

SDChain-Rest-js

- 接口说明
 - 1 账户接口
 - 1.1 获取新钱包
 - 1.2 获得账户余额
 - 1.3 获得账户发行通证流通总量
 - 2 账户设置
 - 2.1 查看账户信息
 - 2.2 账户设置
 - 3 支付接口
 - 3.1 支付请求
 - 3.2 批量支付请求
 - 3.3 获得支付信息
 - 3.4 获得支付历史
 - 4 授信接口
 - 4.1 查看授信记录
 - 4.2 授信/取消授信
 - 5 交易接口
 - 5.1 获取交易历史
 - 5.2 获取交易详情
 - 5.3 签名
 - 5.4 提交
 - 6 服务器基本信息接口
 - 6.1 获取默认费率
 - 6.2 获取服务器信息
 - 7 智能合约接口
 - 7.1 部署智能合约
 - 7.2 调用智能合约
 - 8 挂单接口
 - 8.1 提交挂单
 - 8.2 获取账户挂单列表
 - 8.3 取消挂单
 - 8.4 获取当前挂单列表

接口说明

接口地址: <https://testnet-cable.sdchain.io>

基础通证: SDA

资产对象: Amount

| 参数 | 类型 | 说明 |
|----------|--------|------|
| value | String | 数量 |
| currency | String | 通证符号 |

| 参数 | 类型 | 说明 |
|--------------|--------|------------------|
| counterparty | String | 发行方, 当通证为基础通证时为空 |

```
{
  "value": "1.1",
  "currency": "USD",
  "counterparty": "6wDwUi99cwgiEv8Zb3whNUzqWpyi7Jm6KX"
}
```

交易资产对象: AmountI

| 参数 | 类型 | 说明 |
|----------|--------|------------------|
| value | String | 数量 |
| currency | String | 通证符号 |
| issuer | String | 发行方, 当通证为基础通证时为空 |

```
{
  "value": "1.1",
  "currency": "USD",
  "issuer": "6wDwUi99cwgiEv8Zb3whNUzqWpyi7Jm6KX"
}
```

账户交易提交url通用参数

| 参数 | 类型 | 说明 |
|-----------|--------|-----------------------|
| submit | String | 是否立即提交交易 (可选, 默认true) |
| validated | String | 是否等待验证 (可选, 默认false) |
| unsign | String | 是否在线签名 (可选, 默认false) |

接口错误返回参数

| 参数 | 类型 | 说明 |
|------------|---------|------------------------------------|
| success | boolean | 返回false |
| error_type | String | 错误类型 |
| message | String | 错误信息 |
| error | String | 错误类型 |
| hash | String | 交易hash |
| state | String | 交易状态(pending: 不一定失败, failed: 一定失败) |

```
{
  "success": false,
  "error_type": "tecNO_DST",
  "message": "Destination does not exist. ",
  "error": "tecNO_DST",
  "hash": "9EF2FE1E71E135FFF1D94E97F046376857ED3BFCEB93008AD8A41CD476AB096C",
  "state": "pending"
}
```

1 账户接口

1.1 获取新钱包

接口：/v1/wallet/new, GET方法

请求参数：

| 参数 | 类型 | 说明 |
|--------|--------|------------------------|
| secret | String | 秘钥 (可选,如果传入, 可获取对应的地址) |

例子：

/v1/wallet/new?secret=ss6Vi*****pDAJ

结果：

```
{
  "wallet": {
    "address": "6wDwUi99cwgiEv8Zb3whNUzqWpyi7Jm6KX",
    "secret": "ss6Vi*****pDAJ"
  },
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|------|
| success | Boolean | 请求结果 |
| wallet | Object | 钱包 |
| address | String | 钱包地址 |
| secret | String | 秘钥 |

注：新获取的钱包为未激活。激活方式：使用支付接口从已激活账户中向新账户转账0.6个基础通证。

1.2 获得账户余额

接口：/v1/accounts/balances/{:address}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

请求参数：

| 参数 | 类型 | 说明 |
|--------------|--------|-------------------|
| currency | String | 通证名称（可选） |
| counterparty | String | 发行商地址（可选） |
| marker | String | 下一页参数（可选） |
| limit | String | 每页显示条数(可选, 默认100) |

例子：

/v1/accounts/balances/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5

结果：

