

梧桐链

SDKV2开发手册



梧桐链数字科技研究院

读者对象

需要基于梧桐链开发上层应用的开发者或者组织，具有一定的开发能力

Web API 接口

接口调用说明

SDK Web API是构建于http协议之上，使用了JSON数据编码格式，并以Restful架构方式发布的一组与梧桐链进行互操作的服务接口，通常是部署在局域网中，提供给参与方的业务系统使用。

梧桐链实现了应用链的架构，支持多条链并行运行，在调用Web API时，可以通过以下方式来指定需要访问哪一条链：

https请求

访问应用链（异步方式）：**https://host:port/v2/xxxx/xxx?**

ledger=blkc2022

访问应用链（同步方式）：**https://host:port/v2/xxxx/xxx?**

ledger=blkc2022&sync=true

- ledger指定应用链ID
- 所有请求参数建议采用驼峰命名方式
- 所有返回参数采用驼峰命名法分割命名方式
- 所有URL均小写
- ledger属于必传字段
- 同步方式调用需要开启事件通知功能

接口返回数据包含**公共参数**和**业务参数**，所有接口都会返回公共参数，不同的接口返回的业务参数不同

公共返回参数说明

```
{
  "state":200,
  "message":"success",
  "data": {any}
}
```

参数	说明
state	接口请求返回码， 详情参考
message	接口请求成功或错误描述
data	任意类型，返回的具体的业务参数

返回码说明

返回码	说明
200	请求成功
400	客户端参数错误
404	查找的实体不存在
500	服务端内部错误
503	请求超时

接口列表

API	请求方式	说明
/v2/block/height	GET	获取区块链的最新高度
/v2/block/detail/{id}	GET	根据区块ID获取区块详情
/v2/block/detail/{height}	GET	根据区块高度获取区块详情
/v2/block/height/{txId}	GET	根据交易ID获取交易所在区块高度
/v2/tx/store	POST	发送存证交易

API	请求方式	说明
/v2/tx/store/query	POST	获取隐私存证交易内容
/v2/tx/store/querybypubkey	POST	获取隐私存证交易内容（不传私钥）
/v2/tx/store/authorize	POST	隐私存证授权
/v2/tx/store/authorizebypubkey	POST	隐私存证授权（不传私钥）
/v2/tx/detail/{txid}	GET	查询交易详情
/v2/tx/raw/{txid}	GET	获取交易原始数据(可以查询所有类型交易)
/v2/tx/sc/install	POST	安装合约
/v2/tx/sc/invoke	POST	调用合约
/v2/tx/sc/query	POST	合约查询
/v2/tx/sc/destory	POST	销毁合约
/v2/block/tx/index/{txId}	GET	根据交易ID获取交易在区块的索引
/v2/tx/inlocal/{txId}	GET	根据交易ID查询交易是否同步到数据库
/v2/chain/apihealth	GET	获取系统健康状态
/health	GET	检测应用链是否可以正常运行
/v2/ledger	GET	获取应用链信息
/v2/memberlist	GET	获取节点列表详细信息
/v2/tx/token/issue/apply	POST	smart token发行申请
/v2/tx/token/issue/approve	POST	smart token发行审核
/v2/tx/token/transfer/apply	POST	smart token转让申请
/v2/tx/token/exchange/apply	POST	smart token转换申请
/v2/tx/token/destroy/apply	POST	smart token销毁（回收）申请
/v2/tx/token/approve	POST	smart token转让、转换、销毁审核
/v2/tx/token/address/generate	POST	根据公钥生成多签地址
/v2/tx/token/balance	POST	根据地址查询余额
/v2/tx/token/tokenype/freeze	POST	smart token冻结
/v2/tx/token/tokenype/recover	POST	smart token解冻
/v2/event/msg/{height}	GET	根据区块高度查询事件信息

API	请求方式	说明
/v2/event/msg/tx/{txId}	GET	根据交易ID查询事件信息

1、区块接口

提供了一组区块访问的接口，方便快速获取区块相关信息。

1.1、获取最新区块高度

返回区块链网络中当前已确认的最新区块高度。

接口调用请求说明

```
https请求方式：GET
https://host:port/v2/block/height?ledger=blkc2022
```

请求示例

```
https://10.1.3.160:8080/v2/block/height?ledger=blkc2022
```

请求参数说明

无

返回说明

正常返回结果示例：

```
{
  "state": 200,
  "message": "success",
  "data": 100
}
```

返回参数说明

参数	说明
data	当前最新区块高度

错误返回结果示例(连接节点失败):

```
{
  "state": 500,
  "message": "grpcQueryInvoke err:ERROR[1]:rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing dial tcp 10.1.3.159:9000: connect: connection refused\";",
  "data": null
}
```

错误返回结果示例(未传应用链id):

```
{
  "state": 400,
  "message": "ledger id should be assigned",
  "data": null
}
```

1.2、根据区块ID获取区块详情

根据区块ID返回对应区块详情。

接口调用请求说明

```
https请求方式: GET
https://host:port/v2/block/detail/{id}?ledger=blkc2022
```

请求示例

```
https://10.1.3.160:8080/v2/block/detail/00d08384565549dd6fbbf
f550fc0b1c2b6472c1eb3ce90f659b32fe5979c0c05?ledger=blkc2022
```

请求参数说明

参数	类型	是否必要参数	说明
id	string	yes	区块id

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": {
    "header": {
      "version": 0,
      "height": 55138,
      "timestamp": 1590976301572,
      "blockId":
"00d08384565549dd6fbbff550fc0b1c2b6472c1eb3ce90f659b32fe5979c
0c05",
      "previousId":
"f0dcaf6c5f078f6d0b85a7bb730af28b69f0f9f6baab9a619dec56fcc7f7
4c33",
      "worldStateRoot":
"d1da1cdc029389b56b95fd282f1ee5f057c3d0c0a4f1096a61f7db43840b
f6a44242424242424242424242424242424242424242424242424242424242
2424242",
      "transactionRoot":
"392a539f3f0483d5481e457e1820012cf007ccef66c74ef8be9d3e4a4abb
e8d0"
    },
    "txs": [

      "7b245a1d3eb1c27f5b0270e0716327e9325e84feab1ea2134e5f529ab88
397f5",

      "8c318b09ad2c63c527286fc6b8d99c8f73b253529c3a956e0f91b51e720
eed7e",

      "66d04cf478349ebddb0b59c223f4fbf51de8e5ff8e0011629ae80b994b
e81e3",

      "8accdd94e130c81867f487e694cbe08cef78a7cfc3dff03301555d5d977
5fe77",

      "494204064cd68938d872e2d0c20f0b636b8503eba71ec9d86bd4b3195c0
39aad",
```

```
"958fac8ad97f003ce03ecfdb5b379bd00b52ed497bc4dd58ac380229dbe
f48e6",

"20be72b01f7571aaa78426fb50484fa953557abf80f58a6245da46c14f7
ce55e",

"ee9943eba1fc7fa50cba34a9a055ba1b4a072d31c2a8e4198573403e31f
cbe06",

"abac167cd66d014e969e25f6788e297a0f02c5c944f032aed2ee0e45924
e2238",

"8888e7fe141d8d8a3489ee12ab3b778ce45dd7b4b56fb33884123b63101
376ec",

"15d1e8525f76b9d59e4c1cfa192a14df79843a677bd3d20b6db0ba7d688
ef57b",

"e6d05c744ed05106db38e850ce76321b8a47b59bcbf4141a5ff497ea4c8
ce796"
  ],
  "extra": ""
}
}
```

返回结果参数说明

参数	说明
header.height	区块所在高度
header.timestamp	区块生成时间戳，值为Unix时间戳
header.blockId	区块id
header.previousId	前一个区块id
header.worldStateRoot	世界状态的根id，用于校验世界状态数据的正确性
header.transactionRoot	根据区块中所有交易计算出Merkle Tree根值
txs	区块中所有交易id列表

错误返回结果示例(没有找到高度为60000的区块):

```
{
  "state": 404,
  "message": "grpcQueryInvoke err:ERROR[1]:rpc error: code = Unknown desc = BlockchainGetBlockByHeight: failed to find block:60000;",
  "data": null
}
```

1.3、根据区块高度获取区块详情

根据区块高度返回区块详情。

接口调用请求说明

```
https请求方式：GET
https://host:port/v2/block/detail/{height}?ledger=blkc2022
```

请求示例

```
https://10.1.3.160:8080/v2/block/detail/55138?ledger=blkc2022
```

请求参数说明

参数	类型	是否必要参数	说明
height	number	yes	区块高度，创世区块高度为0

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": {
    "header": {
      "version": 0,
      "height": 55138,
      "timestamp": 1590976301572,

```

```
    "blockId":
    "00d08384565549dd6fbbff550fc0b1c2b6472c1eb3ce90f659b32fe5979c
    0c05",
    "previousId":
    "f0dcaf6c5f078f6d0b85a7bb730af28b69f0f9f6baab9a619dec56fcc7f7
    4c33",
    "worldStateRoot":
    "d1da1cdc029389b56b95fd282f1ee5f057c3d0c0a4f1096a61f7db43840b
    f6a4424242424242424242424242424242424242424242424242424242424
    2424242",
    "transactionRoot":
    "392a539f3f0483d5481e457e1820012cf007ccefc66c74ef8be9d3e4a4abb
    e8d0"
  },
  "txs": [

    "7b245a1d3eb1c27f5b0270e0716327e9325e84feab1ea2134e5f529ab88
    397f5",

    "8c318b09ad2c63c527286fc6b8d99c8f73b253529c3a956e0f91b51e720
    eed7e",

    "66d04cf478349ebddb0b59c223f4fbf51de8e5ff8e0011629ae80b994b
    e81e3",

    "8accdd94e130c81867f487e694cbe08cef78a7cfc3dff03301555d5d977
    5fe77",

    "494204064cd68938d872e2d0c20f0b636b8503eba71ec9d86bd4b3195c0
    39aad",

    "958fac8ad97f003ce03ecfdb5b379bd00b52ed497bc4dd58ac380229dbe
    f48e6",

    "20be72b01f7571aaa78426fb50484fa953557abf80f58a6245da46c14f7
    ce55e",

    "ee9943eba1fc7fa50cba34a9a055ba1b4a072d31c2a8e4198573403e31f
    cbe06",

    "abac167cd66d014e969e25f6788e297a0f02c5c944f032aed2ee0e45924
    e2238",
```

```
"8888e7fe141d8d8a3489ee12ab3b778ce45dd7b4b56fb33884123b63101376ec",

"15d1e8525f76b9d59e4c1cfa192a14df79843a677bd3d20b6db0ba7d688ef57b",

"e6d05c744ed05106db38e850ce76321b8a47b59bcbf4141a5ff497ea4c8ce796"
],
"extra": ""
}
}
```

返回结果参数说明

参数	说明
header.height	区块所在高度
header.timestamp	区块生成时间戳，值为Unix时间戳
header.blockId	区块id
header.previousId	前一个区块id
header.worldStateRoot	世界状态的根id，用于校验世界状态数据的正确性
header.transactionRoot	根据区块中所有交易计算出Merkle Tree根值
txs	区块中所有交易id列表

