**SDK 接口定义**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ ] 正式发布  [ √ ]正在修改 | 文件标识： | BUMO-SDK |
| 当前版本： | 1.0.1 |
| 作 者： |  |
| 完成日期： | 2018/07/02 |

目录

[1. 目录结构 8](#_Toc523390138)

[1.1 sdk文件 9](#_Toc523390139)

[1.2 blockchain目录 9](#_Toc523390140)

[1.3 account目录 9](#_Toc523390141)

[1.4 token目录 10](#_Toc523390142)

[1.5 contract目录 10](#_Toc523390143)

[1.6 log目录 11](#_Toc523390144)

[1.7 base文件 11](#_Toc523390145)

[1.8 autoload文件 11](#_Toc523390146)

[1.9 loginfo目录 11](#_Toc523390147)

[2.0 monolog目录 11](#_Toc523390148)

[2. 接口使用说明 12](#_Toc523390149)

[2.1 交易提交过程 12](#_Toc523390150)

[2.1.1 获取交易nonce 12](#_Toc523390151)

[2.1.2 生成blob 12](#_Toc523390152)

[2.1.3 签名blob 12](#_Toc523390153)

[2.1.4 提交transaction 13](#_Toc523390154)

[2.2 查询或其他方法 13](#_Toc523390155)

[3. 参数与响应 13](#_Toc523390156)

[3.1 请求参数 14](#_Toc523390157)

[3.1.1 查询及验证的方法 14](#_Toc523390158)

[3.1.2 生成交易操作的方法 14](#_Toc523390159)

[3.2 响应结果 15](#_Toc523390160)

[3.2.1 查询及验证的方法 15](#_Toc523390161)

[3.2.2 生成交易操作的方法 16](#_Toc523390162)

[3. account类 16](#_Toc523390163)

[3.1 生成交易操作的方法 17](#_Toc523390164)

[3.1.1 activate 17](#_Toc523390165)

[3.1.2 setMetadata 18](#_Toc523390166)

[3.1.3 setPrivilege 19](#_Toc523390167)

[3.2 查询及验证方法 21](#_Toc523390168)

[3.2.1 checkValid 21](#_Toc523390169)

[3.2.2 create 22](#_Toc523390170)

[3.2.3 getInfo 23](#_Toc523390171)

[3.2.4 getNonce 25](#_Toc523390172)

[3.2.5 getBalance 26](#_Toc523390173)

[3.2.6 getAssets 27](#_Toc523390174)

[3.2.7 getMetadata 28](#_Toc523390175)

[4. asset类 29](#_Toc523390176)

[4.1 生成交易操作的方法 29](#_Toc523390177)

[4.1.1 issue 29](#_Toc523390178)

[4.1.2 send 31](#_Toc523390179)

[4.2 查询方法 32](#_Toc523390180)

[4.2.1 getInfo 32](#_Toc523390181)

[5. bu类 34](#_Toc523390182)

[5.1 生成交易操作的方法 34](#_Toc523390183)

[5.1.1 send 34](#_Toc523390184)

[6. ctp10Token类(不用封装) 35](#_Toc523390185)

[6.1 生成交易操作的方法 36](#_Toc523390186)

[6.1.1 issue 36](#_Toc523390187)

[6.1.2 transfer 37](#_Toc523390188)

[6.1.3 transferFrom 39](#_Toc523390189)

[6.1.4 approve 41](#_Toc523390190)

[6.1.5 assign 42](#_Toc523390191)

[6.1.6 changeOwner 44](#_Toc523390192)

[6.2 查询及验证方法 45](#_Toc523390193)

[6.2.1 checkValid 45](#_Toc523390194)

[6.2.2 allowance 46](#_Toc523390195)

[6.2.3 getInfo 48](#_Toc523390196)

[6.2.4 getName 49](#_Toc523390197)

[6.2.5 getSymbol 50](#_Toc523390198)

[6.2.6 getDecimals 51](#_Toc523390199)

[6.2.7 getTotalSupply 52](#_Toc523390200)

[6.2.8 getBalance 53](#_Toc523390201)

[7. contract类 54](#_Toc523390202)

[7.1 生成交易操作的方法 54](#_Toc523390203)

[7.1.1 create 54](#_Toc523390204)

[7.1.2 invokeByAsset 56](#_Toc523390205)