```
{
  "ledger": 92261,
  "account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
  "ownerCount": 0,
  "balances": [
    {
      "value": "1959.99994",
      "currency": "SDA",
      "counterparty": ""
    }
  ],
  "success": true
}
```

返回的结果信息：

| 参数 | 类型 | 说明 |
|------------|---------|--------------|
| success | Boolean | 请求结果 |
| ledger | Integer | 区块高度 |
| account | String | 当前地址 |
| ownerCount | Integer | 冻结基数 |
| balances | Array | 余额, Amount数组 |

| 参数 | 类型 | 说明 |
|--------------|--------|------------------|
| value | String | 数量 |
| currency | String | 通证符号 |
| counterparty | String | 发行方, 当通证为基础通证时为空 |

1.3 获得账户发行通证流通总量

接口：/v1/accounts/gatewayBalances/{:address}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

例子：

/v1/accounts/gatewayBalances/6UPd52jHtu1d88nc3S3WeroACFQpKfybhU

结果：

```
{
  "ledger": 1873409,
  "account": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
  "obligations": {
    "AAA": "4000000000",
    "BBB": "5.01",
    "GD": "9.988"
  },
  "success": true
}
```

返回的结果信息：

| 参数 | 类型 | 说明 |
|-------------|---------|--------|
| success | Boolean | 请求结果 |
| ledger | Integer | 区块高度 |
| account | String | 当前地址 |
| obligations | Object | 发行通证列表 |
| 通证名称 | String | 流通量 |

2 账户设置

2.1 查看账户信息

接口：/v1/accounts/settings/{:address}，GET方法

接口参数

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

例子：

/v1/accounts/settings/6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc

结果：

```
{
  "settings": {
    "account": "6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc",
    "default_spread": false
  },
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|----------------|---------|-------------------------------|
| success | Boolean | 请求结果 |
| settings | Object | 设置信息 |
| account | String | 钱包地址 |
| default_spread | String | 是否能流通 默认false，作为发行商时设置为true |
| limitAmount | Array | 总量限制 |
| value | String | 限制总量 |
| currency | String | 限制通证 |
| transferFee | Amount | 二级通证费用(可选，counterparty需为当前地址) |

2.2 账户设置

接口：/v1/accounts/settings/{:address}，POST方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|-------|
| address | String | 六域链地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|----------------|--------|------------------------------------|
| secret | String | 秘钥 |
| settings | Object | 设置信息 |
| default_spread | String | 是否能流通 (可选, 默认false, 作为发行商时设置为true) |
| limitAmount | Amount | 总量限制,(可选, counterparty需为当前地址) |
| transferFee | Amount | 二级通证费用,(可选, counterparty需为当前地址) |

例子：

/v1/accounts/settings/6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc

提交的数据：

```
{
  "secret": "snoNfF*****fgXvmT3c",
  "settings": {
    "default_spread": true,
    "limitAmount": {
      "value": "1",
      "currency": "USD",
      "counterparty": "6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc"
    }
  }
}
```

结果：

```
{
  "settings": {
    "default_spread": false
  },
  "hash": "113085DCB6783C8E8C6689F2F4C07CC8DC5BA8461F9E3E99C4984FBB0DE3C115",
  "ledger": "354718",
  "state": "pending",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|--------|
| success | Boolean | 请求结果 |
| hash | String | 交易hash |
| ledger | String | 区块高度 |

| 参数 | 类型 | 说明 |
|----------|--------|------|
| state | String | 交易状态 |
| settings | Object | 设置信息 |
| account | String | 钱包地址 |

3 支付接口

3.1 支付请求

接口：/v1/accounts/payments/{:address}, POST方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|---------------------|-----------------|-------------------------------|
| secret | String | 秘钥 |
| payment | Object | 支付对象 |
| source_account | String | 发起地址 |
| destination_account | String | 目标地址 |
| destination_tag | String(Integer) | 转账标志（可选） |
| amount | String/AmountI | 支付金额对象（当类型为String时，通证默认为基础通证） |
| value | String | 数量 |
| currency | String | 通证名称 |
| issuer | String | 发行商地址，当通证为基础通证时为空 |
| memos | Array | 备注数组（可选） |
| memo_type | String(字母+数字) | 备注类型 |
| memo_data | String | 备注内容 |

例子：

/v1/accounts/payments/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5

```
//基础通证
{
```



```

"secret": "saNYbpu4*****3Hhg8kRu",
"payment": {
  "source_account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
  "destination_account": "6PrXdWVoXzmGCmi56wTFTC3e6xezLWh2dc",
  "amount": "5.01",
  "memos": [
    {
      "memo_type": "memos",
      "memo_data": "第一笔交易"
    }
  ]
}

```

```

//二级通证
{
  "secret": "saNYbpu4*****3Hhg8kRu",
  "payment": {
    "source_account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
    "destination_account": "6PrXdWVoXzmGCmi56wTFTC3e6xezLWh2dc",
    "amount": {
      "value": "5.01",
      "currency": "USD",
      "issuer": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5"
    },
    "memos": [
      {
        "memo_type": "memos",
        "memo_data": "第一笔交易"
      }
    ]
  }
}

```

结果：