错误返回结果示例(没有权限):

```
{
  "state": 500,
  "message": "grpcQueryInvoke err:ERROR[1]:rpc error: code = Unknown desc = no permissions;",
  "data": null
}
```

1.4、根据交易ID获取交易所在区块高度

接口调用请求说明

`https请求方式：GET`

`https://host:port/v2/block/height/{txid}?ledger=blkc2022`

请求示例

`https://host:port/v2/block/height/7b245a1d3eb1c27f5b0270e0716327e9325e84feab1ea2134e5f529ab88397f5?ledger=blkc2022`

参数说明

参数	类型	是否必要参数	说明
txId	string	yes	交易id

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": 55138
}
```

返回结果参数说明

参数	说明
data	交易所在区块高度

2、交易接口

『交易』是一组对区块链操作行为的统称，沿用了比特币使用的概念，一笔『交易』既可以是一个简单的数据存证操作，也可以是一个对智能合约的调用执行指令，每一笔交易都有一个长度为256 bits的全局唯一ID，通过对交易内容进行Hash计算得到，所有的交易历史记录会按预定的规则分组保存于区块中，『交易』是区块链网络中最小的数据单元。

梧桐链支持存证交易、数字资产交易，智能合约交易等多种交易类型操作，方便企业根据自身的业务场景灵活选用。

2.1、存证交易接口

存证交易是指把数据原封不变直接存储在区块链网络中，借助区块链的链式存储特征来实现数据共享且不可篡改的目的，比较常用的场景如：食药溯源，数字版权，电子合同等。为了应对企业对数据隐私安全保护的要求，梧桐链支持数据加密存储(结合非对称和对称加密算法)，并支持数据按需授权访问。

众所周知的，受限于分布式账本技术的运行原理，区块链网络中不适合存储体积较大的数据，对于大体积数据(>=100KB)，我们建议原数据由业务系统保存，链上只保存数据的特征值(Hash值)，用于验证数据的准确性。

2.1.1、数据存证

数据存证上链，支持明文保存和加密保存两种方式。

接口调用请求说明

https请求方式：POST

https://host:port/v2/tx/store?ledger=blk2022

post请求示例

https://host:port/v2/tx/store?ledger=blk2022

```
{
  "data": "hello wutongchain"
}
```

```
{
  "pubkeys": [

    "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1lIS29aSxpqMEN
    BUVlJS29FY3oxVUJnaTBEUwdBRWQxdkt0aGhISmw3SWZFbHNWN3A5QmZVWEtY
    SWcNCnRhYm8zeEZyNU41d2x4VnRCbTAXVDEwQjlQSFYZOWthYXFNaS9XcURPM
    01ucStDMmJ6bmpSa04wMGc9PQ0KLS0tLS1FTkQgUUFVCTE1DIEtFWS0tLS0tdQ
    o=",

    "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1lIS29aSxpqMEN
    BUVlJS29FY3oxVUJnaTBEUwdBRU1UYjc5WHRSUEJOb0dCaEtEZUpCemZPeXhT
    UVoNCjVksUFTOXo1WgdZL1gwSTNiU1d5WUF4UjBRNXpmWjFSNEZtMGxDUCtPM
    Tk4Q2FkdKf0L045TEVVQ0E9PQ0KLS0tLS1FTkQgUUFVCTE1DIEtFWS0tLS0tdQ
    o=",

    "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1lIS29aSxpqMEN
    BUVlJS29FY3oxVUJnaTBEUwdBRWlnUFZ5dnp0Y05YcXRtN3BiYURPYzdHRVhy
    cWONckZrM1dsU1VvckNiYDR5LzBIWTFFU3ozNEcxSUZ2VHDHa3Vudk5HRzRMe
    dI15Q0VwOEhIVm84NFZEV1E9PQ0KLS0tLS1FTkQgUUFVCTE1DIEtFWS0tLS0tdQ
    o="
  ],
  "data": "test8651"
}
```

参数说明

参数	类型	是否必要参数	说明
data	string	yes	要存的数据
pubKeys	string array	no	公钥，传递该参数为隐私存证，多个公钥可以使用任意对应的私钥查询该笔交易（公钥为base64字符串）

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
    "259626dcfc9e67100166f3a763c0b9b99dc7bd51e1ab6287076a46bf30b61d23"
}
```

返回结果参数说明

参数	说明
data	交易id

2.1.2、存证数据查询

获取存证在链上的用户数据。

2.1.2.1、 获取普通存证内容

请参考2.3.1获取交易详情接口

2.1.2.2、 获取隐私存证内容

接口调用请求说明

```
https请求方式： POST
https://host:port/v2/tx/store/query?ledger=b1kc2022
```

post请求示例

```
https://host:port/v2/tx/store/query?ledger=b1kc2022
```

```
{
  "txId":
"56a1bb7427c24c0979d9ee6272e067e2ca1af0877052493642cfbbc31698
78a5",
  "prikey":
"LS0tLS1CRUdJTiBQVklwQVRFIETfWS0tLS0tDQpNSUdUQWdFQU1CTUdCeXFH
U000OUFnRUdDQ3FCSE05VkFZSXRCSGt3ZHdJQkFRUWdyUXBFS0VwdVpDakNFZ
C9RDQpaNS8rVHZRc2NwYi92UHVsaUgzYndnV0xyVUdnQ2dZSutvRWn6MVVCZ2
kyaFJBTKNBQVizVzhxMkdFY21Yc2g4DQpTV3hydW4wRjlSY3BjaUMxCHVqZkV
XdmszbkNYRlcwR2JUUVlBYUUGwOGRYZjJSCHFveUw5YW9NN2N5ZXI0TFp2DQpP
ZU5HUTNUUw0KLS0tLS1FTkQgUUFJJVkfURSBkRVktLS0tLQ0K",
  "password": ""
}
```

参数说明

参数	类型	是否必要 参数	说明
txId	string	yes	数据存证接口返回的交易ID，16进制字符串
priKey	string	yes	用户私钥，用于解密隐私存证数据，base64字符串
password	string	no	用户私钥密码，可选

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": "Hello wutong Chain!"
}
```

没有权限返回结果：

```
{
  "state": 500,
  "message": "you have no permission to get this transaction!",
  "data": null
}
```

2.1.2.3、 获取隐私存证内容（不传私钥）

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/store/querybypubkey?ledger=blkc2022

post请求示例

https://host:port/v2/tx/store/querybypubkey?ledger=blkc2022

```
{
  "txId":
  "56a1bb7427c24c0979d9ee6272e067e2ca1af0877052493642cfbbc3169878a5",
  "pubkey":
  "LS0tLS1CRUdJTiBQVk1wQVRFIetFWS0tLS0tdQpNSUdUQWdFQU1CTUdCeXFHU000OUFnRUdDQ3FCSE05VkFZSXRCSGt3ZHDJQkFRUwdyUXBFS0VwdVpDakNFZC9RDQpaNS8rVHZRc2NwYi92UHVSaUgzYndnV0xyVUdnQ2dZSutvRWN6MVVCZ2kyaFJBTKNBQVizVzhxMkdFY21Yc2g4DQpTV3hYdw4wRjlSY3BjaUMxchVqZkVxdmszbkNYRlcwR2JUUVlBYUUGwOGRYZjJSCHFveUw5YW9NN2N5ZXI0TFp2DQpPZU5HUTNUUw0KLS0tLS1FTkQgUFJJVkfFURSBLRVktLS0tLQ0K"
}
```

参数说明

参数	类型	是否必要参数	说明
txId	string	yes	数据存证接口返回的交易ID，16进制字符串
pubKey	string	yes	用户公钥，用于验证用户身份

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": "{\\\"SecretData\\\":\\\"80fd93d292a0b789c210f6674616260f65abdbfd039589e1be6dac18e0ef29f1\\\",\\\"SecretKey\\\":\\\"043cc124135ca9452189dc1968ec894da212b904e2346c7702cd3807d2ea1315f20ebd1bfeb6e2960e8422119efd61b08068bc0d95c57cb48168af2b6cc85b8f4a5ae90d4f56cb8071adeaad657bebb133e006414451b9effdda007fad82ccaad37df2ccc9307c8e3e0d606dbc0ce65cac\\\"}"
}
```

返回的Data中json字符串的说明

字段	说明
SecretData	由sm4加密存证原文得到的密文，其字符编码格式为16进制
SecretKey	一段由存证公钥加密得到的密文。该密文的原文为加密存证原文的sm4的密码，即加密得到SecretData的密码。

没有权限返回结果:

```
{
  "state": 500,
  "message": "you have no permission to get this transaction!",
  "data": null
}
```

2.1.3、隐私存证授权

将在链上的隐私存证授权给其他人查看

接口调用请求说明

https请求方式： POST

https://host:port/v2/tx/store/authorize?ledger=b1kc2022

POST请求示例

https://host:port/v2/tx/store/authorize?ledger=b1kc2022

请求示例

```
{
  "pubkeys": [
    "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1lIS29aSXpqMEN
    BUVlJS29FY3oxVUJnaTBEUwdBRWlnUFZ5dnp0Y05YcXRtN3BiYURPYzdHRVhy
    cWONckZrM1dSU1VvckNiYDR5LzBIWTFFU3ozNEcxSUZ2VHdHa3Vudk5HRzRMe
    dI15Q0VwOEhIVm84NFZEV1E9PQ0KLS0tLS1FTkQgUUFVCTE1DIETfWS0tLS0tDQ
    o="
  ],
  "password": "222",
  "prikey":
    "LS0tLS1CRUdJTiBFTkNSWVBURUQgUUFJJVkfURsBLRVktLS0tLQ0KTU1IOE1G
    Y0dDU3FHU01iMORRRUZEVEJLTUNrR0NTCudTSWIZRFFFRkREQWNCQWkwCjdQM
    kphZDczQU1DQ0FBdw0KREFZSutvWklodmNOQWdjRkFEQWRCZ2xnaGtnQ1pRTU
    VBU29FRU9CY2hov3NlQXQzTmwzMGZUTGtJVkFFZ2FERQ0KWG56dmpsVEVvNlh
    xY1RvMmRlWGtsL1BiUUtJSHFCYzdEaHZZK1hmU1ZVOGxHas8zR09GZjBMNDdw
    K0FGUjAvag0Kem1RMU9ZOGJFbg1QZ1hzQ1Jhwm1ubTR2V2cyQU1pMk9HQi90a
    k1XS3BoewxMREJSN3lnUESrvkxTNjn4V1NiZw0KCHZ3QUVxK0xVemJvd1JQLy
    tORghyUk1qaE1LS0JhbW1Lb1UrT09krktkQmNzUVE3UEJGUW5qNGhwdTJxakp
    Tcw0Kd1MycGYrb3F3WUozdHJaWER0ewINCi0tLS0tRU5EIEVOQ1JZUFRFRCBQ
    Uk1WQVRFIETfWS0tLS0tDQo=",
  "txId":
    "2ad2d83595d1cc342b88ad030097a936e3d2912c9d4a9f7e968606fd3718
    415a"
}
```