[7.1.3 invokeByBU 57](#_Toc523390206)

[7.2 查询及验证方法 59](#_Toc523390207)

[7.2.1 getInfo 59](#_Toc523390208)

[7.2.2 getAddress 60](#_Toc523390209)

[7.2.3 checkValid 61](#_Toc523390210)

[7.2.4 call 62](#_Toc523390211)

[8. log类 67](#_Toc523390212)

[8.1 生成交易操作的方法 67](#_Toc523390213)

[8.1.1 create 67](#_Toc523390214)

[9. transaction类 69](#_Toc523390215)

[9.2 buildBlob 69](#_Toc523390216)

[9.3 evaluationFee 71](#_Toc523390217)

[9.4 sign 72](#_Toc523390218)

[9.5 submit 73](#_Toc523390219)

[9.6 getInfo 74](#_Toc523390220)

[10. block类 79](#_Toc523390221)

[10.1 getNumber 79](#_Toc523390222)

[10.2 checkStatus 80](#_Toc523390223)

[10.3 getTransactions 81](#_Toc523390224)

[10.4 getInfo 85](#_Toc523390225)

[10.5 getLatestInfo 86](#_Toc523390226)

[10.6 getValidators 87](#_Toc523390227)

[10.7 getLatestValidators 88](#_Toc523390228)

[10.8 getReward 89](#_Toc523390229)

[10.9 getLatestReward 90](#_Toc523390230)

[10.10 getFees 91](#_Toc523390231)

[10.11 getLatestFees 92](#_Toc523390232)

[11. sdk 93](#_Toc523390233)

[11.1 getInstance 93](#_Toc523390234)

[12. 错误码 94](#_Toc523390235)

[12.1 统一错误码 94](#_Toc523390236)

[12.2 Java错误码 99](#_Toc523390237)

[12.3 Javascript错误码 99](#_Toc523390238)

[12.4 go错误码 99](#_Toc523390239)

# 1. 目录结构

根据功能，将目录分为account，token，sdk，common，model, crypto，contract，log, exception。详细介绍:

|\_\_ account目录: 主要用于账户信息的操作与查询

|\_ account文件: 用于账户信息的操作与查询

|\_\_ token目录: 主要用于token信息的操作与查询

|\_\_ asset文件: 账户资产的操作与查询

|\_\_ bu文件: 账户BU的发送

|\_\_ ctp10Token文件: 合约资产的操作与查询(本期不写)

|\_\_ blockchain目录: 主要用于接口的呈现，所有对外的接口都是以这个为基础呈现的

|\_\_ transaction文件: 交易的处理，包括交易的评估、生成blob、签名、查询等

|\_\_ block文件: 区块查询

|\_\_ common目录: 主要用于编写公共处理模块，如http的响应等

|\_\_ model目录: 主要存放请求和响应的数据结构

|\_\_ request: 存放请求数据结构

|\_\_ response: 存放响应数据结构

|\_\_ crypto目录: 主要用于加密模块，比如protobuf等

|\_\_ contract目录: 主要用于合约的创建与查询

|\_\_ contract文件: 合约的创建与查询

|\_\_ log目标: 主要用于在链上写日志

|\_\_log文件: 创建

|\_\_ exception目录: 主要提供错误码与错误描述

|\_\_ sdk文件: 统一所有接口

## 1.1 sdk文件

sdk类，统一所有类，成员函数包括[getInstance](#11.1 getInstance)

## 1.2 blockchain目录

目前有transaction, block和sdk三个类

1. transaction类，成员函数包括 [buildBlob](#9.2 buildBlob), [buildBlob](#9.2 buildBlob)Object, [evaluationFee](#9.3 evaluationFee), [sign](#9.4 sign), [sign](#9.4 sign)Object, [submit](#9.5 submit), [submit](#9.5 submit)Object, [getInfo](#9.6 getInfo)
2. block类, 成员函数包括[getNumber](#10.1 getNumber), [checkStatus](#10.2 checkStatus), [getTransactions](#10.3 getTransactions), [getInfo](#10.4 getInfo), [getLatestInfo](#10.5 getLatestInfo), [getValidators](#10.6 getValidators), [getLatestValidators](#10.7 getLatestValidators), [getReward](#10.8 getReward), [getLatestReward](#10.9 getLatestReward), [getFees](#10.10 getFees), [getLatestFees](#10.11 getLatestFees)