```

{
  "hash": "12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E",
  "resultType": "tesSUCCESS",
  "resultMessage": "The transaction was applied. Only final in a validated ledger.",
  "ledger": 0,
  "state": "pending",
  "status_url": "https://testnet-cable.sdchain.io/v1/accounts/payments/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5/12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E",
  "success": true
}

```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------------|---------|---------------------------------|
| success | Boolean | 请求结果 |
| hash | String | 支付交易Hash |
| resultType | String | 交易结果 |
| resultMessage | String | 交易结果信息 |
| ledger | Integer | 交易所在的区块高度 (未验证时统一返回0) |
| state | String | 交易状态(validated,pending, failed) |
| status_url | String | 查看交易状态的地址 |

3.2 批量支付请求

接口：/v1/accounts/multiPayments/{:address}, POST方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|-------------|---------------------------------|---------------------------------|
| secret | String | 密钥 |
| payments | Object | 支付对象 |
| payees | Array | 目标地址数组 |
| destination | String | 目标地址 |
| amount | String/ AmountI | 支付金额对象 (当类型为String时, 通证默认为基础通证) |
| memos | Array | 备注数组 (可选) |
| memo_type | String(字母+数字) | 备注类型 |
| memo_data | String | 备注内容 |

例子：

/v1/accounts/multiPayments/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5

```
{
  "secret": "saNYbpu4*****3Hhg8kRu",
  "payments": {
    "payees": [
```

```

        {
            "destination": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
            "amount": "1"
        },
        {
            "destination": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
            "amount": {
                "value": "1",
                "currency": "BBB",
                "issuer": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
            }
        }
    ],
    "memos": [
        {
            "memo_type": "emoe",
            "memo_data": "这是一个批量转账"
        }
    ]
}

```

结果：

```

{
    "hash": "12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E",
    "resultType": "tesSUCCESS",
    "resultMessage": "The transaction was applied. Only final in a validated ledger.",
    "ledger": 0,
    "state": "pending",
    "success": true
}

```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------------|---------|---------------------------------|
| success | Boolean | 请求结果 |
| hash | String | 支付交易Hash |
| resultType | String | 交易结果 |
| resultMessage | String | 交易结果信息 |
| ledger | Integer | 交易所在的区块高度 (未验证时统一返回0) |
| state | String | 交易状态(validated,pending, failed) |
| status_url | String | 查看交易状态的地址 |

3.3 获得支付信息

接口：/v1/accounts/payments/{:address}/{:hash}，GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|--------|
| address | String | 钱包地址 |
| hash | String | 交易hash |

例子：

/v1/accounts/payments/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5/12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E

结果：

```
{
  "hash": "12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E",
  "ledger": "86520",
  "state": "validated",
  "result": "tesSUCCESS",
  "source_account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
  "destination_account": "6PrXdWVoXzmGCmi56wTFTC3e6xezLWh2dc",
  "destination_tag": "",
  "amount": {
    "currency": "SDA",
    "value": "20",
    "issuer": ""
  },
  "currency": "SDA",
  "issuer": "",
  "direction": "outgoing",
  "timestamp": "2018-03-06T08:15:42.000Z",
  "date": 1520324142,
  "fee": "0.000012",
  "memos": [
    {
      "memo_type": "6D656D6F73",
      "memo_data": "E7ACACE4B880E7AC94E4BAA4E69893"
    }
  ],
  "simple_memos": [
    {
      "memo_type": "memo",
      "memo_data": "第一笔交易"
    }
  ],
  "success": true
}
```

返回的结果信息：

| 参数 | 类型 | 说明 |
|---------------------|---------|--------------------------------------|
| success | Boolean | 请求结果 |
| hash | String | 交易Hash |
| ledger | String | 区块高度 |
| state | String | 状态 (validated : 已完成 ; pending : 交易中) |
| result | String | 交易结果(tesSUCCEs,为成功) |
| source_account | String | 付款方 |
| destination_account | String | 收款方 |
| direction | String | 交易类型(转出:outgoing,接收:incoming) |
| amount | AmountI | 交易资产对象 |
| timestamp | String | 交易时间 (UTC) |
| date | Integer | 交易时间 (时间戳s) |
| fee | String | 费率 |
| memos | Array | 支付的备注, 对象数组 (hex) |
| memo_type | String | 备注类型 (hex) |
| memo_data | String | 备注内容 (hex) |
| simple_memos | Array | 支付的备注, 对象数组 (UTF-8) |
| memo_type | String | 备注类型 (UTF-8) |
| memo_data | String | 备注内容 (UTF-8) |

3.4 获得支付历史

接口：/v1/accounts/payments/{:address}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|---------------------|--------|---------------|
| source_account | String | 支付方的钱包地址 (可选) |
| destination_account | String | 收款方的钱包地址 (可选) |

| 参数 | 类型 | 说明 |
|----------------|---------|-------------------------------------|
| direction | String | 交易类型, (可选,incoming:收款, outgoing:支付) |
| start_ledger | Integer | 起始区块高度, (可选) |
| end_ledger | Integer | 结束区块高度 (可选) |
| per_page | Integer | 每页数量 (可选) |
| page | Integer | 页码 (可选) |
| exclude_failed | Boolean | 是否包含失败的交易 (可选, 默认false) |

例子：

/v1/accounts/payments/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5?per_page=2&page=1