参数说明

参数	类型	是否必要 参数	说明
txId	string	yes	数据存证接口返回交易txid值
pubKeys	string array	yes	希望被授权的用户公钥，可以一次授权多个用户
priKey	string	yes	存证交易创建人的私钥，base64编码字符串
password	string	no	私钥密码，无密码时可以不填。

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
    "8558f357d49ad63448b5f2151f86d344d6ecb0deca3b7d23d9f7660da25a6d9b"
}
```

2.1.4、隐私存证授权（不传私钥）

将在链上的隐私存证授权给其他人查看

接口调用请求说明

```
https请求方式： POST
https://host:port/v2/tx/store/authorizebypubkey?
  ledger=blkc2022
```

POST请求示例

```
https://host:port/v2/tx/store/authorizebypubkey?
  ledger=blkc2022
```


请求示例

```
{
  "txId":
    "2ad2d83595d1cc342b88ad030097a936e3d2912c9d4a9f7e968606fd3718415a",
  "secrets": [{
    "PubKey": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1lIS29aSXpqMENBUVlJS29FY3oxVUJnaTBEUWdBRUFyL2ZRanlIdW5DZk9xv2FTcwo5VGE0ZWVudWENCldtewR3YndvdWFIQXhpU1crNG1oZmRlVTVZS90SHRLN2g2Ymg4a2orZ3l3MXVJSnRndXdnNk8xdVE9PQ0KLS0tLS1FTkQgUFVCTE1DIEtFWS0tLS0tDQo=", "SecurityKey": "BLnwpta9n2x7a8FsrXX49X8G443ktikD15S+VUARo3Rjk4E+qXGwkFL30/dOIdovynxCHkdpVqH6B83ykaQ+3GcI6sftgVmnC8K54+bAr/5htYIkpHdzftUq96qcj6M6JKXOTpJztmXUHkeEs4X7R6k="
  }],
  "ownerPubKey": "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1lIS29aSXpqMENBUVlJS29FY3oxVUJnaTBEUWdBRUViUnpwU1FxeKhkL1Rxyi9ZckVFUFRzcGh4MnQNCnNqc0puR0svd3VoZ3Rlc0JHb3Yvb0pwaFNsU2VzTm15TVFibUwvU2tkSDlsc1VvRFNpY0lmSTU0aGc9PQ0KLS0tLS1FTkQgUFVCTE1DIEtFWS0tLS0tDQo=",
  "sign": "MEQCIDZNMJOrF5uXsNzdpHsJ5y3HdgSR07dzj10ueRBEn6DfAiBDEDHdi9C182xbSkntZRZECykFXYtIOH3RJXcts1xFeA=="
}
```

参数说明

参数	说明
txId	数据存证接口返回交易的交易id值
secrets	希望被授权的用户公钥+该公钥加密的密码（密码由获取隐私存证的SecretKey解密得到）
ownerPubKey	存证交易创建人的公钥
sign	使用OwnerPubKey对应的私钥，对Secrets传递的对象转换的json字符串签名

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
    "8558f357d49ad63448b5f2151f86d344d6ecb0deca3b7d23d9f7660da25a6d9b"
}
```

2.2、 其他交易接口

2.2.1、 根据交易ID获取交易在区块的索引

接口调用请求说明

https请求方式：**GET**

https://host:port/v2/block/tx/index/{txId}?ledger=blkc2022

GET请求示例

https://host:port/v2/block/tx/index/a439f65f6db2e6052f95daff74bbab2440621220852d27022e6cd50afe409741?ledger=blkc2022

参数说明

参数	类型	是否必要参数	说明
txId	string	yes	交易id

返回说明

返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": 0
}
```

返回结果参数说明

参数	说明
data	交易在区块中的索引

2.2.2、 根据交易ID查询交易是否同步到本地数据库

使用这个接口需要开启钱包功能

接口调用请求说明

https请求方式：GET

https://host:port/v2/tx/inlocal/{txId}?ledger=blkc2022

GET请求示例、

https://10.1.3.150:9999/v2/tx/inlocal/a439f65f6db2e6052f95daf74bbab2440621220852d27022e6cd50afe409741?ledger=blkc2022

参数说明

参数	类型	是否必要参数	说明
txId	string	yes	交易id

返回说明

返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": {
    "exist": true
  }
}
```

返回结果参数说明

参数	说明
data.exist	true表示本地数据库已同步

2.3、获取交易详情

根据交易id获取交易详情数据。

2.3.1、 获取交易详情

接口调用请求说明

https请求方式：GET
https://host:port/v2/tx/detail/{txid}?ledger=blkc2022

请求示例

**https://host:port/v2/tx/detail/fe60e43f99e89d94bd0935b784cffd
adba2286c1225652d74304f7fcb10130a6?ledger=blkc2022**

参数说明

参数	类型	是否必要参数	说明
txId	string	yes	交易id, 16进制

返回说明

返回结果参数说明

参数	说明
header	交易头对象
-header.version	交易版本号
-header.timestamp	时间戳(int64精确到毫秒,开始于 January 1, 1970 UTC.)
-header.type	交易大类(主要分为:0-存证;1-UTXO;2-合约;4-系统交易;20-链管理类交易)
-header.subType	交易子类型,每个大类的交易下还会细分子交易类型 存证子类型:0-基础存证;1-加密存证; 合约子类型:子类型: 30-安装;31-销毁 UTXO子类型:10-发行;11-转账;12-回收; 系统交易:0-修改节点;1-删除节点;3-权限交易;4-创建应用链;5-冻结应用链;6-解冻应用链;7-销毁应用链;8-升级系统合约;9-冻结合约;10-激活合约;11-节点关断;12-节点关断恢复;13-配置合约方法权限; 链管理类交易:0-注册归集地址;1-删除归集地址;2-注册发行地址;3-删除发行地址;4-tokenType冻结;5-tokenType解冻
-header.transactionId	交易id
store	存证对象
-store.storeData	具体存证内容(隐私存证返回加密数据)
-store.extra	额外字段(预留字段)
wvm	合约交易对象
-wvm.wvmContractTx.arg	合约交易参数
-wvm.wvmContractTx.name	合约地址
-wvm.wvmContractTx.src	合约代码,只有安装合约交易才会返回。

参数	说明
-wvm.result	合约执行结果，合约执行过程中发生变化的KV
utxo	utxo交易对象
-utxo.data	备注信息
utxo.txRecords	utxo交易记录列表
-utxo.txRecords.from	转出账号,支持单签和多签账户
-utxo.txRecords.to	转入账号,支持单签和多签账户
- utxo.txRecords.tokenType	数字资产类型
-utxo.txRecords.amount	数字资产转移数量
raw	交易原始数据

存证交易返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": {
    "header": {
      "version": 0,
      "type": 0,
      "subType": 0,
      "timestamp": 1606717185209,
      "transactionId":
"085ac42d8faa5304a43a2ddb020eab8646a8d30b1889912b3d66d98c24f24f"
    },
    "store": {
      "storeData": "{\"test\":\"jsonnCj3\"}",
      "extra": null
    }
  },
}
```

```
    "raw":  
    "000000000000000000000000b99cce177601000035000000007b22446174  
61223a2265794a305a584e30496a6f69616e4e76626d3544616a4d6966513  
d3d222c22536563726574223a5b5d7d0a0000000040000000c1f99857a142  
55726bed01fe60f19eb7c8ab9863b9992a116ff22cbaff49efec2f8933e25  
81432a564807917d46307dd7b0db5d19b2bbd72a12d12ad3c273c17000000  
00"  
  }  
}
```

隐私存证交易返回结果

```
{  
  "state": 200,  
  "message": "success",  
  "data": {  
    "header": {  
      "version": 1,  
      "type": 0,  
      "subType": 1,  
      "timestamp": 1606717214339,  
      "transactionId":  
      "2f78eddb4276af5729ffebf6fbaffc0c952e50dacf20b15cccd77b62635c  
1c1b"  
    },  
    "store": {  
      "storeData":  
      "a85a48dd1973db12c61e7f8c21020af5f0d56b543ddb94e282c64011f672  
7593",  
      "extra": null  
    },  
  },  
}
```

```
"raw":
"010000000000000001000000830ecf177601000058020000007b22446174
61223a227146704933526c7a32784c47486e2b4d4951494b3966445661315
1393235546967735a4145665a79645a4d3d222c22536563726574223a5b7b
225075626b6579223a22416e64627972595952795a657948784a624665366
651583146796c79494c576d364e3852612b5465634a635662515a744e5539
644166547831642f5a476d716a49763171677a747a4a367667746d3835343
05a44644e493d222c22436970686572223a224248364d432f67474342546f
52517537384c762b4149445a624e4538797474425a634f544932765646753
7324d7675444a49506c5a475345652b7a6b47504e5a626c5930526e567331
7271426e48784b53664a53622b36724d41413465704570793763375033343
45153576d506770432f56592f70616f777a58504367486a31594c6a447830
5061694f7978364c79534b583759616f513d227d2c7b225075626b6579223
a2241735475376946454a676d313431486a38322f72356c514b65354b4853
4c4e45666b4d66526963446f57427866646b4342547454557350334d7a457
570334f386b34397074787939596668787747493751516268634b4d3d222c
22436970686572223a22424a707870384b5962457235544e6f756154494f5
46f64376d4e343642504a2b4550674d5650616c644d5a694345716a554751
617a2b4534744259743657303547795366354f636a316e6a64506438436e7
147324631424d442f703273773170782f3567796762737653485272774e68
504e536939552b2f6b35477a746b7838516864427a5431636258346349716
12f4c686f414578553d227d5d7d0a0000000040000000c1f99857a1425572
6bed01fe60f19eb7c8ab9863b9992a116ff22cbaff49efec2f8933e258143
2a564807917d46307dd7b0db5d19b2bbd72a12d12ad3c273c1700000000"
}
}
```

utxo交易返回结果

```
{
  "state": 200,
  "message": "success",
  "data": {
    "header": {
      "version": 1,
      "type": 1,
      "subType": 11,
      "timestamp": 1606802816051,
      "transactionId":
"3264f9f04e056eb200c2a3fcce50d42e0033edd31860e63a5003664347e2
fe4f"
    },
  },
}
```



```
"utxo": {
  "txRecords": [
    {
      "from":
"4Mh3kKyvqjhw1EXQwUr7DTrwV3x8fAoQnH1RnjdAUvEYYa4u7F",
      "to":
"SnvgtVkq96kkGgspgxu6ReeNFQPzY14M71wk9Lxp3WnyDGizJkZ",
      "tokenType": "2020120114004086445",
      "amount": "1002",
      "subType": "0"
    }
  ],
  "data": ""
},
```

