

# 安网 3V2.0 应用协议

## V1.0

1. 规则.....	3
2. 应用接口.....	4
2.1. 错误码.....	4
2.2. 注册应用.....	5
2.3. 设置权限.....	6
2.4. 创建外带数据交易.....	7
2.5. 根据交易 ID 返回外带数据.....	8
2.6. 按 AppID 返回交易列表.....	8
2.7. 按钱包地址+AppID 返回交易列表.....	9
2.8. 获取应用信息.....	10
2.9. 获取所有应用列表.....	11
2.10. 按钱包地址查询应用列表.....	11
2.11. 获取应用详情.....	12
2.12. 获取应用的权限列表.....	13
3. 安资接口.....	13
3.1. 错误码.....	13
3.2. 发行.....	14
3.3. 追加发行.....	16
3.4. 转账.....	16
3.5. 销毁.....	17
3.6. 发放糖果.....	18
3.7. 领取糖果.....	18
3.8. 获取钱包可领取糖果列表.....	19
3.9. 查询资产信息.....	19
3.10. 按资产 ID 查询交易列表.....	20
3.11. 按钱包地址+资产 ID 查询交易列表.....	21
3.12. 按钱包地址+资产 ID 查询余额.....	22
3.13. 按交易 ID 查询安资详情.....	22
3.13.1. 发行.....	23
3.13.2. 追加发行.....	23
3.13.3. 转账.....	23
3.13.4. 销毁.....	24
3.13.5. 糖果发放.....	24
3.13.6. 糖果领取.....	24
3.14. 获取所有资产列表.....	24

---

3.15. 根据 safe 地址返回资产列表.....	25
3.16. 获取指定高度和地址对应的 SAFE 金额.....	25
3.17. 获取所有发放糖果对应的区块高度列表.....	26
3.18. 获取地址领取的糖果记录.....	26
3.19. 查询本地资产信息.....	27
3.20. 查询本地资产列表.....	27
4. 公共接口.....	28
4.1. 获取交易信息.....	28

## 1. 规则

### 1.1. 调用 RPC

RPC 远程调用是基于 HTTP 协议完成的，数据依托于 JSON 数据。

假如在节点服务器上配置的 RPC 用户名和密码分别是 “rpcuser” 和 “rpcpass”，远程 RPC 接口调用需要认证，认证采用的是 Base64 编码方式，具体格式是 “Basic “ + base64(“RPC 用户名:RPC 密码”)，将 rpcuser 和 rpcpass 代入，得到的结果也就是 “Basic cnBjdXNlcjpycGNwYXNz”。

JSON 数据格式如下：

```
{
  "jsonrpc": "1.0",
  "id": "test",
  "method": "registerapp",
  "params":
  [
    参数 1,
    参数 2,
    ...
  ]
}
```

其中 params 字段参数值，不需要带上 Key，只放入 Value 值就行。例如调用 registerapp 接口，它的 params 填充如下：

```
"params":
[
  “我的游戏”,
  “我的游戏，XX 公司出品”,
  1,
  “XX 公司”,
  “http://www.abc.com”,
  “http://www.abc.com/applog.png”,
  “http://www.abc.com/appcover.png”
]
```

]

## 2. 应用接口

### 2.1. 错误码

错误码	错误原因
-500	不存在的应用 ID
-501	已存在的应用 ID
-502	公钥地址上 SAFE 金额不足
-503	无效的消耗 SAFE 金额
-504	无效的应用名称
-505	已经存在的应用名称
-506	无效的应用描述
-507	无效的开发者类型
-508	无效的开发者名称
-509	无效的官网 URL
-510	无效的应用 Logo URL
-511	无效的应用封面 URL
-512	获取管理员地址失败
-513	无效的权限设置类型
-514	无效的公钥地址
-515	您的钱包不包含该公钥地址
-516	无效的应用内命令
-517	重复设置应用内命令
-518	已存在的应用内命令
-519	不存在的应用内命令
-520	您不是管理员
-521	公钥地址没有操作权限
-522	无效的应用交易类型
-523	无效的外带数据
-524	获取外带数据失败

-525	获取相应的交易 ID 失败
-526	获取应用信息失败
-527	正在同步区块, 无法创建交易

## 2.2. 注册应用

方法:

registerapp

请求:

```
{
  "appName": "",           // 应用名称
  "appDesc": "",          // 应用描述
  "devType":,              // 开发者类型
  "devName": "",          // 开发者名称 (公司名称、个人姓名)
  "webUrl": "",           // 官网 URL
  "appLogoUrl": "",       // 应用 Logo URL
  "appCoverUrl": ""      // 应用封面 URL
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "adminSafeAddress": "", // 管理员公钥地址
    "appId": "",           // 应用 ID
    "txId": ""             // 交易 ID
  }
}
```