```
{
  "payments": [
    {
      "hash":
"BB4839ABD6484C4AC814F1C3AF651A1FE5808703BA9D6901CC2E14EF677784D7",
      "ledger": "1125346",
      "result": "tesSUCCESS",
      "source_account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
      "destination_account": "6JmTAjeQpXUMZACgGRNwvUG9EMSZmcBXU6",
      "amount": {
        "currency": "SDA",
        "value": "0.02",
        "issuer": ""
      },
      "direction": "outgoing",
      "timestamp": "2018-04-18T10:58:50.000Z",
      "date": 1524049130,
      "fee": "0.0001",
      "memos": [
        {
          "memo_type": "6D656D6F",
          "memo_data": "546869732069732061206D656D6F21"
        }
      ],
      "simple_memos": [
        {
          "memo_type": "memo",
          "memo_data": "第一笔交易"
        }
      ]
    },
    {
      "hash":
"81F36B67F23DF5E4D651EBA39537239E02A1F78534A48F05C010C4CF852F1768",
      "ledger": "1125346",
      "result": "tesSUCCESS",
```

```

    "source_account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
    "destination_account": "6JmTAjeQpXUMZACgGRNwvUG9EMSZmcBXU6",
    "amount": {
      "currency": "SDA",
      "value": "0.02",
      "issuer": ""
    },
    "direction": "outgoing",
    "timestamp": "2018-04-18T10:58:50.000Z",
    "date": 1524049130,
    "fee": "0.0001"
  }
],
"success": true
}

```

返回的结果信息：

| 参数 | 类型 | 说明 |
|----------|---------|-----------------|
| success | Boolean | 请求结果 |
| payments | Array | 支付历史, 同交易记录中的信息 |

4 授信接口

4.1 查看授信记录

接口：/v1/accounts/trustlines/{:address}, GET方法

接口参数

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

请求参数：

| 参数 | 类型 | 说明 |
|--------------|--------|-------------------|
| currency | String | 通证名称（可选） |
| counterparty | String | 发行商地址（可选） |
| marker | String | 下一页参数（可选） |
| limit | String | 每页显示条数(可选, 默认100) |

例子：

/v1/accounts/trustlines/6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc

结果：

```
{
  "trustlines": [
    {
      "account": "6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc",
      "counterparty": "6E7gcs7TS3by2iZCXrjYZkd2HthJcph34y",
      "currency": "USD",
      "limit": "500",
      "reciprocated_limit": "0"
    }
  ],
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|--------------------------------|---------|---------------|
| success | Boolean | 请求结果 |
| trustlines | Object | 授信信息 |
| account | String | 六域链地址 |
| currency | String | 通证名称 |
| counterparty | String | 授信方 |
| limit | String | 限制持有的最大数量 |
| reciprocated_limit | String | 对方限制持有的最大数量 |
| account_allows_sdchaining | String | 余额为0时是否保留授信 |
| counterparty_allows_sdchaining | String | 对方余额为0时是否保留授信 |

4.2 授信/取消授信

接口：/v1/accounts/trustlines/{:address}, POST方法

接口参数

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|-----------|--------|------|
| secret | String | 密钥 |
| trustline | Object | 授信信息 |

| 参数 | 类型 | 说明 |
|---------------------------|---------|---------------------------------|
| currency | String | 通证名称 |
| counterparty | String | 授信方地址 |
| limit | String | 限制持有的最大数量(取消授信时，设置为0，生效前提：余额为0) |
| account_allows_sdchaining | Boolean | 余额为0时是否保留授信（可选，默认true） |

例子：

/v1/accounts/trustlines/6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc

提交的数据：

```
{
  "secret": "snoNfFVWPK*****vefgXvmT3c",
  "trustline": {
    "limit": "110",
    "currency": "AAA",
    "counterparty": "6E7gcs7TS3by2iZCXrjYZkD2HthJcph34y "
  }
}
```