## 1.3 account目录

目前只有一个account类，成员函数包括

a. 生成交易操作的方法: [activate](#3.1.1 activate), [setMetadata](#3.1.2 setMetadata), [setPrivilege](#3.1.3 setPrivilege)

b. 查询及其他方法: [checkValid](#3.2.1 checkValid), [create](#3.2.2 create), [getInfo](#3.2.3 getInfo), [getAssets](#3.2.6 getAssets), [getNonce](#3.2.4 getNonce), [getBalance](#3.2.5 getBalance), [getMetadata](#3.2.7 getMetadata)

## 1.4 token目录

目前asset, bu两个类

(1) asset类，遵循ATP1.0协议，成员函数包括

a. 生成交易操作的方法: [issue](#4.1.1 issue), [send](#4.1.2 send)

b. 查询方法: [getInfo](#4.2.1 getInfo)

(2) bu类，成员函数 包括生成交易操作的方法send()

## 1.5 contract目录

目前只有一个contract类，成员函数包括:

a. 生成交易操作的方法：[create](#7.1.1 create), [invokeByAsset](#7.1.2 invokeByAsset), [invokeByBU](#7.1.3 invokeByBU)

b. 查询，验证，调用合约操作: [getInfo](#7.2.1 getInfo), [getAddress](#7.2.2 getAddress), [call](#7.2.4 call), [checkValid](#7.2.3 checkValid)

## 1.6 log目录

目前只有一个log类，成员函数包括：

生成交易操作的方法：[create](#8.1.1 create)

## 1.7 base文件

Base类，所有业务接口类的基类，包含http get请求、post请求等

## 1.8 autoload文件

Autoload类，自动加载需要的类文件

## 1.9 loginfo目录

服务器日志目录

## 2.0 monolog目录

用于生成服务器日志，方便调试

# 2. 接口使用说明

这里详细介绍接口的使用规范，包括交易提交的接口调用过程，查询的接口调用过程和其他接口等等

## 2.1 交易提交过程

交易的提交过程，接口分为四部分：组装operation，生成blob，签名blob，提交transaction

### 2.1.1 获取交易nonce

获取交易序列号，是由account目录下的account类提供。

### 2.1.2 生成blob

生成blob的操作接口，是由sdk目录下的transaction类提供的。 返回值是Protocol Buffer中的Transaction类序列化后，再进行16进制编码后得到的16进制字符串。

### 2.1.3 签名blob

签名blob的操作接口，是由sdk目录下的transaction类提供的。这里不是对2.1.2中生成blob本身进行签名，是对blob进行16进制解码后得到的数据进行签名，也就是对Protocol Buffer中的Transaction类序列化后得到的数据进行签名。

因此，这个接口的入参是blob和私钥，接口中需要先对blob进行16进制解码，然后再用私钥进行签名; 返回值是签名后的数据和公钥。

### 2.1.4 提交transaction

提交transaction的操作接口，是由sdk目录下的transaction类提供的。这里需要将2.1.2中生成的blob和2.1.3中生成的签名后数据和公钥组装成json对象，调用底层api接口submitTransaction，实现交易的提交工作。

注意：这里的api接口submitTransaction返回的错误码，如果错误码非0，表示交易提交失败。如果错误码是0，表示交易提交成功，但是并不表示交易执行成功，要确定交易是否执行成功，需要在每个区块生成后（区块每10秒生成一个）调用api接口getTransactionHistory进行查询，如果超过50个区块后，getTransactionHistory一直返回错误码4，则表示，该交易执行失败。

## 2.2 查询或其他方法

直接调用对应的方法即可得到结果

# 3. 参数与响应

请求参数与响应结果都以固定的格式提供

## 3.1 请求参数

### 3.1.1 查询及验证的方法

该类方法供用户调用的。

请求参数的格式，是[类名][方法名]Request，比如account.create的请求参数是AccountCreateRequest。

请求参数的成员，是各个方法的入参的成员变量名。例如：account.create的入参成员是privateKey, publicKey, address，那么AccountCreateOperation的结构如下：

Class AccountCreateOperation{

String privateKey;

String publicKey;

String address;

}

AccountCreateOperation继承于基类OperationBase，OperationBase有两个共有成员变量：

Class OperationBase {

String sourceAddress;

String metadata;

}

### 3.1.2 生成交易操作的方法

该类方法不供用户调用，供内部接口transaction.buildBlob()方法调用的。

请求参数的格式，是[类名][方法名]Operation，比如 account.activate的请求参数是AccountActiviateOperation.