字段说明:

字段名	字段名称	字段描述
appName	应用名称	最大 50 个字节
appDesc	应用描述	最大 600 个字节
devType	开发者类型	1--公司 2--个人
devName	开发者名称	最大 100 个字节, 如果是公司, 填写公司名称 如果是个人, 填写个人姓名
webUrl	官网 URL	最大 300 个字节
appLogoUrl	应用 Logo URL	最大 300 个字节

appCoverUrl	应用封面 URL	最大 300 个字节
adminSafeAddress	管理员公钥地址	
appld	应用 ID	从 1001 开始累加

### 2.3. 设置权限

方法:

setappauth

请求:

```
{
  "setType": "",           // 设置类型
  "appld": "",           // 应用 ID
  "userSafeAddress": "", // 公钥地址
  "appAuthCmd":          // 应用内的命令
}
```

返回:

```
{
  "error": "",
  "id": "",
  "result":
  {
    "txId": ""           // 交易 ID
  }
}
```

字段说明:

字段名	字段名称	字段描述
setType	设置类型	1--添加 2--删除
appld	应用 ID	
userSafeAddress	公钥地址	
appAuthCmd	应用内的命令	由开发者自己定义,只能是数字,范围在 1000--4294967296

一些快捷设置权限方法, 如下:

公钥地址	应用内的命令	描述
ALL_USER	0	设置所有用户不能操作应用内的所有命令
ALL_USER	1	设置所有用户可以操作应用

		内的所有命令
Xxft275SXtRNqqge6Em5a5Fp mhjydUBtuz	0	公钥地址对应的用户不能操作应用内的所有命令
XpwZ5ojkMPFr42AKofDB8RhA Bddis6U4cj	1	公钥地址对应的用户可以操作应用内的所有命令
ALL_USER	某个应用命令	设置所有用户可以操作某个应用命令

权限优先级如下：

- 1) 优先级 1：（ALL\_USER， 0），（ALL\_USER， 1），这两种权限只能存在一个，后面的设置将覆盖上一个；
- 2) 优先级 2：（ADDRESS1， 0），（ADDRESS1， 1），这两种权限只能存在一个，后面的设置将覆盖上一个；
- 3) 优先级 3：（ALL\_USER， 某个应用命令），所有用户可以操作某个应用命令；
- 4) 优先级 4：（ADDRESS2， 某个应用命令），某个用户只能操作某个应用命令；

#### 2.4. 创建外带数据交易

方法：

createextenddatatx

请求：

```
{
  "appTxType":,           // 应用交易类型
  "userSafeAddress": "", // 用户公钥地址
  "appId": "",           // 应用 ID
  "appAuthCmd":,        // 应用内的命令
  "extendData": ""      // 外带数据
}
```

返回：

```
{
  "error":,
  "id": "",
  "result":
  {
    "txId": ""           // 交易 ID
  }
}
```

字段说明:

字段名	字段名称	字段描述
appTxType	应用交易类型	1--正常交易 2--即时交易
userSafeAddress	用户公钥地址	
appId	应用 ID	
appAuthCmd	应用内的命令	如果为 0
extendData	外带数据	最大不超过 2986 个字节

## 2.5. 根据交易 ID 返回外带数据

方法:

getextenddata

请求:

```
{
  "txId":""           // 交易 ID
}
```

返回:

```
{
  "error":,
  "id":,
  "result":
  {
    "extendDataList":
    [
      {
        "appId":,      // 应用 ID
        "appData":    // 应用数据
      },
      ...
    ]
  }
}
```

## 2.6. 按 AppID 返回交易列表

方法:

getapptxids

请求:

```
{
  "appId":""         // 应用 ID
  "appOperType":    // 应用操作类型
}
```



```

    "setType":           // 设置权限类型
  }

```

返回:

```

{
  "error":,
  "id": "",
  "result":
  {
    "txList":
    [
      "txId"           // 交易 ID
      ...
    ]
  }
}

```

字段说明:

字段名	字段名称	字段描述
appId	应用 ID	
appOperType	应用操作类型	1--所有的 2--注册应用交易 3--设置权限交易 4--创建外带数据交易
setType	设置权限类型	当 appOperType=3 时有效 1--添加 2--删除