结果：

```
{
  "trustline": {
    "account": "6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc",
    "limit": "110",
    "currency": "AAA",
    "counterparty": "6E7gcs7TS3by2iZCXrjYZkD2HthJcph34y",
    "account_allows_sdchaining": true
  },
  "hash": "D8E82E7303B5E8F0B1312F7C2BCB9159A265DE8AD9D773081169224EDD28B1F3",
  "ledger": 0,
  "state": "pending",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|--------|
| success | Boolean | 请求结果 |
| hash | String | 交易hash |

| 参数 | 类型 | 说明 |
|--------------|---------|---------------------------------|
| ledger | Integer | 交易所在的区块高度 (未验证时统一返回0) |
| state | String | 交易状态(validated,pending, failed) |
| trustline | Object | 授信信息 |
| currency | String | 通证名称 |
| counterparty | String | 授信方 |
| limit | String | 限制持有的最大数量 |

5 交易接口

5.1 获取交易历史

接口：/v1/accounts/transactions/{:address}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|----------------|---------|--|
| type | String | 交易类型（可选，payment,offercreate,offercancel,trustset, 可提供多个用“,”隔开） |
| start_ledger | Integer | 起始区块高度（可选） |
| end_ledger | Integer | 结束区块高度（可选） |
| per_page | Integer | 每页数量（可选） |
| page | Integer | 页码（可选） |
| exclude_failed | Boolean | 是否包含失败的交易（可选，默认true） |

例子：

/v1/accounts/transactions/6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5?
type=payment,offercreate&per_page=2&page=1

结果：

```
{
  "transactions": [
    {
      "hash":
```

```

"BB4839ABD6484C4AC814F1C3AF651A1FE5808703BA9D6901CC2E14EF677784D7",
  "ledger": "1125346",
  "state": "validated",
  "account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
  "date": 1524049130,
  "time": "2018-04-18T10:58:50.000Z",
  "type": "outgoing",
  "fee": "0.0001",
  "result": "tesSUCCESS",
  "counterparty": "6JmTAjeQpXUMZACgGRNwvUG9EMSZmcBXU6",
  "amount": {
    "currency": "SDA",
    "value": "0.02",
    "issuer": ""
  },
  "effects": [],
  "memos": [
    {
      "memo_type": "6D656D6F",
      "memo_data": "546869732069732061206D656D6F21"
    }
  ],
  "simple_memos": [
    {
      "memo_type": "memos",
      "memo_data": "第一次交易"
    }
  ]
},
{
  "hash":
"81F36B67F23DF5E4D651EBA39537239E02A1F78534A48F05C010C4CF852F1768",
  "ledger": "1125346",
  "state": "validated",
  "account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
  "date": 1524049130,
  "time": "2018-04-18T10:58:50.000Z",
  "type": "outgoing",
  "fee": "0.0001",
  "result": "tesSUCCESS",
  "counterparty": "6JmTAjeQpXUMZACgGRNwvUG9EMSZmcBXU6",
  "amount": {
    "currency": "SDA",
    "value": "0.02",
    "issuer": ""
  },
  "effects": []
}
],
"success": true
}

```

返回的结果参数：

| 参数 | 类型 | 说明 |
|-----------------|---------|------------------------|
| success | Boolean | 请求结果 |
| transactions | Array | 交易对象数组 |
| hash | String | 交易hash |
| ledger | String | 区块高度 |
| state | String | 状态 |
| account | String | 发起地址 |
| date | Integer | 交易时间(s) |
| time | String | 交易时间 (UTC) |
| type | String | 类型 |
| fee | String | 费率 |
| result | String | 结果 |
| price | String | 单价(type=offercreate) |
| pair | String | 交易对(type=offercreate) |
| seq | Integer | 订单编号(type=offercreate) |
| destination_tag | Integer | 交易标志 |
| destination | String | 对方地址(type=payment) |
| amount | AmountI | 数量对象 |
| offertype | String | 挂单类型(type=offercreate) |
| effects | Array | 影响 |
| effect | String | 影响类型 |
| seq | Integer | 订单编号 |
| price | String | 单价 |
| pair | String | 交易对 |
| amount | Object | 数量对象 |
| memos | Array | 备注(hex) |
| simple_memos | Array | 备注(UTF-8) |

5.2 获取交易详情

接口：/v1/transactions/{:hash}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|------|--------|--------|
| hash | String | 交易hash |

例子：

/v1/transactions/BB4839ABD6484C4AC814F1C3AF651A1FE5808703BA9D6901CC2E14EF677784D7

结果：