请求参数的成员，是各个方法的入参的成员变量名。例如：account.activate的入参成员是sourceAddress, destAddress, initBalance, String metadata，那么AccountActivateRequest的结构如下：

Class AccountActivateRequest {

String sourceAddress;

String destAddress;

int64 initBalance;

String metadata;

}

## 3.2 响应结果

### 3.2.1 查询及验证的方法

该类方法供用户调用的。

响应结果的格式，包含错误码，错误描述和result，格式是[类名][方法名]Response，例如account.create的结构体名是AccountCreateResponse：

class AccountCreateResponse{

int errorCode;

string errorDesc;

AccountCreateResult result;

}

说明：

(1) errorCode: 0表示无错误，大于0表示有错误

(2) errorDesc: 空表示无错误，有内容表示有错误

(3) result: 返回结果的结构体，其中结构体的名称，格式是[类名][方法名]Result，例如account.create的结构体名是AccountCreateResult。

其中结构体的成员是返回值的成员变量名，例如：account.create()的返回值是privateKey, publicKey, address，那么AccountCreateResult的结构如下：

class AccountCreateResult{

string privateKey;

string publicKey;

string address;

}

### 3.2.2 生成交易操作的方法

该类方法不供用户调用，供内部接口transaction.buildBlob()方法调用的。

返回protocol buffer的操作对象Chain.Operation.

# 3. account类

接口包括所有与账户相关的操作，查询和验证接口。

## 3.1 生成交易操作的方法

### 3.1.1 activate

激活账户，指将未写在区块链上的账户记录在区块链上，通过getInfo接口可以查询到的账户。前提是待处理的账户必须是未激活账户。

**(1) 调用方法**

Chain.Operation Activate(AccountActivateOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| destAddress | String | 必填，待激活的目标账户地址 |
| initBalance | int64 | 必填，给目标账户的初始化资产，大小[0.1, max(int64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_DESTADDRESS\_ERROR | 11003 | Invalid destAddress |
| INVALID\_INITBALANCE\_ERROR | 11004 | InitBalance must between 1 and max(int64) |
| SOURCEADDRESS\_EQUAL\_  DESTADDRESS\_ERROR | 11005 | SourceAddress cannot be equal to destAddress |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 3.1.2 setMetadata

设置账户metadata消息

**(1) 调用方法**

Chain.Operation setMetadata (AccountSetMetadataOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| key | String | 必填，metadata的关键词，长度[1, 1024] |
| value | String | 必填，metadata的内容，长度[0, 256000] |
| version | int64 | 选填，metadata的版本 |
| deleteFlag | boolean | 选填，是否删除metadata |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_DATAKEY\_ERROR | 11011 | The length of key must between 1 and 1024 |
| INVALID\_DATAVALUE\_ERROR | 11012 | The length of value must between 0 and 256000 |
| INVALID\_DATAVERSION\_ERROR | 11013 | The version must be equal or bigger than 0 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 3.1.3 setPrivilege

设置账户权限，包括设置账户权重，签名者权重，交易门限，指定交易门限。

**(1) 调用方法**

Chain.Operation setPrivilege (AccountSetPrivilegeOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| masterWeight | String | 选填，账户自身权重，大小[0, max(uint32)] |
| signers | Signer[] | 选填，签名者权重列表 |
| address | String | 选填，签名账户地址 |
| weight | int64 | 选填，签名账户权重，大小[0, max(uint32)] |
| txThreshold | String | 选填，交易门限，大小[0, max(int64)] |
| typeThreshold | TypeThreshold | 选填，指定类型交易门限 |
| type | int32 | 选填，操作类型，大小[1, 100] |
| threshold | int64 | 选填，门限，大小[0, max(int64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_MASTERWEIGHT \_ERROR | 11015 | MasterWeight must between 0 and max(uint32) |
| INVALID\_SIGNER\_ADDRESS\_ERROR | 11016 | Invalid signer address |
| INVALID\_SIGNER\_WEIGHT \_ERROR | 11017 | Signer weight must between 0 and max(uint32) |
| INVALID\_TX\_THRESHOLD\_ERROR | 11018 | TxThreshold must between 0 and max(int64) |
| INVALID\_OPERATION\_TYPE\_ERROR | 11019 | Operation type must between 1 and 100 |
| INVALID\_TYPE\_THRESHOLD\_ERROR | 11020 | TypeThreshold must between 0 and max(int64) |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 3.2 查询及验证方法