```
"raw":
```

[illegible]

53384f6a73646d6f533030464938734e6233356545456f5665466a4648757
6723136384637535652723564393559634c44337561717a66794f44436357
5438456c734e734957416f523073316a6b7953494176342b57564b6369574
c784b793258444371662f496835375957744337304b2f4478516d59477a6b
4378674f634e5073327a72652b4c5261395967394c4163486f65775531656
966704374624837387569302f354244484434525769776f747657574c574f
6256315430375555692f4130646c75526a2f6d304639586a6c55574667474
8687051726b63347855694476512b6870467075474d34626c5256632b2b4d
56777142624a755252632f427954793343507767796975757165483269592
b6a71735a5a2f71772f45664b2b57764f63434c597432656d575275583457
78424866476f476e5a2b71586d6633474c594256543966784d50485768465
36f4d424951712b574170674232306c356742586c4359457647563061716f
6c63557a72566641515639554e455955626b786f4653396e6c784b62786a4
6727a463675344e2f763959626232665778436d326c52356b6d3652612b62
533559377a37656e783533557961455a4b5873303559486e64682b4b38614
f5450695173557537663844544f366d6e646e4c54764b786d5175336c3044
5a6d5454336b51504735336366776b563356395845574b536f325a54652b3
158646d5041627a4a666c5a48504b37584b31344656654664456d53576a36
4239445975654c6775352b534b7351624236334f4a414e4361694e4d74716
943662b737076636744322f786478724f687a365652425474745a70624756
4556634a2f485862392f73586459374e32317162494a30734f74472f76717
865767454306b676133557333524b6f6168456d2f4653725a6d3031416272
323564526555396f632b636c7036363070596653464e77502f45394757497
364304e79774e343765756966612f6c572b345764494c446338774e6a3749
2b514c6a7a4b6a3571595a34536a6b66705069334f613262656a2f5955743
64a7a48744b6a46393447314d2b5075544a6848636b65674a684f64413362
6a686641344e3957374c657337544f594b5746585438326c6f4f73626f4c6
c784a5371712b4a774d756f4e354f44753450554534385648513033657777
584f47545962576f6f6738796e6b4d55657a59757055496c546439796f444
d74687472596e37394e3343784d78305a7943586159635a5078464d7a6b65
3443614b33596e38386a435544427a566e535961504c6e694376576d51566
9697938314e6557725179672b676246534c4e436b4b327538387957486474
4a732b5764463759466f4f737a514534477a75324241776c7753457956775
14c4969424a4a73497861374a46354d324448726f316c7130633048416a4d
4a53496749677a53634d36376b6e4132734e73576375744f48595a5651724
34162533470426a3668716d6e4567396e73733752416635437152726f4775
51727a36366e4f305a41386c41623436304763624654454258376b2b59797
a4b4f643164703534356b6a2f47357a695867726345554a66565271657233
794562554a674d46516734544a4a7739746f71344a68796e68617556706a6
c486c2b584e497a56664c6673357873725634643667634234437a76776e39
714f77614f6b316532446744334375394379507775646c426d32453639755
248442b6243515538547672424437724556325233744870463830472b676b

```
59724d4e6d376a6f7843415a587432617a696d452b6b6e6e525a324752563
13356766e624844346e637475685768445249582b466559633743757a4338
4d647a71534933782b46473443734c4c4d43596a6758482b67397a5638727
26c584c4a4654574c6b4a5657475a7a5832733049572f3556554c62706e6c
595565576676565864756c6f58615158574f4c347562696f4d784d5953412
f376243517476434b5454556a556e4f62552f2f456f4f6275704963686e36
6a504d47564970473250782b46725875656c3034547648635a545a4f57427
85a50344759726e75546951457357302f2f7942335633434543556e446e47
2b4167754e616e413556315438307073476142633675675337517456576e3
24c495658366a775a583871756c72625a47554b482b3537497868316c7737
67473832353458316831413966326c3763784d4b59444c6154726d554b593
87a6766745356684a56645c227d225d0000000000000000400000005f63f1
a2338eca071b8befaba342c00f3efe405ce89f76e5d9878b661ad79a79d4a
ae32ab5b21f3dae95a1e378c7eab3d659fb16509716b46d98e1dea45ef142
00000000"
  }
}
```

合约交易返回结果

```
{
  "state": 200,
  "message": "success",
  "data": {
    "header": {
      "version": 2,
      "type": 3,
      "subType": 42,
      "timestamp": 1606717740391,
      "transactionId":
"0bdf5972ba63dd0b53f5ad3bcdeb3d3aa246cd3596a720b10351abd81b35
0301"
    },
    "wvm": {
      "wvmContractTx": {
        "arg": {
          "method": "initAccount",
          "caller":
"wfmYV6FCVXJr7QH+YPGet8irmGO5mSoRb/Isuv9J7+wwiTPiWBQypWSAerfu
Ywfdew210ZsrVXKhLRKtPCC8Fw==",
          "version": "",
          "args": [
```

```

        "A",
        50
    ]
},
"name":
"6a0b4d2106016e69b468ad518fad416b8af4762076c75cd0b9024c5c712e
fe89"
},
"result": "{\\"Return\\":\\"success\\",\\"DBSet\\":\\"
[{\\"\\\\"KVPairList\\"\\":
[{\\"\\\\"key\\"\\":\\"\\\\"A\\"\\",\\"\\\\"value\\"\\":\\"\\\\"2\\"\\\\"\\u0000\\"\\\\"\\u0000
\\"\\\\"\\u0000\\"\\\\"}\\],\\"\\\\"RootHash\\"\\":\\"\\\\"v8FrRr62oFMg1jHQkrXJt81A
rFF/qN44jM6FB2ludLA=\\"\\\\"}\\]\\\\"}"
},
"raw":
"02000000030000002a0000006715d71776010000bb0000000b000000696e
69744163636f756e74fffffffff40000000c1f99857a14255726bed01fe60f
19eb7c8ab9863b9992a116ff22cbaff49efec2f8933e2581432a564807917
d46307dd7b0db5d19b2bbd72a12d12ad3c273c1700000000080000005b224
1222c35305d40000000366130623464323130363031366536396234363861
6435313866616434313662386166343736323037366337356364306239303
23463356337313265666538390000000000000000000000000000000000
000040000000c1f99857a14255726bed01fe60f19eb7c8ab9863b9992a116
ff22cbaff49efec2f8933e2581432a564807917d46307dd7b0db5d19b2bbd
72a12d12ad3c273c1700000000"
}
}

```

2.3.2、 获取交易详情原始数据

接口调用请求说明

https请求方式: **get**

https://host:port/v2/tx/raw/{txId}?ledger=blkc2022

请求示例

**https://host:port/v2/tx/raw/f0f9e1ef74876866b39556a2cbc264fa5
c4209ef14f255a7612cda5454eca07a?ledger=blkc2022**

参数说明

参数	类型	是否必要参数	说明
txId	string	yes	交易id, 16进制字符串

返回说明

返回结果参数说明

参数	说明
header	交易头对象
-header.version	交易版本号
-header.timestamp	时间戳(int64精确到毫秒, 始于 January 1, 1970 UTC.)
-header.type	交易大类(主要分为:0-存证;1-UTX0;2-合约;4-系统交易;20-链管理类交易)
-header.subType	交易子类型, 每个大类的交易下还会细分子交易类型 存证子类型:0-基础存证;1-加密存证; 合约子类型:子类型:30-安装;31-销毁 UTX0子类型:10-发行;11-转账;12-回收; 系统交易:0-修改节点;1-删除节点;3-权限交易;4-创建应用链;5-冻结应用链;6-解冻应用链;7-销毁应用链;8-升级系统合约;9-冻结合约;10-激活合约;11-节点关断;12-节点关断恢复;13-配置合约方法权限; 链管理类交易:0-注册归集地址;1-删除归集地址;2-注册发行地址;3-删除发行地址;4-tokentype冻结;5-tokentype解冻
-header.transactionId	交易id
-data	交易内容
-pubkey	发送交易者的公钥
-sign	签名数据
-result	合约执行结果

参数	说明
-extra	扩展字段
-raw	交易原始数据
-txproof	当前交易在区块中的左侧和右侧交易列表

返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": {
    "header": {
      "version": 0,
      "type": 0,
      "subType": 0,
      "timestamp": 1606460863732,
      "transactionId":
"45b0034a79e2df89504665e2f641e493c6e2dae84705ec7292dda16ac70b
208f"
    },
    "data":
"AHsiRGF0YSI6I1NFOUljRlJRuzNwce9WwjFUMUJTT1ZKM01qbENOV2xYYOV3
NVlYTlphr0ZoZec5SlvrouhOMnR2TudwNWJYTmtNVTB5V0RoQ1oxVkJUVTuxZ
EZkUFYzbEJNV1ZpVFRsd1NGSjjZaTgyUTBKUVNrdFlRMnR1ZvdRwvtbE1TVl
ZRV1ZSc2IyOXFWMmxay0VaRk1swNJOVkpHZVVawlptYzBja3B2TwpnNVZEBE1
ZME5LwnpWT1UyaFBLMjVuUkZwTFNqVkJhSGhYV1ZkMmNYuk1im2x2U0d4R1ZH
bHdkVwx0TDFORGMwNVhWMjFsYjB0Q1VwVjZSa2R5UwtkT05uaFRMMjVTYVdOR
GVFTkvaM2xGTkcwd1NVaG5wbFl2TjFkUVNFUjjjMEZaSzBsdfMyaGxZbGRzZE
dGM2NpOUVVe1ZSY1VsbvppdHlPV3A0WlhkdE9YUXpiMGcwYlRkUVpsSkhRV3B
LYnpsVVVFTTRZVFJCYjBOMU5sSlNSMlpQY21sV1pHeDVZa2x6Y0RGWmNsUm1R
VEkzykvJemJXdDBTa1ZKZDNKQ1JqR1lNV015V0RoMVpuRTBVSE4zSzNadGVVR
nBUMDEXTTJwb1BUMD0iLCJTZWNyZXQiOltdfQo=",
    "pubkey":
"f8tuJJH/wz+DvHhSkgukgy8rL/ZPiwqLU+aypJK+R1R+u5l57IgHwwS8rHgt
sppdT42tfbTL6SsxL62BVW4JPw==",
    "sign":
"MEUCIEKAPEaT1Lh+UI+v/vLT2gteHEjfJ3spXhqqVB9uftwTAiEA9cBw5awP
C4Gkl/RwwjxbubGo6ZhI4AyxxErv0e2EDCu=",
    "result": null,
    "extra": null,
  }
}
```

```

"raw":
"AAAAAAAAAAAAAAAAA9HSHCHYBAAD1AQAAHsiRGF0YSI6I1NFOU1jr1JRuzNw
ce9wwjFUMUJTT1ZKM01qbENOV2xYY0V3NV1YT1phR0ZoZEc5S1vrOUhOMnR2T
UdwNWJYtmtNVTB5V0RoQ1oxVkJUVtuxZEZkUFYzbEJNV1ZpVFRsd1NGSjjZaT
gyUTBKUVNrdf1RMnR1ZVdRwvTbE1TV1ZRV1ZSc2IyOXFWMmxaY0VaRk1swNJ
OVkPHZVVaw1ptYzBja3B2TwpnNVZEbE1ZME5LWnpWT1UyaFBLMjVuUkZWTFNq
VkzhSGhYV1ZkMmNYUk1im2x2U0d4R1ZHbHdkVwx0TDFORGMwNVhWMjFsYjB0Q
1VWvjZSa2R5UwtkT05uaFRMMjVTYVdORGvFTkvaM2xGtKcWd1NVaG5wbF12Tj
FkUVNFUjjjMEZaSzBsdFMyaGxZbGRzZEdGM2NpOUVVe1ZSY1VSbvppdH1PV3A
0W1hkde9YUXpiMGcwY1RkUVpsSkhRV3BLynpSVVFTTRZVFJCYjBOMU5sS1NS
M1pQY21sV1pHeDVZa2x6Y0RGwmNSum1RVEkzyKvJemJXdDBTa1ZKZDNKQ1JqR
11NV015V0RoMvpuRTBVSE4zSzNadGVVRnBUMDEXTTJwb1BUMD0iLCJTZWNYZX
Qio1tdfQoAAAAAQAAAAH/LbiSR/8M/g7x4UpILpIMvKy/2T41qi1PmsqSSvkd
UfruZeeyIB8MEvKx4LbKaXU+Nrx20y+krMS+tgbluCT8AAAAA",
  "txproof": {
    "left": [

      "15ea1bf391354b4fdf671f94300d48408b6ab85d6843818828020fa1877
8a441"
    ],
    "right": [

      "a947278fe5872c280b49679b6bd5285e172e616ebe5e6b93b457c8cd662
90e77",

      "bcfb97e540fb05b0016b01360b9b07e774566981497cdd8c429390c7453
50d9e",

      "358939f79431ccf40c722819655911178f6baebbf2ca8918c2be71960e5
6d4de"
    ]
  }
}