```
{
  "transaction": {
    "hash":
"BB4839ABD6484C4AC814F1C3AF651A1FE5808703BA9D6901CC2E14EF677784D7",
    "ledger": "1125346",
    "state": "validated",
    "date": 1524049130,
    "type": "Payment",
    "fee": "0.0001",
    "result": "tesSUCCESS",
    "account": "6faLUhmp9gNgS9jXS3rAWbg8hFfi9PbWH5",
    "destination": "6JmTAjeQpXUMZACgGRNwvUG9EMSZmcBXU6",
    "amount": {
      "currency": "SDA",
      "value": "0.02",
      "issuer": ""
    },
    "effects": [],
    "memos": [
      {
        "memo_type": "6D656D6F",
        "memo_data": "546869732069732061206D656D6F21"
      }
    ],
    "simple_memos": [
      {
        "memo_type": "memos",
        "memo_data": "第一次交易"
      }
    ]
  },
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|------|
| success | Boolean | 请求结果 |

| 参数 | 类型 | 说明 |
|-----------------|---------|------------------------|
| transaction | Object | 交易对象 |
| hash | String | 交易hash |
| ledger | String | 区块高度 |
| state | String | 状态 |
| account | String | 发起地址 |
| date | Integer | 交易时间(s) |
| time | String | 交易时间 (UTC) |
| type | String | 类型 |
| fee | String | 费率 |
| result | String | 结果 |
| price | String | 单价(type=offercreate) |
| pair | String | 交易对(type=offercreate) |
| seq | Integer | 订单编号(type=offercreate) |
| destination_tag | Integer | 交易标志 |
| destination | String | 对方地址(type=payment) |
| amount | AmountI | 数量对象 |
| effects | Array | 影响 |
| effect | String | 影响类型 |
| seq | Integer | 订单编号 |
| price | String | 单价 |
| pair | String | 交易对 |
| amount | Amount | 数量对象(type=offercreate) |
| memos | Array | 备注 (hex) |
| simple_memos | Array | 备注 (UTF-8) |

5.3 签名

接口：/v1/transaction/sign, POST方法

提交参数详情：

| 参数 | 类型 | 说明 |
|--------|--------|----|
| secret | String | 秘钥 |

| 参数 | 类型 | 说明 |
|---------|--------|------|
| tx_json | Object | 交易信息 |

例子：

/v1/transaction/sign

提交的数据：

```
{
  "secret": "fafasfsa",
  "tx_json": {
    "TransactionType": "Payment",
    "Account": "6GiohV7XVhabbtV1DfKjWEyikeyimiDSNx",
    "Destination": "6BCYyTw4kthmnSH7jYWSdMkd1M6dtFYYP",
    "Amount": {
      "value": "100",
      "currency": "DUOS",
      "issuer": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
    },
    "Fee": "100",
    "Sequence": 15430
  }
}
```

结果：

```
{
  "tx_blob": "D8E82E7303B5E8F0B1312F7C2BCB9159A265DE8AD9D773081169224EDD28B1F3D8E82E7303B5E8F0B1312F7C2BCB9159A265DE8AD9D773081169224EDD28B1F3s",
  "hash": "D8E82E7303B5E8F0B1312F7C2BCB9159A265DE8AD9D773081169224EDD28B1F3",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|--------|
| success | Boolean | 请求结果 |
| hash | String | 交易hash |
| tx_blob | String | 交易信息 |

5.4 提交

接口：/v1/transaction/submit, POST方法

提交参数详情：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| tx_blob | String | 交易信息 |

例子：

/v1/transaction/submit

提交的数据：

```
{
  "tx_blob":
  "12001422000200002400003C4763D503E871B540C000000000000000000000000041414100000000
  0009ECE0D45F5224105C7912359B63754BE3519E4466840000000000000064732102E2982FD91D92E8E
  765961BB3E543869DA6A6905CDD58C3C1B1F8732B216ABADA74463044022026846B3A3AC52429F2B2E
  8BF0629CA3F3020C673D8CE16EAA6197E52BCCEC1DA02200D2F754BB1A3AC92EC09A7EFD89B849208A
  B38FACC8DC757AC6F89EDAEA10CDD8114ADF4B7CE6C8A91A10B720B058352DC24695DD029"
}
```

结果：

```
{
  "hash": "8492A2BFC1EE1D45389C5892E8CB634A3C1F766C9BE3AE292F280598BF519438",
  "resultType": "tesSUCCESS",
  "resultMessage": "The transaction was applied. Only final in a validated
  ledger.",
  "ledger": 0,
  "state": "pending",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------------|---------|---------------------------------|
| success | Boolean | 请求结果 |
| hash | String | 交易Hash |
| resultType | String | 交易结果 |
| resultMessage | String | 交易结果信息 |
| ledger | Integer | 交易所在的区块高度 (未验证时统一返回0) |
| state | String | 交易状态(validated,pending, failed) |