### 3.2.1 checkValid

检测账户地址的有效性

**(1) 调用方法**

AccounCheckValidResponse checkValid(AccountCheckValidRequest)

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 必填，账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **result类型** | **说明** |
| isValid | boolean | 是否有效 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| SYSTEM\_ERROR | 20000 | System error |

### 3.2.2 create

创建未激活账户，指未写在区块链上的账户信息，通过getInfo接口无法查询到的账户。

**(1) 调用方法**

AccountCreateResponse create();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| privateKey | String | 账户私钥 |
| publicKey | String | 账户公钥 |
| address | String | 账户地址 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| ACCOUNT\_CREATE\_ERROR | 11001 | Create account failed |

### 3.2.3 getInfo

获取账户信息，包括账户地址，账户余额，账户交易序列号，账户资产和账户权重

**(1) 调用方法**

AccountGetInfoResponse GetInfo(AccountGetInfoRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 待查询的账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 账户地址 |
| balance | int64 | 账户余额 |
| nonce | int64 | 账户交易序列号 |
| priv | Priv | 账户权限 |
| masterWeight | int64 | 账户自身权重 |
| signers | Signer[] | 签名者权重 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重 |
| threshold | Threshold | 门限 |
| txThreshold | int64 | 交易默认门限 |
| typeThresholds | TypeThreshold[] | 不同类型交易的门限 |
| type | int64 | 操作类型 |
| threshold | int64 | 门限 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 3.2.4 getNonce

查询账户交易序列号

**(1) 调用方法**

AccountGetNonceResponse getNonce(AccountGetNonceRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 待查询的账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| nonce | int64 | 该账户的交易序列号 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 3.2.5 getBalance

获取账户余额

**(1) 调用方法**

AccountGetBalanceResponse getBalance(AccountGetBalanceRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 待查询的账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| balance | int64 | 该账户的]余额 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 3.2.6 getAssets

获取账户指定资产数量

**(1) 调用方法**

AccountGetAssets getAssets(AccountGetAssetsRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 必填，待查询的账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| asset | AssetInfo[] | 账户资产 |
| key | Key | 资产惟一标识 |
| code | String | 资产编码 |
| issuer | String | 资产发行账户地址 |
| amount | int64 | 资产数量 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| NO\_ASSET\_ERROR | 11009 | The account does not have the asset |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 3.2.7 getMetadata

获取账户的metadata信息

**(1) 调用方法**

AccountGetMetadataResponse getMetadata(AccountGetMetadataRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 必填，账户地址 |
| key | String | 选填，metadata关键字，长度[1, 1024] |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| metadata | Metadata | 账户 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| NO\_METADATA\_ERROR | 11010 | The account does not have the metadata |
| INVALID\_DATAKEY\_ERROR | 11011 | The length of key must between 1 and 1024 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 4. asset类

遵循ATP1.0协议，接口包括所有与资产相关的操作，查询。

## 4.1 生成交易操作的方法

### 4.1.1 issue

该接口实现资产的发行

**(1) 调用方法**

Chain.Operation issue(AssetIssueOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| code | String | 必填，资产编码，长度[1 1024] |
| assetAmount | int64 | 必填，资产发行数量，大小[0, max(int64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_ASSET\_CODE \_ERROR | 11023 | The length of key must between 1 and 64 |
| INVALID\_ASSET\_AMOUNT\_ERROR | 11024 | AssetAmount must between 0 and max(int64) |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 4.1.2 send

该接口实现资产的转移

**(1) 调用方法**

Chain.Operation send(AssetSendOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| destAddress | String | 必填，目标账户地址 |
| code | String | 必填，资产编码，长度[1 1024] |
| issuer | String | 必填，资产发行账户地址 |
| assetAmount | int64 | 必填，资产数量，大小[ 0, max(int64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_DESTADDRESS\_ERROR | 11003 | Invalid destAddress |
| SOURCEADDRESS\_EQUAL\_  DESTADDRESS\_ERROR | 11005 | SourceAddress cannot be equal to destAddress |
| INVALID\_ASSET\_CODE \_ERROR | 11023 | The length of asset code must between 1 and 64 |
| INVALID\_ASSET\_AMOUNT\_ERROR | 11024 | AssetAmount must between 0 and max(int64) |
| INVALID\_ISSUER\_ADDRESS\_ERROR | 11027 | Invalid issuer address |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 4.2 查询方法