## 2.7. 按钱包地址+AppID 返回交易列表

方法:

getaddressapptxids

请求:

```

{
  "safeAddress": "",   // 钱包地址
  "appId": "",        // 应用 ID
  "appOperType":     // 应用操作类型
  "setType":         // 设置权限类型
}

```

返回:

```

{
  "error":,
  "id": "",

```

```

    "result":
    {
        "txList":
        [
            "txId"          // 交易 ID
            ...
        ]
    }
}

```

字段说明:

字段名	字段名称	字段描述
safeAddress	钱包地址	
appId	应用 ID	
appOperType	应用操作类型	1--所有的 2--注册应用交易 3--设置权限交易 4--创建外带数据交易
setType	设置权限类型	当 appOperType=3 时有效 3--添加 4--删除

## 2.8. 获取应用信息

方法:

getappinfo

请求:

```

{
    "appId":""           //应用 ID
}

```

返回

```

{
    "error":,
    "id":,
    "result":
    {
        "appName":""    // 应用名称
        "appDesc":""    // 应用描述
        "devType":      // 开发者类型
        "devName":""    // 开发者名称 (公司名称、个人姓名)
    }
}

```

```
        "webUrl":""           // 官网 URL
        "appLogoUrl":""       // 应用 Logo URL
        "appCoverUrl":""     // 应用封面 URL
        "adminSafeAddress":"" // 管理员公钥地址
    }
}
```

## 2.9. 获取所有应用列表

方法:

getapplist

请求参数: 无

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "appList":
    [
      "appld" //应用 ID
      ...
    ]
  }
}
```

## 2.10. 按钱包地址查询应用列表

方法:

getapplistbyaddress

请求:

```
{
  "safeAddress":"" // safe 地址
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    [
      "appld" //应用 ID
    ]
  }
}
```

```
        ....  
    ]  
  }  
}  
}
```

## 2.11. 获取应用详情

方法:

getappdetails

请求:

```
{  
  "txId":"" //交易 id  
}
```

返回:

```
{  
  "error":,  
  "id":,  
  "result":  
  
  (1) 交易是注册应用类型  
  {  
    "appName":," // 应用名称  
    "appDesc":," // 应用描述  
    "devType":," // 开发者类型  
    "devName":," // 开发者名称 (公司名称、个人姓名)  
    "webUrl":," // 官网 URL  
    "appLogoUrl":," // 应用 Logo URL  
    "appCoverUrl":," // 应用封面 URL  
    "adminSafeAddress":," // 管理员地址  
  }  
  
  (2) 交易是设置权限类型  
  {  
    "setType":," // 设置类型  
    "appId":," // 应用 ID  
    "userSafeAddress":," // 用户地址  
    "appAuthCmd":," // 应用内的命令  
  }  
  
  (3) 交易是创建外带数据类型  
  {  
    "userSafeAddress":," // 用户地址  
    "appId":," // 应用 ID  
    "appAuthCmd":," // 应用内的命令  
    "extendData":," // 外带数据
```

```
    }  
  }
```

## 2.12. 获取应用的权限列表

方法:

getauthlist

请求:

```
{  
  "appld":""           // 应用 ID  
  "address":""        //可以是 ALL_USER 或者 safe 地址  
}
```

返回:

```
{  
  "error":,  
  "id":",  
  "result":  
  (1) 如果 address 是 ALL_USER:  
  {  
    "ALL_USER":  
    [  
      auth1,  
      .....  
    ]  
  }  
  (2) 如果 address 是 safe 地址  
  {  
    "ALL_USER":  
    [  
      auth1,  
      .....  
    ],  
    "CURRENT_USER":  
    [  
      auth1,  
      .....  
    ]  
  }  
}
```

## 3. 安资接口

### 3.1. 错误码

错误码	错误原因
-600	无效的资产简称
-601	已存在的资产简称
-602	无效的资产名称
-603	已存在的资产名称
-604	无效的资产描述
-605	无效的资产单位
-606	无效的资产总量
-607	无效的初次发行数量
-608	初次发行数量超过资产总量
-609	无效的初次实际发行数量
-610	无效的资产小数位
-611	无效的糖果数量
-612	糖果数量超过初次发行数量
-613	无效的糖果过期时间
-614	无效的备注
-615	已存在的资产 ID
-616	不存在的资产 ID
-617	无效的资产数量
-618	超出资产总量
-619	禁止资产销毁
-620	资产数量不足
-621	无效的锁定月份数
-622	不存在的资产糖果
-623	获取所有地址的 SFAE 金额失败
-624	无效的所有地址 SAFE 金额
-625	获取资产信息失败

### 3.2. 发行

方法:

issueasset

请求:

```
{
  "assetShortName": "", // 资产简称
  "assetName": "", // 资产名称
  "assetDesc": "", // 资产描述
  "assetUnit": "", // 资产单位
  "assetTotalAmount":, // 资产总量
  "firstIssueAmount":, // 初次发行总量
  "assetDecimals":, // 小数点
  "isDestory":, // 是否可以销毁
  "isPayCandy":, // 是否分发糖果
  "assetCandyAmount":, // 糖果数量
  "candyExpired":, // 糖果过期时间
  "remarks": "" // 备注
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "assetId": "", // 资产 ID
    "txId": "" // 交易 ID
  }
}
```

字段说明:

字段名	字段名称	字段描述
assetShortName	资产简称	最大 20 个字节
assetName	资产名称	最大 20 个字节
assetDesc	资产描述	最大 300 个字节
assetUnit	资产单位	最大 10 个字节
assetTotalAmount	资产总量	资产总量, 最大不超过 2000000000000000
firstIssueAmount	初次发行总量	不能小于等于 0, 也不能大于资产总量
assetDecimals	小数点	最小 4 位, 最大 10 位, 例如: 100000000, 代表是 10 的负 8 次方

isDestory	是否可以销毁	false--不可销毁 true--可以销毁
isPayCandy	是否分发糖果	false--不分发 true--分发
assetCandyAmount	资产糖果总额	
candyExpired	糖果过期时间	最小 1 个月，最大 6 个月
remarks	备注	最大 500 个字节
assetId	资产 ID	格式、长度同钱包地址一样

### 3.3. 追加发行

方法:

addissueasset

请求:

```
{
  "assetId": "",           // 资产 ID
  "addAmount": "",        // 追加数量
  "remarks": ""           // 备注
}
```

返回:

```
{
  "error": "",
  "id": "",
  "result": {
    "txId": ""           // 交易 ID
  }
}
```

字段说明:

字段名	字段名称	字段描述
assetId	资产 ID	
addAmount	追加数量	
remarks	备注	最大 500 个字节

### 3.4. 转账

方法:

transferasset

请求:

```
{
  "safeAddress": "",      // 收款地址
}
```



```

    "assetId": "",           // 资产 ID
    "assetAmount": "",      // 资产数量
    "lockTime": "",        // 锁定时长
    "remarks": ""          // 备注
}

```

返回:

```

{
  "error": "",
  "id": "",
  "result": {
    "txId": ""           // 交易 ID
  }
}

```

字段说明:

字段名	字段名称	字段描述
safeAddress	收款地址	
assetId	资产 ID	
assetAmount	资产数量	
lockTime	锁定时长	最小 1 个月, 最大 120 个月
remarks	备注	最大 500 个字节

### 3.5. 销毁

方法:

destoryasset

请求:

```

{
  "assetId": "",           // 资产 ID
  "assetAmount": "",      // 资产数量
  "remarks": ""          // 备注
}

```

返回:

```

{
  "error": "",
  "id": "",
  "result": {
    "txId": ""           // 交易 ID
  }
}

```

```
}
```

字段说明：

字段名	字段名称	字段描述
assetId	资产 ID	
assetAmount	资产数量	
remarks	备注	最大 500 个字节

### 3.6. 发放糖果

方法：

putcandy

请求：

```
{
  "assetId": "",           // 资产 ID
  "assetCandyAmount": ,   // 糖果数量
  "candyExpired": ,      // 糖果过期时间
  "remarks": ""          // 备注
}
```

返回：

```
{
  "error": ,
  "id": "",
  "result":
  {
    "txId": ""           // 交易 ID
  }
}
```

字段说明：

字段名	字段名称	字段描述
assetId	资产 ID	
assetCandyAmount	资产糖果总额	
candyExpired	糖果过期时间	最小 1 个月，最大 6 个月
assetAmount	资产数量	
remarks	备注	最大 500 个字节

### 3.7. 领取糖果

方法：

getcandy

请求:

```
{
  "assetId":"" // 资产 ID
}
```

返回:

```
{
  "error":,
  "id":,
  "result":
  {
    "txId":, // 交易 ID
    "assetAmount": // 领取到的资产数量
  }
}
```

### 3.8. 获取钱包可领取糖果列表

方法:

getavailablecandylist

请求:

无

返回:

```
{
  "error":,
  "id":,
  "result":
  {
    "candyList":
    [
      {
        "putTime": // 发放时间
        "assetId":, // 资产 ID
        "assetCandyAmount":, // 糖果数量
        "candyExpired": // 过期时间
      }
      ,
      ...
    ]
  }
}
```

### 3.9. 查询资产信息

方法:

getassetinfo

请求:

```
{
  "assetId":"" // 资产 ID
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "assetShortName": "", // 资产简称
    "assetName": "", // 资产名称
    "assetDesc": "", // 资产描述
    "assetUnit": "", // 资产单位
    "assetTotalAmount": "", // 资产总量
    "firstIssueAmount": "", // 初次发行总量
    "firstActualAmount": "", // 初次实际发行数量
    "alreadyIssueAmount": "", // 已经发行数量
    "assetDecimals":, // 最小单位
    "isDestory": // 是否可以销毁
    "isPayCandy": // 是否发放糖果
    "candyTotalAmount": "", // 糖果总量
    "destoryTotalAmout": "", // 销毁资产总量
    "candyExpired":, // 糖果过期时间
    "remarks": "", // 备注
    "issueTime":, // 发行时间
    "adminSafeAddress": "" // 管理员地址
  }
}
```

### 3.10.按资产 ID 查询交易列表

方法:

getassetidtxids

请求:

```
{
  "assetId": "", // 资产 ID
  "txClass": // 交易分类
}
```

返回:

```
{
```

```

    "error":,
    "id": "",
    "result":
    {
        "txList":
        [
            "txId",
            ...
        ]
    }
}

```

字段说明:

字段名	字段名称	字段描述
assetId	资产 ID	最大 20 个字节
txClass	交易分类	1--所有的 2--正常的 3--锁定的 4--发行 5--追加发行 6--销毁

### 3.11. 按钱包地址+资产 ID 查询交易列表

方法:

getaddrassettxids

请求:

```

{
    "safeAddress": "", // 地址
    "assetId": "", // 资产 ID
    "txClass": // 交易分类
}

```

返回:

```

{
    "error":,
    "id": "",
    "result":
    {
        "txList":
        [
            "txId",
            ...
        ]
    }
}

```

```

    }
  }

```

字段说明:

字段名	字段名称	字段描述
safeAddress	钱包地址	
assetId	资产 ID	最大 20 个字节
txClass	交易分类	1--所有的 2--正常的 3--锁定的 4--发行 5--追加发行 6--销毁

### 3.12. 按钱包地址+资产 ID 查询余额

方法:

getaddrassetbalance

请求:

```

{
  "safeAddress": "", // 地址
  "assetId": "", // 资产 ID
}

```

返回:

```

{
  "error": "",
  "id": "",
  "result": {
    "ReceiveAmount": "", // 总接收数量
    "SendAmount": "", // 总发送数量
    "totalAmount": "", // 总数量
    "lockAmount": "" // 锁定数量
  }
}

```

字段说明:

字段名	字段名称	字段描述
safeAddress	钱包地址	
assetId	资产 ID	最大 20 个字节

### 3.13. 按交易 ID 查询安资详情

方法:

getassetdetails

请求:

```
{
  "txId":"" // 交易 ID
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "txData":
    {
      ... // 此处可能是发行、追加发行、转账、销毁数据;
    }
  }
}
```

### 3.13.1. 发行

```
{
  "adminSafeAddress": "", // 管理员地址
  "assetShortName": "", // 资产简称
  "assetName": "", // 资产名称
  "assetDesc": "", // 资产描述
  "assetUnit": "", // 资产单位
  "assetTotalAmount": "", // 资产总量
  "firstIssueAmount": "", // 初次发行总量
  "assetDecimals":, // 小数点
  "isDestory": // 是否可以销毁
  "isPayCandy":, // 是否分发糖果
  "assetCandyAmount": "", // 糖果数量
  "candyExpired":, // 糖果过期时间
  "remarks": "" // 备注
}
```

### 3.13.2. 追加发行

```
{
  "adminSafeAddress": "", // 管理员地址
  "assetId": "", // 资产 ID
  "addAmount": "", // 追加数量
  "remarks": "" // 备注
}
```

### 3.13.3. 转账