6 服务器基本信息接口

6.1 获取默认费率

接口：/v1/transaction-fee, GET方法

例子：

/v1/transaction-fee

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|--------|
| success | Boolean | 请求结果 |
| fee | String | 当前默认费率 |

6.2 获取服务器信息

接口：/v1/server, GET方法

例子：

/v1/server

结果：

```
{
  "sdchaind_server_status": {
    "complete_ledgers": "1-2520216",
    "server_state": "full",
    "reserve_base": 0.6,
    "reserve_inc": 0.2
  },
  "success": true
}
```

| 参数 | 类型 | 说明 |
|------------------------|---------|---|
| success | Boolean | 请求结果 |
| sdchaind_server_status | Object | 六域链服务器状况 |
| complete_ledgers | String | 完成的区块 |
| server_state | String | 服务器状况 |
| reserve_base | Double | 激活至少需要的基础通证数量 |
| reserve_inc | Double | 冻结的基础通证数量（账户基础通证冻结数量=reserve_base+reserve_inc*ownerCount） |

7 智能合约接口

7.1 部署智能合约

接口：/v1/accounts/smartContract/{:address}, POST方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|---------------|--------|-------------|
| secret | String | 秘钥 |
| smartContract | String | 智能合约代码(lua) |

例子：

/v1/accounts/smartContract/6hZD8GnU3j9qcCCf9b9yaJbuZUNtWEV9Zc

```
{
  "secret": "snoNfF*****fgXvmT3c",
  "smartContract": " for i=10,1,-1 do print(i);print(\"Hello SmartContract\");end;
"
}
```

结果：

```
{
  "smartContract": " for i=10,1,-1 do print(i);print(\"Hello SmartContract\");end; ",
  "hash": "3B5FC73019C8CA77791D6BAA3D8D80FC9AD5AC87797574CE76C457B3A5287E5F",
  "ledger": 0,
  "state": "pending",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------|---------|-----------------------|
| success | Boolean | 请求结果 |
| hash | String | 交易hash |
| ledger | String | 交易所在的区块高度 (未验证时统一返回0) |

| 参数 | 类型 | 说明 |
|---------------|--------|--------|
| state | String | 交易状态 |
| smartContract | String | 智能合约代码 |

7.2 调用智能合约

接口：/v1/smartContract/{:hash}, POST方法

接口参数：

| 参数 | 类型 | 说明 |
|------|--------|----------|
| hash | String | 智能合约hash |

提交参数详情：

| 参数 | 类型 | 说明 |
|--------|--------|------|
| txJson | Object | 提交参数 |

例子：

/v1/smartContract/3B5FC73019C8CA77791D6BAA3D8D80FC9AD5AC87797574CE76C457B3A5287E5F

```
{
  "txJson": {
    "option1": "123"
  }
}
```

结果：

```
{
  "smart_contract": "my smart contract success",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|----------------|---------------|------|
| success | Boolean | 请求结果 |
| smart_contract | String/Object | 返回结果 |

8 挂单接口

8.1 提交挂单

接口：/v1/accounts/orders/{:address}, POST方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|------------|--------|----------------------|
| secret | String | 秘钥 |
| order | Object | 挂单对象 |
| type | String | 挂单类型 (sell：卖, buy：买) |
| taker_pays | Amount | 对方应该花费的数量 |
| taker_gets | Amount | 对方应该得到的数量 |

例子：

/v1/accounts/orders/6GiohV7XVhabbtV1DfKjWEyikeymiDSNx

```
{
  "secret": "spoWzgEy9****1stWp2vsC",
  "order": {
    "type": "buy",
    "taker_pays": {
      "currency": "DDD",
      "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
      "value": "1"
    },
    "taker_gets": {
      "currency": "SDA",
      "counterparty": "",
      "value": "4"
    }
  }
}
```

结果：

```
{
  "hash": "12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E",
}
```

```
"resultType": "tesSUCCESS",
"resultMessage": "The transaction was applied. Only final in a validated
ledger.",
"ledger": 0,
"state": "pending",
"success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------------|---------|---------------------------------|
| success | Boolean | 请求结果 |
| hash | String | 交易hash |
| resultType | String | 交易结果 |
| resultMessage | String | 交易结果信息 |
| ledger | Integer | 交易所在的区块高度 (未验证时统一返回0) |
| state | String | 交易状态(validated,pending, failed) |

8.2 获取账户挂单列表

接口：/v1/accounts/orders/{:address}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|------|
| address | String | 钱包地址 |

提交参数详情：

| 参数 | 类型 | 说明 |
|-------|--------|-----------------|
| page | String | 页码（可选，默认1） |
| limit | String | 每页显示个数（可选，默认10） |

例子：

/v1/accounts/orders/6GlohV7XVhabbtV1DfKjWEyikymiDSNx?limit=1&page=2

结果：

