drop-down module encoding	string	
cseValue	Drop-down value	string	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "cseDesc": "Pending",
            "csePCode": "META_TRANSACTION_STATUS",
            "cseValue": "1"
        }
    ],
    "message": "ok"
}
```

06.Get meta transaction list.

url: `/v1/udpn/processing/metatx/manage/info/searchs`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "beneficiaryAccount": "",
  "currencyType": "USDC",
  "inPage": {
    "pageNum": 1,
    "pageSize": 10
  },
  "platformType": "ETH",
  "sendAccount": "",
  "status": "1",
  "submitEndDate": 0,
  "submitStartDate": 0,
  "transactionKey": ""
}
```

Params:

name	description	in	require	type	schema
inMetaTxManageAllTransactionSearches	InMetaTxManageAllTransactionSearches	body	true	InMetaTxManageAllTransactionSearches	InMetaTxManageAllTransactionSearches
beneficiaryAccount	Beneficiary account		false	string	
currencyType	Currency type		false	string	
inPage	Paging config:If InPage has no value, the first page is displayed by default, containing 10 pieces of data		true	InPage	InPage
pageNum	Current page: DefaultValue:1		true	integer	
pageSize	Page size: DefaultValue:10		true	integer	
platformType	Platform type		false	string	
sendAccount	Send account		false	string	

status	Meta transaction status:
	1=Pending;
	2=Permit processing;
	3=Permit processed successfully;
	4=Permit failed;
	5=Transfer processing;
	6=Transfer processed successfully;
	7=Transfer failed;
	8=BN init;
	9=Submitted vn;
			false	string	

submitEndDate	Submit end date		false	integer(int64)	
submitStartDate	Submit start date		false	integer(int64)	
transactionKey	Meat transaction key		false	string	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«List«OutMetaTxManageAllTransactionSearches»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	array	OutMetaTxManageAllTransactionSearches
amount	Meat transaction amount	number(bigdecimal)	
beneficiaryAccount	Beneficiary account	string	
currencyType	Splice currency type and platform type	string	
sendAccount	Send account	string	
status	Meta transaction status	string	
submitDate	Submit date	integer(int64)	
transactionKey	Meat transaction key	string	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": [
        {
            "amount": 1.235,
            "beneficiaryAccount": "0x7392749234092840982",
            "currencyType": "USDC-ETH",
            "sendAccount": "0x7392749234092840982",
            "status": "Pending",
            "submitDate": 1704693848,
            "transactionKey": "UDPN0001_UDPN000100013_a711"
        }
    ],
    "message": "ok"
}
```

07.Get meta transaction detail information.

url: `/v1/udpn/processing/metatx/manage/detail/info/search`

method: `POST`

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

consumes: `*/*`

Note:

Example:

```
{
  "transactionKey": "Metae29a2f631c644e0c8bb475309814a1c0"
}
```

Params:

name	description	in	require	type	schema
inMetaTransactionManageDetail	inMetaTransactionManageDetail	body	true	InMetaTransactionManageDetail	InMetaTransactionManageDetail
transactionKey	Meta transaction key		true	string	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«OutMetaTxManageDetail»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	OutMetaTxManageDetail	OutMetaTxManageDetail
amount	Meta transaction amount	number(bigdecimal)	
beneficiaryAccount	Beneficiary account	string	
beneficiaryName	Beneficiary name	string	
bnDeductGasFeeAccount	Business Node deduct gas fee address	string	
channelFeeCurrency	Channel fee currency type	string	
currencyType	Currency type	string	
failReason	Fail reason	string	
isKyc	Does it support KYC. Remark:1=True;2=False;999=Unknown	integer(int32)	
outMetaTxManageHistories	Operating record	array	OutMetaTxManageHistory
blockNumber	Block number	number	
bnDeductGasFeeAccount	Business Node deduct gas fee address	string	
failReason	Fail reason	string	
gasFee	Gas fee	string	
processDate	Process date	integer	
receivedDate	Received date	integer	
transactionHash	Meta transaction hash on the platform	string	
transactionId	Meat transaction Id	string	

transactionStatus	Meta transaction status:
	1=Pending;
	2=Permit processing;
	3=Permit processed successfully;
	4=Permit failed;
	5=Transfer processing;
	6=Transfer processed successfully;
	7=Transfer failed;
	8=BN init;
	9=Submitted vn;
		integer	

receivedAmount	Received amount	number(bigdecimal)	
sendAccount	Send account	string	
status	Meta transaction status	string	
submitDate	Submit date	integer(int64)	
tnCode	Transaction Node code	string	
tnName	Transaction Node name	string	
transactionKey	Meta transaction key	string	
message	Message code	string	

Response Example:

```
{
    "code": 0,
    "data": {
        "amount": 1.235,
        "beneficiaryAccount": "0x7392749234092840982",
        "beneficiaryName": "Beneficiary Name",
        "bnDeductGasFeeAccount": "0x7392749234092840982",
        "channelFeeCurrency": "Ether",
        "currencyType": "USDC-ETH",
        "failReason": "fail",
        "isKyc": 1,
        "outMetaTxManageHistories": "
[{\"transactionId\":\"Meta71f4f57fc7c2402890c4a82159b5fcde\",\"blockNumber\":nul
l,\"transactionHash\":null,\"bnDeductGasFeeAccount\":null,\"gasFee\":null,\"rece
ivedDate\":1704472682,\"processDate\":null,\"transactionStatus\":8,\"failReason\
":null},
{\"transactionId\":\"Meta71f4f57fc7c2402890c4a82159b5fcde\",\"blockNumber\":null
,\"transactionHash\":null,\"bnDeductGasFeeAccount\":null,\"gasFee\":null,\"recei
vedDate\":1704472682,\"processDate\":null,\"transactionStatus\":9,\"failReason\"
:null}]",
        "receivedAmount": 0,
        "sendAccount": "0x7392749234092840982",
        "status": "Pending",
        "submitDate": 0,
        "tnCode": "TN111",
        "tnName": "testName",
        "transactionKey": "Metae29a2f631c644e0c8bb475309814a1c0"
    },
    "message": "ok"
}
```

08.Reset nonce.

url: `/v1/udpn/processing/metatx/manage/reset/nonce`

method: `POST`

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

consumes: `*/*`

Note:

When the nonce is confused due to multiple calls to the 02 interface, you need to call this interface to reset the nonce.

Example:

```
{
  "sourceAccountAddress": "0x78AB5678CDF8907654909089765432345906AAAC"
}
```

Params:

name	description	in	require	type	schema
inMetaTransactionManageResetNonce	inMetaTransactionManageResetNonce	body	true	InMetaTransactionManageResetNonce	InMetaTransactionManageResetNonce
sourceAccountAddress	Source account address. Remark:max length is 100		true	string	
apiToken		header	false		

Status:

code	description	schema
200	OK	ResultInfo«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

Response Params:

name	description	type	schema
code	Return code.Remark:code is 0 means success	integer(int32)	integer(int32)
data	Api return data	boolean	
message	Message code	string	

Response Example:

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