```

2.4、合约交易接口

智能合约是由计算机语言编写的，通过事件驱动、具有状态的、通过共识机制获得多方承认的、运行于区块链之上的分布式应用程序。

WVM合约编写指南请参考[合约开发指南](#)。

2.4.1、合约安装

通过P2P网络将合约安装在网络中所有节点上, 合约升级和安装为同一接口。

接口调用请求说明

https请求方式： POST

<https://host:port/v2/tx/sc/install?ledger=blkc2022>

请求示例

```
{
  "category": "wvm",
  "file":
"Ly/1kiJnuqbnPLrkvosKICAgIGNvbnRyYWN0IFRlc3RFeGFtcGx1MiB7CiAg
ICAgICAgLy/1iJ3lp4v1jJbkuIDkuKrotKbmiLcKICAgICAgICBwdWJsawMgc
3Ryaw5nIGluaXQoKXSKICAgICAgICAgICAgc3Ryaw5nw10gbmFtZSA9IHsiYm
9iIiwibWFyeSJ9CiAgICAgICAgICAgICAgIGRiX3NldChuyW1lwzBdLDEwMCKKICA
gICAgICAgICAgZGJfc2V0KG5hbWVbMV0sMTAwKQogICAgICAgICAgICBpbmQg
ayA9IGdldeJSyW5jZShuYW1lwzBdKQogICAgICAgICAgICBwcm1udChrKQogI
CAgICAgICAgICByZXR1cm4gInN1Y2Nlc3MiCiAgICAgICAgfQogICAgICAgIA
ogICAgICAgIC8v6L2s6LSm5pON5L2cCiAgICAgICAgcHVibG1jIHN0cm1uzyB
0cmFuc2Z1cihzdHJpbmcgZnJvbSwgc3Ryaw5nIHRvLCBpbmQgYW1vdW50KSB7
CiAgICAgICAgICAgICgludCBiYwxBID0gZGJfZ2V0PGludD4oZnJvbSkKICAgI
CAgICAgICAgaw50IGJhbEIGPSBky19nZXQ8aw50Pih0bykkICAgICAgICAgIC
AgYmFsQSA9IGJhbEEtYW1vdW50CiAgICAgICAgICAgICglmIChiYwxBPjApewo
gICAgICAgICAgICAgICAgYmFsQiA9IGJhbEIrYW1vdW50CiAgICAgICAgICAgIC
ICAgICBky19zZXQoZnJvbSwgYmFsQSkkICAgICAgICAgICAgICAgIGRiX3Nld
Ch0bywgYmFsQikkICAgICAgICAgICAgfWVsc2V7CiAgICAgICAgICAgICAgIC
ByZXR1cm4gImZhawxlZCIKICAgICAgICAgICAgfSAKICAgICAgICAgICAgcmV
0dXJuICJzdWnjZXNzIgogICAgICAgIH0KICAgIAogICAgICAgIC8v5p+16K+i
6LSm5oi35L2Z6akdCiAgICAgICAgcHVibG1jIGludCBnZXRCbGFuY2Uoc3Rya
w5nIGFjY291bnQpewogICAgICAgICAgICBwcm1udChhy2NvdW50KQogICAgIC
AgICAgICBpbmQgYSA9IGRiX2dlldDxpbnQ+KGFjY291bnQpCiAgICAgICAgICA
gIHJldHVybiBhCiAgICAgICAgfQogICAgICAgIAogICAgfQ=="
}
```

参数说明

参数	类型	是否必须参数	说明
category	string	yes	合约类型
file	string	yes	合约代码内容，需要base64编码
version	string	no	合约版本

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": {
    "txId":
"652992e1dcfdf6fbcf1731d5e3305c72351c3e20284e2d9b7d646d15c72d0554",
    "name":
"bcd27baef3d5ada8cac6402faf940cc4c71ae1a82cd5348bb99ab3bcc77fd346"
  }
}
```

参数说明

参数	说明
txId	交易id
name	合约名字

2.4.2、合约调用

通过指定相关参数触发合约在链上执行。

接口调用请求说明

https请求方式： POST

https://host:port/v2/tx/sc/invoke?ledger=b1kc2022

请求示例

```
{
  "category": "wvm",
  "name":
    "bcd27baef3d5ada8cac6402faf940cc4c71ae1a82cd5348bb99ab3bcc77
    fd346",
  "method": "transfer",
  "args": ["bob", "mary", 1]
}
```

参数说明

参数	类型	是否必须 参数	说明
category	string	yes	合约类型
name	string	yes	合约名字，全局唯一
method	string	yes	指定执行的合约方法
args	array	yes	执行合约方法需要的参数列表
version	string	no	合约版本（安装接口传了该参数时，本字段必传）

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": {
    "txId":
    "f155c934b3fab5011178328510d402e92390b5e3d52c191b29a3241bd4cf
    3a3f"
  }
}
```

返回结果参数说明

参数	说明
txId	交易id

2.4.3、合约查询

接口调用请求说明

https请求方式：POST

https://host:port/v2/tx/sc/query?ledger=blkc2022

请求示例

```
{
  "category": "wvm",
  "name":
  "01f5735cbe64d35b3f4a6d61306e795849a43e428be6ff46e4a15097f4fd
  18b3",
  "args": ["bob"],
  "method": "getBlance",
}
```

参数说明

参数	类型	必须参数	说明
category	string	yes	合约类型
name	string	yes	合约名字
method	string	yes	调用合约方法

参数	类型	必须参数	说明
args	array	yes	执行合约方法需要的参数列表
version	string	no	合约版本

返回说明

```
{
  "state": 200,
  "message": "success",
  "data": {
    "result": "99"
  }
}
```

返回结果参数说明

参数	说明
result	查询结果

2.4.4、合约销毁

销毁合约会致使合约代码文件从链上删除，但合约数据仍存在链上。

接口调用请求说明

```
https请求方式：POST
https://host:port/v2/tx/sc/destroy?ledger=blkc2022
```

请求示例

```
{
  "name":
  "bcd27baef3d5ada8cac6402faf940cc4c71ae1a82cd5348bb99ab3bcc77fd346",
  "category": "wvm"
}
```

参数说明

参数	类型	是否必要参数	说明
name	string	yes	需要被删除的合约名字
category	string	yes	合约类型

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": {
    "txId":
    "b566f3995f2862d4ec439f68db399d126c8190873f0083c7497b01954f32f10d"
  }
}
```

返回结果参数说明

参数	说明
txId	交易id

3、其他功能接口

3.1、 获取系统健康状态接口

接口调用请求说明

```
https请求方式: GET
https://host:port/v2/chain/apihealth?ledger=blkc2022
```

GET请求示例

```
https://host:port/v2/chain/apihealth?ledger=blkc2022
```

返回说明

```
{
  "state": 200,
  "message": "success",
  "data": 0
}
```

返回结果参数说明

参数	说明
message	success表示系统正常

系统不正常

```
{
  "State": 503,
  "Message": "[wutongchain]:grpcQueryInvoke
err:ERROR[1]:rpc error: code = Unavailable desc = connection
error: desc = \"transport: Error while dialing dial tcp
ip:port: connect: connection refused\";ERROR[2]:rpc error:
code = Unavailable desc = connection error: desc =
\"transport: Error while dialing dial tcp ip:port: connect:
connection refused\";;",
  "Data": null
}
```

3.2、 获取应用链信息

接口调用请求说明

https请求方式：**GET**
https://host:port/v2/ledger

GET请求示例（返回所有应用链信息）

https://host:port/v2/ledger

返回结果

```
{
  "state": 200,
  "message": "success",
  "data": {
    "ledgers": [
      {
        "name": "app3",
        "memberList": [
          {
            "id":
            "QmTVRM2yLLNs85owynJrm5fLm4Q9etKe9AnGaMSATK5DFS",
            "port": 9300,
            "shownName": "app1",
            "inAddr": "/ip4/10.1.3.164/tcp/60030"
          },
          {
            "id":
            "QmZZii538SgvDXmEmXaody8Ad9XsFXUDwykjdfkCKcSRy4",
            "port": 9300,
            "shownName": "app2",
            "inAddr": "/ip4/10.1.5.161/tcp/60030"
          },
          {
            "id":
            "QmPqPJzkbLUyHff6m1cwa4h8hg6UwsPGrLtATLDgcgRSTX",
            "port": 9300,
            "shownName": "peer1",
            "inAddr": "/ip4/10.1.3.162/tcp/60030"
          },
        ]
      }
    ]
  }
}
```



```
{
  "id":
  "QmSpqH85rNkcw79ofvjdT8YmJwi ku9qkYmoMLfYwaAfhrT",
  "port": 9300,
  "shownName": "peer2",
  "inAddr": "/ip4/10.1.3.163/tcp/60030"
},
{
  "name": "app1",
  "memberList": [
    {
      "id":
      "QmTVRM2yLLNs85owynJrm5fLm4Q9etKe9AnGaMSATK5DFS",
      "port": 9300,
      "shownName": "app1",
      "inAddr": "/ip4/10.1.3.164/tcp/60030"
    },
    {
      "id":
      "QmPqPJzkbLUyHfF6m1cwa4h8hG6UwsPGrLtATLDgcgRSTX",
      "port": 9300,
      "shownName": "peer1",
      "inAddr": "/ip4/10.1.3.162/tcp/60030"
    },
    {
      "id":
      "QmSpqH85rNkcw79ofvjdT8YmJwi ku9qkYmoMLfYwaAfhrT",
      "port": 9300,
      "shownName": "peer2",
      "inAddr": "/ip4/10.1.3.163/tcp/60030"
    }
  ],
  "hashType": "sm3",
  "word": "app3 test",
  "cons": "raft",
  "timeStamp": 1609229954046,
  "id":
  "b9de6506e13125bf7356147866195708e93f1e2a4aa91bf215a2e07a0176
  732b"
```