```
{
  "orders": [
    {
      "type": "buy",
      "taker_gets": {
```

```
        "currency": "SDA",
        "value": "4",
        "counterparty": ""
      },
      "taker_pays": {
        "currency": "DDD",
        "value": "1",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
      },
      "sequence": 3123
    }
  ],
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|------------|---------|--------------------|
| success | Boolean | 请求结果 |
| orders | Array | 挂单数组 |
| type | String | 挂单类型（sell：卖，buy：买） |
| taker_pays | Amount | 对方应该花费的数量 |
| taker_gets | Amount | 对方应该得到的数量 |
| sequence | Integer | 挂单编号（用于取消挂单） |

8.3 取消挂单

接口：/v1/accounts/orders/{:address}/{:sequence}, DELETE方法

接口参数：

| 参数 | 类型 | 说明 |
|----------|---------|------|
| address | String | 钱包地址 |
| sequence | Integer | 挂单编号 |

提交参数详情：

| 参数 | 类型 | 说明 |
|--------|--------|----|
| secret | String | 秘钥 |

例子：

/v1/accounts/orders/6GiohV7XVhabbtV1DfKjWEyikeymiDSNx/5

```
{
  "secret": "sp0WzgEy9****1stWp2vsC"
}
```

结果：

```
{
  "hash": "12C323E8DEAF789B790E4837438E5E59E95CA888979B4BE1311CA4BEDAF7C05E",
  "resultType": "tesSUCCESS",
  "resultMessage": "The transaction was applied. Only final in a validated ledger.",
  "ledger": 0,
  "state": "pending",
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|---------------|---------|---------------------------------|
| success | Boolean | 请求结果 |
| hash | String | 交易hash |
| resultType | String | 交易结果 |
| resultMessage | String | 交易结果信息 |
| ledger | Integer | 交易所在的区块高度 (未验证时统一返回0) |
| state | String | 交易状态(validated,pending, failed) |

8.4 获取当前挂单列表

接口：/v1/order_book/{:base}/{:counter}, GET方法

接口参数：

| 参数 | 类型 | 说明 |
|---------|--------|-----------------------------------|
| base | String | 基础通证 (currency+counterparty或基础通证) |
| counter | String | 计算通证 (currency+counterparty或基础通证) |

提交参数详情：

| 参数 | 类型 | 说明 |
|-------|--------|-------------------|
| limit | String | 每页显示个数 (可选, 默认10) |

例子：

/v1/order_book/SDA/CNY+6UPd52jHtu1d88nc3S3WeroACFQpKfybhU?limit=2

结果：

```
{
  "order_book": "SDA/DDD+6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
  "bids": [],
  "asks": [
    {
      "price": {
        "currency": "DDD",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
        "value": "0.25"
      },
      "taker_gets_funded": {
        "currency": "SDA",
        "value": "4",
        "counterparty": ""
      },
      "taker_gets_total": {
        "currency": "SDA",
        "value": "4",
        "counterparty": ""
      },
      "taker_pays_funded": {
        "currency": "DDD",
        "value": "1",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
      },
      "taker_pays_total": {
        "currency": "DDD",
        "value": "1",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
      },
      "order_maker": "6GiohV7XVhabbtV1DfKjWEyikeymiDSNx",
      "sequence": 3117,
      "passive": false,
      "sell": false
    },
    {
      "price": {
        "currency": "DDD",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU",
        "value": "0.25"
      },
      "taker_gets_funded": {
        "currency": "SDA",
        "value": "4",
        "counterparty": ""
      },
      "taker_gets_total": {
        "currency": "SDA",
        "value": "4",
```



```
        "counterparty": ""
      },
      "taker_pays_funded": {
        "currency": "DDD",
        "value": "1",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
      },
      "taker_pays_total": {
        "currency": "DDD",
        "value": "1",
        "counterparty": "6UPd52jHtu1d88nc3S3WeroACFQpKfybhU"
      },
      "order_maker": "6GiohV7XVhabbtV1DfKjWEyikeymiDSNx",
      "sequence": 3118,
      "passive": false,
      "sell": false
    }
  ],
  "success": true
}
```

结果返回参数：

| 参数 | 类型 | 说明 |
|------------------|---------|-----------|
| success | Boolean | 请求结果 |
| order_book | String | 交易对 |
| bids | Array | 买单信息 |
| price | Amount | 单价 |
| taker_gets_total | Amount | 对方应该获取的数量 |
| taker_pays_total | Amount | 对方应该花费的数量 |
| order_maker | String | 挂单方地址 |
| sequence | Integer | 挂单编号 |
| asks | Array | 卖单信息 |