```
{
  "safeAddress": "", // 收款地址
}
```

```
    "assetId": "", // 资产 ID
    "assetAmount": "", // 资产数量
    "lockTime": "", // 锁定时长
    "remarks": "" // 备注
}
```

#### 3.13.4. 销毁

```
{
    "assetId": "", // 资产 ID
    "assetAmount": "", // 资产数量
    "remarks": "" // 备注
}
```

#### 3.13.5. 糖果发放

```
{
    "assetId": "", // 资产 ID
    "assetCandyAmount": "", // 糖果数量
    "CandyExpired": "", // 糖果过期时间
    "remarks": "" // 备注
}
```

#### 3.13.6. 糖果领取

```
{
    "assetId": "", // 资产 ID
    "assetCandyAmount": "", // 糖果数量
    "remarks": "" // 备注
}
```

### 3.14. 获取所有资产列表

方法:

getassetlist

请求:

返回:

```
{
    "error": "",
    "id": "",
    "result": {
        "assetList": [
            "assetId", // 资产 ID
            ....
        ]
    }
}
```



```
}
```

### 3.15. 根据 safe 地址返回资产列表

方法:

getassetlistbyaddress

请求:

```
{
  "safeAddress":"" // safe 地址
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "assetList":
    [
      "assetId",
      ....
    ]
  }
}
```

### 3.16. 获取指定高度和地址对应的 SAFE 金额

方法:

getaddressamountbyheight

请求:

```
{
  "height": // 高度
  "address":"" // safe 地址
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "amount":, // 该地址在当前高度下的 SAFE 金额
    "totalAmount": // 当前高度下所有 SAFE 地址的金额总和
  }
}
```

### 3.17. 获取所有发放糖果对应的区块高度列表

方法:

getallcandyheight

请求: 无

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "heightList":
    [
      10000,      // 区块高度
      .....
    ]
  }
}
```

### 3.18. 获取地址领取的糖果记录

方法:

getaddresscandylist

请求:

```
{
  "assetId": ""           // 资产 id
  "safeAddress": ""      // safe 地址 (可选)
}
```

返回:

```
{
  "error":,
  "id": "",
  "result":
  {
    "candyBlockTime":
    "details ":
    [
      {
        "safeAddress ": "",
        "candyAmount": ""
      }
      .....
    ]
  }
}
```

```
    ]  
  }  
}
```

### 3.19. 查询本地资产信息

方法:

getlocalassetinfo

请求:

```
{  
  "assetId":"" // 资产 id  
}
```

返回:

```
{  
  "error":,  
  "id":",  
  "result":  
  {  
    "assetShortName":", // 资产简称  
    "assetName":", // 资产名称  
    "assetDesc":", // 资产描述  
    "assetUnit":", // 资产单位  
    "assetDecimals":, // 最小单位  
    "issueTime":, // 发行时间  
    "assetAvailAmount":", // 资产可用总量  
    "assetWaitAmount":", // 资产等待中总量  
    "assetLockAmount":", // 资产锁定总量  
    "assetLocalTotalAmount":", // 本地资产总量  
  }  
}
```

### 3.20. 查询本地资产列表

方法:

getlocalassetlist

请求:

返回:

```
{  
  "error":,  
  "id":",  
  "result":  
  {  
    "assetList":
```

```
    [
      "assetId",      // 资产 ID
      ....
    ]
  }
}
```

#### 4. 公共接口

##### 4.1. 获取交易信息

方法:

getrawtransaction

请求:

参照原来的, 它有两个参数, 分别是: 交易 ID、输出数据的格式

返回:

```
{
  "hex": "",
  "txid": "",
  "size": ,
  "version": ,
  "locktime": ,
  "vin":
  [
    {
      "coinbase": "",
      "sequence":
    }
  ],
  "vout":
  [
    {
      "value": ,
      "valueSat": ,
      "n": ,
      "scriptPubKey":
      {
        "asm": "",
        "hex": "",
        "reqSigs": ,
        "type": "",
        "addresses":
        [
```

```

        ""
        ,
        ...
    ]
},
"unlockHeight": ,
"reserve": "",
"txType": "",           // 交易类型
"assetId": ""          // 资产 ID
}
,
...
]
,
"blockhash": "",
"height": ,
"confirmations": ,
"time": ,
"blocktime":
}

```

在 vout 中增加标注为红色的字段，字段说明如下：

字段名	字段名称	字段描述
txType	交易类型	1--SAFE 2--安资 3--应用
assetId	资产 ID	交易类型为安资时,该字段才有值, 其它的都为空