```
    "word": "app1 test",
    "cons": "raft",
    "timeStamp": 1609229867901,
    "id":
"5a4e44806cb031d1e9e49c111d81b67843783ac294be504bc07e655acec5
00c3"
  },
  {
    "name": "app2",
    "memberList": [
      {
        "id":
"QmZZii538SgvDXmEmXaodyY8Ad9XsFXUDwykjdfkCKcSRy4",
        "port": 9300,
        "shownName": "app2",
        "inAddr": "/ip4/10.1.5.161/tcp/60030"
      },
      {
        "id":
"QmPqPJzkbLUyHFF6m1cwa4h8hg6UwsPGrLtATLDgcgRstX",
        "port": 9300,
        "shownName": "peer1",
        "inAddr": "/ip4/10.1.3.162/tcp/60030"
      },
      {
        "id":
"QmSpqH85rNkcw79ofVjDT8YmJwi ku9qkYmoMLfYwaAfhrT",
        "port": 9300,
        "shownName": "peer2",
        "inAddr": "/ip4/10.1.3.163/tcp/60030"
      }
    ],
    "hashType": "sm3",
    "word": "app2 test",
    "cons": "raft",
    "timeStamp": 1609229885525,
    "id":
"91d1d4dbb350ca9e158bd6f8953c6965059d63eb315ff3b3f3a16347be8a
cc04"
  }
],
"number": 3
```

```
}  
}
```

返回结果参数说明

参数	说明
number	应用链数量
ledgers.name	应用链名字
ledgers.memberList	应用链节点列表
ledgers.timeStamp	应用链创建时间
ledgers.id	应用链ID
ledgers.hashType	应用链hash算法
ledgers.cons	应用链共识方式
ledgers.word	应用链创世区块信息

3.3、 获取节点列表详细信息

接口调用请求说明

https请求方式：GET
https://host:port/v2/memberlist?ledger=blkc2022

GET请求示例

https://host:port/v2/memberlist?ledger=blkc2022

返回说明

```
{  
  "state": 200,  
  "message": "success",  
  "data": {
```

```
    "memberList": [
      {
        "id":
"QmPsujuU6t6dzaERcFY063QkQmCifgv82EswcrGrwAdd6cE",
        "addr": "/ip4/10.1.3.151/tcp/60005",
        "port": 3105,
        "shownName": "peer151",
        "inAddr": "/ip4/10.1.3.151/tcp/60005",
        "height": 32,
        "tlsEnabled": 1,
        "hashType": "sm3",
        "consensus": "Raft"
      },
      {
        "id":
"QmdTpMMu6jZ2ATKxCYwx6pGRBak7UVkG7g29n4RR9QdrXT",
        "addr": "/ip4/10.1.3.152/tcp/60005",
        "state": 1,
        "port": 3105,
        "shownName": "peer152",
        "inAddr": "/ip4/10.1.3.152/tcp/60005",
        "height": 32,
        "tlsEnabled": 1,
        "hashType": "sm3",
        "consensus": "Raft"
      }
    ]
  }
}
```

返回结果参数说明

参数	说明
id	节点id
addr	节点ip
state	节点在线/离线，1表示离线，字段不显示或者为0表示在线
port	rpc端口

参数	说明
inaddr	内网地址
height	当前节点的区块高度
tlsEnabled	TLS是否开启，0开启，1关闭
hashType	哈希方式
consensus	共识算法

3.4、健康监测接口

接口调用请求说明

https请求方式：**GET**
https://host:port/health

GET请求示例

https://host:port/health

成功返回说明

返回**https**状态码 200

失败返回说明

返回**https**状态码 400

4、Smart Token功能接口

4.1、发行申请

数字资产发行申请。

接口调用请求说明

https请求方式： POST
https://host:port/v2/tx/stoken/issue/apply?ledger=blkc2022

请求示例

```
{
  "contractAddress": "059aef6c0b0c6640274e5d7f8aebbc2868195551982c7b93ac02b030a04701f1",
  "tokenType": "test1",
  "reissued": true,
  "activeDate": 1605515575000,
  "expireDate": 1615515575000,
  "maxLevel": 0,
  "tokenList": [
    {
      "address": "SoQHSJB4h7U4A5qKnnSroaittDY3hMpwenXZdt8wHyiPAwwMZm7",
      "subType": "1",
      "amount": "100"
    }
  ]
}
```

参数说明

参数	类型	是否必要参数	说明
contractAddress	String	yes	调用的合约的地址
tokenType	String	yes	资产类型
expireDate	Number	yes	数字资产有效期时间戳
tokenList	Object array	yes	数字资产接收信息
activeDate	Number	no	数字资产激活日期时间戳
reissued	Bool	no	是否支持增发

参数	类型	是否必要参数	说明
maxLevel	Number	no	最大流转层级，填0则 unlimited 流转层级
extend	String	no	用户合约参数

tokenList对象的结构

字段	类型	是否必要参数	说明
address	String	yes	受让地址
amount	String	yes	接收金额
subType	String	no	数字资产子类型

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": {
    "msg": {
      "contractAddress":
"200f3b89d65e1a6c175804052effce719b41582fb7504033feadc8300db4a768",
      "tokenType": "soloIntf-15i8Ai",
      "reissued": true,
      "activeDate": 0,
      "expireDate": 1607424290357,
      "maxLevel": 0,
      "tokenList": [
        {
          "address":
"3J79ha6NRjrUqViEDFcMVTHWL4BSqH4jYfADtimgzeHnQpQAfz",
          "amount": "10000.123456",
          "subType": "test"
        }
      ]
    }
  }
}
```

```
    }
  ],
  "extend": "",
},
"sigMsg":
"7b224f757470757473223a5b7b2256616c7565223a313030303031323334
35362c22546f6b656e54797065223a226332397362306c75644759744d545
6704f454670222c225075624b657948617368223a224d306f334f5768684e
6b3553616e4a5663565a70525552475930313256456858544452435533464
94e47705a5a6b464564476c745a33706c53473552634646425a6e6f3d222c
2253756254797065223a226447567a64413d3d222c224c6576656c223a312
c22457874223a6e756c6c7d5d2c2255736572436f6e7472616374223a2232
3030663362383964363565316136633137353830343035326566666365373
1396234313538326662373530343033336665616463383330306462346137
3638222c225265697373756564223a747275652c224163746976654461746
5223a302c2245787069726544617465223a31363037343234323930333537
2c224c6576656c223a302c22457874656e64223a22227d"
}
}
```

返回结果参数说明

参数	说明
msg	传入参数的原始数据
sigMsg	待签名数据(16进制格式)

4.2、发行审核

数字资产发行审核

接口调用请求说明

https请求方式：POST
<https://host:port/v2/tx/stoken/issue/approve?ledger=blkc2022>

请求示例


```
{
  "sigCrypt":
    "3045022046c25e52780e15c1adc591b12f3f7235af9b2478d37fee643f5c
    8962d20c316b022100923a4ceec6d88c44c4914feea6dc447b9af84f8fe0c
    c0fbae446e0429099bab4",
  "sigMsg":
    "7b226f757470757473223a5b7b2276616c7565223a313030303235303030
    302c22746f6b656e54797065223a2256454a664f484e51625442505931646
    b61413d3d222c227075624b657948617368223a224d306f334f5768684e6b
    3553616e4a5663565a7052555247593031325645685854445243553346494
    e47705a5a6b464564476c745a33706c53473552634646425a6e6f3d222c22
    73756254797065223a226447567a64413d3d222c226c6576656c223a312c2
    2657874223a6e756c6c7d5d2c2275736572636f6e7472616374223a223230
    3066336238396436356531613663313735383034303532656666636537313
    9623431353832666237353034303333666561646338333030646234613736
    38222c227265697373756564223a747275652c22616374697665446174652
    23a302c2265787069726544617465223a313630373531303237343637332c
    226c6576656c223a302c22657874656e64223a22227d",
  "pubKey":
    "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1l1IS29aSXpqMENB
    UVlJS29FY3oxVUJnaTBEUWdBRWQxdkt0aGhISmw3SWZFbHNWN3A5QmZVWEtYS
    WcNCnRhYm8zeEZyNU41d2x4VnRCbTAXVDEwQjlQSFYzOWthYXFNaS9XcURPM0
    1ucStDMmJ6bmpSa04wMGc9PQ0KLS0tLS1FTkQgUUVFVCTEldiEtFWS0tLS0tDQo
    ="
}
```

参数说明

参数	类型	是否必 要参数	说明
sigMsg	String	yes	签名原文
sigCrypt	String	yes	签名后的数据（对签名原文的byte进行hash后签名再转成16进制）
pubKey	String	yes	签名公钥（base64编码格式）

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
    "58c11e0c69d43c3da43ae45081a7a09f95e142c8420233c0043404bb07ce
    27c2"
}
```

返回结果参数说明

参数	说明
data	交易ID

4.3、转让申请

数字资产转让申请。

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/stoken/transfer/apply?ledger=blk2022

请求示例

```
{
  "extendData": "ADDRESS1 向 MULITADD4 转账10个TB_EswhX4",
  "collectionList": [
    {
      "amount": "10",
      "address":
        "SnjmmuuTyoc2WtpQ3BhUra7vyi1Js5yUPof2YQurz6PP4PQiJSE",;
      "subType": "test"
    }
  ],
  "tokenType": "TB_EswhX4",
  "paymentList": [
    {
      "amount": "10",
      "address":
        "3J79ha6NRjrUqViEDFCMVTHWL4BSqH4jYfADtimgzeHnQpQAfz",

```

```
      "subType": "test"
    }
  ]
}
```

参数说明

参数	类型	是否必要参数	说明
tokenType	String	yes	数字资产类型
paymentList	Object Array	yes	支出列表
collectionList	Object Array	yes	收入列表
newSubType	String	no	数字资产激活日期支出后找零的数字资产子类型，不填则不变。
extendArgs	String	no	账户合约转账参数
extendData	String	no	扩展数据

paymentList对象的结构

字段	类型	是否必要参数	说明
address	String	yes	转出地址
amount	String	yes	转出金额
subType	String	no	数字资产子类型

collectionList对象的结构

字段	类型	是否必要参数	说明
address	String	yes	受让地址
amount	String	yes	受让金额
subType	String	no	数字资产子类型

返回说明

正常返回结果:

[illegible]

```
    "sigMsg": [
      {
        "signMsg":
"3abbd1f2629a4759f434e0ba69458927c71ddd6d0e41d28e294ab21d781
4e8e00000002771f4ee3edcbc144a46774f29ccfc0ee24db9406ca151eb0
56880e195064e21",
        "address":
"3J79ha6NRjrUqViEDFcMVTHWL4BSqH4jYfADtimgzeHnQpQAfz"
      }
    ]
  }
}
```

返回结果参数说明

参数	说明
UTXOInfo	交易原始数据
msg	传入参数的原始数据
sigMsg	待签名数据列表，每个signMsg需要对应的address签名。

4.4、转换申请

数字资产转换申请。

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/stoken/exchange/apply?ledger=blkc2022

请求示例

```
{
  "TokenType": "test1",
  "PaymentList":
  [{ "Address": "SoQHSJB4h7U4A5qKnnSroai ttDY3hMpwenXZdt8wHyi PAwwM
Zm7", "SubType": "1", "Amount": "70" }],
  "NewTokenType": "test2",
  "NewSubType": "test2",
  "ExtendArgs": "" ,
  "ExtendData": "{ \"Address\":
\\ \"SoQHSJB4h7U4A5qKnnSroai ttDY3hMpwenXZdt8wHyi PAwwMZm7\\\", \\ \"Sub
Type\\\": \\ \"1\\\", \\ \"Amount\\\": \\ \"70\\\" }"
}
```

参数说明

参数	类型	是否必要参数	说明
TokenType	String	yes	数字资产类型
PaymentList	Object Array	yes	支出列表
NewSubType	String	no	数字资产激活日期支出后找零的数字资产子类型，不填则不变
NewTokenType	String	yes	转换后的数字资产类型
ExtendArgs	String	no	账户合约转账参数
ExtendData	String	no	UTXO扩展数据

PaymentList对象的结构

字段	类型	是否必要参数	说明
Address	String	yes	受让地址
SubType	String	no	数字资产子类型
Amount	String	yes	接收金额

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": {
    "UTXOInfo":
      "010000002000000058c11e0c69d43c3da43ae45081a7a09f95e142c84202
      33c0043404bb07ce27c20000000000000000000000000000000000000000
      000000001000000000000000000000000000000000000000000000000000
      00746573743133000000536f5148534a4234683755344135714b6e6e53726
      f61697474445933684d7077656e585a64743877487969504177774d5a6d37
      010000003100000000000000000000000000000000000000000000000000
      573743233000000536f366572695559526f4742544e6e687239724a776a75
      327058414c4c4b366336367745744257715558736d4c37737679326f01000
      000310000000000000000000000000000000000000000000000000000000
      22536f5148534a4234683755344135714b6e6e53726f61697474445933684
      d7077656e585a64743877487969504177774d5a6d37222c22537562547970
      65223a202231222c22416d6f756e74223a20223730227d00000000",
    "msg": {
      "TokenType": "test1",
      "PaymentList": [
        {
          "Address":
            "SoQHSJB4h7U4A5qKnnSroaittDY3hMpwenXZdt8wHyiPAwwMZm7",
          "Amount": "70",
          "SubType": "1"
        }
      ],
      "CollectionList": [
        {
          "Address":
            "So6eriUYRoGBTNnhr9rJwju2pXALLK6c66wEtBWqUXsmL7svy2o",
          "Amount": "70",
          "SubType": "1"
        }
      ],
      "NewTokenType": "test2",
      "NewSubType": ""
    }
  }
}
```

```
      "ExtendArgs": "{\\"Address\\":  
    \\"SoQHSJB4h7U4A5qKnnSroaittDY3hMpwenXZdt8wHyiPAwwMZm7\\",\\"Sub  
    Type\\": \\"1\\",\\"Amount\\": \\"70\\"}",  
      "ExtendData": ""  
    },  
    "sigMsg": [  
      {  
        "signMsg":  
        "58c11e0c69d43c3da43ae45081a7a09f95e142c8420233c0043404bb07ce  
        27c201000000af856351af2d7adde8c942b71730e848ccd747419d9989418  
        d94fd1952f8d83f",  
        "address":  
        "SoQHSJB4h7U4A5qKnnSroaittDY3hMpwenXZdt8wHyiPAwwMZm7"  
      }  
    ]  
  }  
}
```

返回结果参数说明

参数	说明
UTXOInfo	未加入签名的交易数据
msg	传入参数的原始数据
sigMsg	待签名数据列表,每个signMsg找对应的address签名

4.5、销毁（回收）申请

数字资产销毁（回收）申请

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/stoken/destroy/apply?ledger=blkc2022

请求示例


```
{
  "extendData": "销毁 ADDRESS1 中的数字资产",
  "tokenType": "TB_hK8x9Z3Ao0",
  "extendArgs": "",
  "paymentList": [
    {
      "amount": "990.25",
      "address":
        "SnQ832BvBDt51FFHauFaXDzkLS7j7nbbM6oh3GDBNRnsnNJmDBb",
      "subType": "test"
    }
  ]
}
```

参数说明

参数	类型	是否必要参数	说明
tokenType	string	yes	数字资产类型
paymentList	Object Array	yes	支出列表
extendArgs	string	no	账户合约销毁参数
extendData	string	no	扩展信息

paymentList对象的结构

字段	类型	是否必要参数	说明
address	String	yes	需要销毁资产的地址
amount	String	yes	接收金额
subType	String	no	数字资产子类型

返回说明

正常返回结果:

```
{
```

```

"state": 200,
"message": "success",
"data": {
  "UTXOInfo":
"01000000200000006fbaf5b3bf59121a7e2a46b83835a3e7b37b26d2bbd5
44d5d3ad2b4dd8c9fe8d0000000000000000000000000000000000000000
000000001000000000000000000000000000000000000000000000000000
0054425f684b3878395a33416f301a0000006f73456f793933334c6b48797
94263676a4537764369766673580400000074657374000000000000000000
000000040000005b22225d23000000e99480e6af812041444452455353312
0e4b8ade79a8454425f684b3878395a33416f30",
  "msg": {
    "tokenType": "TB_hk8x9Z3Ao0",
    "paymentList": [
      {
        "address":
"SnQ832BvBDt51FFHauFaXDzkLS7j7nbbM6oh3GDBNRnsnNJmDBb",
        "amount": "990.25",
        "subType": "test"
      }
    ],
    "extendArgs": "",
    "extendData": "销毁 ADDRESS1 中的数字资产"
  },
  "sigMsg": [
    {
      "signMsg":
"6fbaf5b3bf59121a7e2a46b83835a3e7b37b26d2bbd544d5d3ad2b4dd8c9
fe8d01000000b6864c551872d2c60d95067a6359a15cfa61e2b552dbfe497
9dddb96827703c7",
      "address":
"SnQ832BvBDt51FFHauFaXDzkLS7j7nbbM6oh3GDBNRnsnNJmDBb"
    }
  ]
}
}

```

返回结果参数说明

参数	说明
----	----


```
    "pubkeyList": [

      "LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0Nck1Ga3dFd1l1IS29aSxpqMEN
      BUVlJS29FY3oxVUJnaTBEUwdBRWQxdkt0aGhISmw3SWZFbHNWN3A5QmZVWEtY
      SWcNCnRhYm8zeEZyNU41d2x4VnRCbTAXVDEwQj1QSFYzOWthYXFNaS9XcURPM
      01ucStDMmJ6bmpSa04wMGc9PQ0KLS0tLS1FTkQgUUFVCTE1DIETfWS0tLS0tDQ
      o="

    ]
  }
]
```

参数说明

参数	类型	是否必要 参数	说明
type	String	yes	审核类型，可选值：transfer，exchange，destroy
payAddressInfoList	Object array	yes	转让方信息列表
UTXOInfo	String	yes	交易原始数据

PayAddressInfoList对象的结构

字段	类型	是否必 要参数	说明
address	String	yes	签名地址
pubkeyList	Array	yes	签名地址的所有公钥
signList	Array	yes	每个公钥签名后的数据，需要与pubkeyList的公钥位置一一对应

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
    "574eca245b6079191e82b7475c9bc2b6712d71286790d8629fbf797f344f41e7"
}
```

返回结果参数说明

参数	说明
data	交易Id

4.7、创建账户地址

账户地址

- 由一个或多个公钥生成，公钥数量不超过10，一般最多3个。
- 可设置最少签名数量M
- 转出资产时，有效签名数量 $\geq$ M即可

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/stoken/address/generate?
ledger=blk2022

请求示例

```
{
  "minSignatures": 2,
  "pubkeys": [

    "LS0tLS1CRUdJTiBFQyBQVUJMSUMgS0VZLS0tLS0KTUZrd0V3WUhLb1pJemowQ0FRWU1Lb1pJemowREFRY0RRZ0FFNmteCjVldFBSSmN1MmdSe1ZUcj3NnM2N1VDbgpObwxFV1gzUjFPOWQwzzZFUU96SXA3aGR1OV14VE1aexdkNUNzdjM0N0F0R2xqQzh1aWNBZXBjVmR3PT0KLS0tLS1FTkQgRUMgUFVCTE1DIetFWS0tLS0tCg==",

    "LS0tLS1CRUdJTiBFQyBQVUJMSUMgS0VZLS0tLS0KTUZrd0V3WUhLb1pJemowQ0FRWU1Lb1pJemowREFRY0RRZ0FFU31RZnhudHp1N1JwV2g2ejRaVTJmdCszTz13dApva1EwWHg5T11KcdhpckFLUjJyREFRCVcxZER2c2ZBYUN3Uitubn1IcGpCSmduakp5QTdQeFQ4N0hBPT0KLS0tLS1FTkQgRUMgUFVCTE1DIetFWS0tLS0tCg=="
  ]
}
```

参数说明

参数	类型	是否必要参数	说明
pubkeys	string array	yes	公钥数组
minSignatures	number	yes	需要签名的个数

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": {
    "SqKzag9JtcMoebESithCabP8Ea2facSKKC3FuqRASH8aZcQo457"
  }
}
```

返回结果参数说明

参数	说明
data	账户地址

4.8、 根据账户地址查询余量

根据账户地址查询数字资产余量。

接口调用请求说明

https请求方式：GET

https://host:port/v2/tx/stoken/balance?ledger=b1kc2022

请求示例

查询账户地址下所有资产类型的余量：

**https://host:port/v2/tx/stoken/balance?
address=SqKzag9JtcMoebESiTHcabP8Ea2facSKKC3FuqRASH8aZcQo457?
ledger=b1kc2022**

查询账户地址下指定资产类型的余量：

**https://host:port/v2/tx/stoken/balance?
address=SqKzag9JtcMoebESiTHcabP8Ea2facSKKC3FuqRASH8aZcQo457&t
okentype=type1?ledger=b1kc2022**

查询回收账户下所有资产类型的余量：

**https://host:port/v2/tx/stoken/balance?
address=osEoy933LkHyyBcgjE7vCivfsX?ledger=b1kc2022**

参数说明

参数	类型	必须参数	说明
address	String	yes	账户地址
tokentype	String	no	资产类型

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data": {
    "type1": [
      {
        "subType": "1",
        "amount": "100",
        "confirmed": true
      }
    ]
  }
}
```

返回结果参数说明

参数	说明
data	资产结构的集合，根据资产类型分类
type1	数字资产类型
subType	数字资产子类型
amount	数字资产余量
confirmed	数字资产上链确认标识

4.9、冻结

数字资产冻结

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/stoken/freeze?ledger=blkc2022

请求示例


```
{
  "tokenType": "test1",
  "comments": ""
}
```

参数说明

参数	类型	必须参数	说明
tokenType	String	yes	数字资产类型
comments	String	no	备注

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
  "a108bdd5c2eb90a2470269a40edbb42d3c08b78fd60be3c685dac2203a7c4df2"
}
```

返回结果参数说明

参数	说明
data	交易Id

4.10、解冻

数字资产解冻

接口调用请求说明

https请求方式：POST
https://host:port/v2/tx/stoken/recover?ledger=blkc2022

请求示例

```
{
  "tokentype": "test1",
  "comments": ""
}
```

参数说明

参数	类型	必须参数	说明
tokenType	String	yes	数字资产类型
comments	String	no	备注

返回说明

正常返回结果：

```
{
  "state": 200,
  "message": "success",
  "data":
    "a108bdd5c2eb90a2470269a40edbb42d3c08b78fd60be3c685dac2203a7c4df2"
}
```

返回结果参数说明

参数	说明
data	交易Id

5、事件通知

事件模块是为了使客户端和梧桐链之间的交互更加的自由灵活而设计开发的。它可以使梧桐链的运行状况更加快速的反馈到外部业务系统，从而使梧桐链产品可以支持更多的业务场景。

目前事件模块在梧桐链中有两个地方用到了。其一是区块上链时发送交易的上链事件，其二是合约中开放了一个内置函数，调用合约时，内置函数也会触发合约事件，合约事件由用户自定义。

5.1、事件模块的使用

使用事件模块主要就是配置好事件中心和订阅方的信息。其中节点为事件中心，SDK为订阅方。并不需要所有节点都开启事件模块，只需要开启一个节点的事件模块，就可以使用SDK订阅事件模块的事件。

SDK中需要配置一个外部处理事件的服务，这个服务必须实现一个接口，该接口接受 https POST 请求，content-type 为application/json， 处理成功后返回字符串“success”。并且需要将这个接口的 url 填入SDK配置文件中的EventHandler。

接口接收的json参数如下

参数	数据类型	说明
Topic	String	事件主题
SubTopic	String	事件子主题
Date	String	事件发生时间(UTC时间)，格式（YYYY-MM-DD hh:mm:ss）
LedgerID	String	触发事件的链
BlockHeight	number	触发事件的区块
TxList	Object Array	事件包含的交易信息
Client	String	事件发起方——sdk ID
EventMsg	Object	事件消息

TxList对象的结构

字段	类型	说明
TxHash	String	交易hash
TxIndex	Number	交易在区块中的索引

Topic事件类型定义

字段	类型	说明
txevent	String	交易事件
scevent	String	合约事件

SubTopic事件类型定义

字段	类型	说明
onchain	String	上链交易事件
drop	String	丢弃交易事件

5.2、事件模块中Topic通配符

5.2.1 主题层级分隔符—“/”

主题层级分隔符使得主题名结构化。如果存在分隔符，它将主题名分割为多个主题层级。斜杠（‘/’ U+002F）用于分割主题的每个层级，为主题名提供一个分层结构。当客户端订阅指定的主题过滤器包含两种通配符时，主题层级分隔符就很有用了。主题层级分隔符可以出现在主题过滤器或主题名字的任何位置。相邻的主题层次分隔符表示一个零长度的主题层级。

如主题：

```
room212/electric  
room212/tv/contrl/sensor  
room212/tv/contrl/lightroom212/air/sensor
```

5.2.2 多层通配符—“#”

“#”是用于匹配主题中任意层级的通配符。多层通配符表示它的父级和任意数量的子层级。多层通配符必须位于它自己的层级或者跟在主题层级分隔符后面。不管哪种情况，它都必须是主题过滤器的最后一个字符。

例如，如果客户端订阅主题 “china/xiangtan/#”，它会收到使用下列主题名发布的消息：

```
china/xiangtan
china/xiangtan/yuhu
china/xiangtan/yuetan/hnie
china/xiangtan/jiuhua/jiakao/kemusan
```

订阅主题示例

school/#//也匹配单独的 “school” ，因为 # 包括它的父级。#//是有效的，会收到所有的应用消息。school/teacher/#//有效的。school/teacher#//无效的。school/teacher/#/lever//无效的,必须是主题过滤器的最后一个字符

5.2.3 单层通配符—“+”

加号是只能用于单个主题层级匹配的通配符。在主题过滤器的任意层级都可以使用单层通配符，包括第一个和最后一个层级。然而它必须占据过滤器的整个层级。可以在主题过滤器中的多个层级中使用它，也可以和多层通配符一起使用。

china/+ 只能匹配：

```
china/guangzhou
```

china/+/+/zhongshanlu 能匹配：

```
china/guangzhou/tianhe/zhongshanlu
```

```
china/shenzhen/nanshan/zhongshanlu
```

5.3、 根据区块高度查询事件

根据区块高度返回该高度对应的所有事件。

接口调用请求说明

```
https请求方式：GET
https://host:port/v2/event/msg/{height}?ledger=blkc2022
```

请求示例

```
https://10.1.3.160:8080/v2/event/msg/1?ledger=blkc2022
```

请求参数说明

参数	类型	是否必要参数	说明
height	Number	yes	区块高度

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": [
    {
      "id": 0,
      "topic": "txevent",
      "subTopic": "onchain",
      "date": "2022-06-30 14:28:01",
      "ledgerID": "blkc2022",
      "blockHeight": 1,
      "txList": [
        {
          "txIndex": 0,
          "txId":
            "2658d84c9fd2c596cb3cf29b277006ae5eebb66ce23f69ea4eeafc56ed3fb5c4"
        },
        {
          "txIndex": 1,
          "txId":
            "57da74b4c273b280362b946d3237e55ae1411ed15fa4fab9df4a9629067ef998"
        }
      ],
      "client":
        "a956549c905eb0f5b782a89f5e99bbfdf597248ef05c51334efc8cdb92cd0a71104e1d070b1a57820c010f7749ca6f389c87d7dcae024fdc9e22f15e3d440b70",
      "eventMsg": "SUCCESS"
    }
  ]
}
```

```
}
```

返回结果参数说明

参数	说明
id	事件在区块中的索引
topic	事件的主题 ， 参考5.1节定义
subTopic	事件的子主题， 参考5.1节定义
date	事件的日期
ledgerID	应用链id
blockHeight	区块的高度
eventMsg	事件的内容
client	交易发起方的ID
txList	触发事件的交易列表
txList.txIndex	交易在区块中的索引
txList.txId	交易ID

错误返回结果示例(没有找到高度为100的事件信息):

```
{
  "state": 404,
  "message": " _error:rpc error: code = Unknown desc = event not found",
  "data": null
}
```

5.4、 根据交易ID查询事件

根据交易ID查询该交易相关的所有事件。

接口调用请求说明

https请求方式：GET
https://host:port/v2/event/msg/tx/{txId}?ledger=blkc2022

请求示例

```
https://10.1.3.160:8080/v2/event/msg/tx/9a2cd06805c96108f2244e8aec54d61aa1ff334c0b9bb621bfa12784e0a328f7?ledger=blkc2022
```

请求参数说明

参数	类型	是否必要参数	说明
txId	String	yes	交易ID(十六进制编码格式)

返回说明

正常返回结果:

```
{
  "state": 200,
  "message": "success",
  "data": [
    {
      "id": 0,
      "topic": "txevent",
      "subTopic": "onchain",
      "date": "2022-06-30 14:28:01",
      "ledgerID": "blkc2022",
      "blockHeight": 1,
      "txList": [
        {
          "txIndex": 0,
          "txId":
            "2658d84c9fd2c596cb3cf29b277006ae5eebb66ce23f69ea4eeafc56ed3fb5c4"
        },
        {
          "txIndex": 1,
          "txId":
            "57da74b4c273b280362b946d3237e55ae1411ed15fa4fab9df4a9629067ef998"
        }
      ]
    }
  ],
}
```



```
    "client":
      "a956549c905eb0f5b782a89f5e99bbfdf597248ef05c51334efc8cdb92cd
      0a71104e1d070b1a57820c010f7749ca6f389c87d7dcae024fdc9e22f15e3
      d440b70",
      "eventMsg": "SUCCESS"
    }
  ]
}
```

错误返回结果示例(没有找到对应交易ID的事件信息):

```
{
  "state": 404,
  "message": " _error:rpc error: code = Unknown desc =
event not found",
  "data": null
}
```

附录

WVM合约代码示例

```
//合约示例
contract TestExample2 {
    public string init(){
        return "success"
    }

    //初始化一个账户
    public string set(string account,int amount){
        print(account)
        print(amount)
        db_set(account,amount)___
        return "success"
    }

    //转账操作
```

```

    public string transfer(string from, string to, int
amount) {
        int balA = db_get<int>(from)
        int balB = db_get<int>(to)
        balA = balA-amount
        if (balA>0){
            balB = balB+amount
            db_set(from, balA)
            db_set(to, balB)
        }else{
            return "failed"
        }
        return "success"
    }

    //查询账户余额
    public int getBlance(string account){
        print(account)
        int a = db_get<int>(account)
        return a
    }

}

```

事件模块外部服务Go语言代码示例

```

package main

import (
    "encoding/json"
    "fmt"
    "net/http"

    "github.com/gin-gonic/gin"
)

type Event struct {
    Topic      string    `json:topic,omitempty`
    SubTopic   string    `json:subTopic,omitempty`
    Date       string    `json:date,omitempty` //事件发生时间
}

```

```

    LedgerID    string    `json:ledgerID,omitempty`    //触
发事件的链
    BlockHeight uint64     `json:blockHeight,omitempty` //触
发事件的区块
    TxList      []TxInfo   `json:txList,omitempty`    //交
易信息
    Client      string     `json:client,omitempty`    //事
件发起方--sdk ID
    EventMsg     interface{} `json:eventMsg,omitempty`  //事
件消息
}
type TxInfo struct {
    TxHash []byte //触发事件的交易
    TxIndex uint   //触发事件的交易索引
}

func main() {
    RunWeb("9300")
}

func RunWeb(port string) {
    gin.SetMode(gin.ReleaseMode)
    router := gin.Default()
    router.POST("/event", HandlerEvent)
    fmt.Println(fmt.Sprintf("服务启动...端口为%v...", port))
    router.Run(":" + port)
}

func HandlerEvent(c *gin.Context) {
    var req Event
    err := c.BindJSON(&req)
    if err != nil {
        c.String(http.StatusBadRequest, fmt.Sprintf("err :
%s", err))
        return
    }
    tmp, _ := json.Marshal(req)
    fmt.Printf("%s\n", string(tmp))
    c.String(http.StatusOK, "SUCCESS")
}

```