### 4.2.1 getInfo

获取账户指定资产信息

**(1) 调用方法**

AssetGetInfoResponse getInfo(AssetGetInfoRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| address | String | 必填，待查询的账户地址 |
| code | String | 必填，资产编码，长度[1 1024] |
| issuer | String | 必填，资产发行账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| asset | AssetInfo[] | 账户资产 |
| key | Key | 资产惟一标识 |
| code | String | 资产编码 |
| issuer | String | 资产发行账户地址 |
| amount | int64 | 资产数量 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| NO\_ASSET\_ERROR | 11009 | The account does not have this asset |
| INVALID\_ASSET\_CODE \_ERROR | 11023 | The length of asset code must between 1 and 64 |
| INVALID\_ISSUER\_ADDRESS\_ERROR | 11027 | Invalid issuer address |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 5. bu类

接口包括所有与bu相关的操作

## 5.1 生成交易操作的方法

### 5.1.1 send

该接口实现BU资产的发送

**(1) 调用方法**

Chain.Operationsend(BUSendOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| destAddress | String | 必填，目标账户地址 |
| buAmount | int64 | 必填，资产发行数量，大小[0, max(int64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_DESTADDRESS\_ERROR | 11003 | Invalid destAddress |
| SOURCEADDRESS\_EQUAL\_  DESTADDRESS\_ERROR | 11005 | SourceAddress cannot be equal to destAddress |
| INVALID\_BU\_AMOUNT\_ERROR | 11026 | BuAmount must between 0 and max(int64) |
| INVALID\_ISSUER\_ADDRESS\_ERROR | 11027 | Invalid issuer address |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 6. ctp10Token类(不用封装)

ctp10Token指合约token，遵循CTP1.0协议，接口。该类包括所有与合约token相关的操作

## 6.1 生成交易操作的方法

### 6.1.1 issue

该接口用于发行合约token

**(1) 调用方法**

Chain.Operation issue(Ctp10TokenIssueOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| initBalance | int64 | 必填，给合约账户的初始化资产，大小[1, max(64)] |
| name | String | 必填，ctp10Token名称，长度[1, 1024] |
| symbol | String | 必填，ctp10Token符号，长度[1, 1024] |
| decimals | int | 必填，ctp10Token数量的精度，大小[0, 8] |
| totalSupply | String | 必填，ctp10Token发行的总供应量，大小[1, max(int64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_INITBALANCE\_ERROR | 11004 | InitBalance must between 1 and max(int64) |
| INVALID\_TOKEN\_NAME\_ERROR | 11031 | The length of token name must between 1 and 1024 |
| INVALID\_TOKEN\_SYMBOL\_ERROR | 11032 | The length of symbol must between 1 and 1024 |
| INVALID\_TOKEN\_DECIMALS\_ERROR | 11033 | Decimals must between 1 and 8 |
| INVALID\_TOKEN\_TOTALSUPPLY\_ERROR | 11034 | TotalSupply must between 1 and max(int64) |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.1.2 transfer

该接口用于转移合约token

**(1) 调用方法**

Chain.Operation transfer(Ctp10TokenTransferOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，合约token的持有者账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| destAddress | String | 必填，待转移的目标账户地址 |
| tokenAmount | String | 必填，待转移的ctp10Token数量，大小[1, int(64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_DESTADDRESS\_ERROR | 11003 | Invalid destAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| SOURCEADDRESS\_EQUAL\_  DESTADDRESS\_ERROR | 11005 | SourceAddress cannot be equal to destAddress |
| CONTRACTADDRESS\_NOT\_  CONTRACTACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| INVALID\_TOKEN\_AMOUNT\_ERROR | 11039 | Token amount must between 1 and max(int64) |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| METADATA\_NOT\_HEX\_STRING\_ERROR | 11008 | Metadata must be a hex string |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.1.3 transferFrom

该接口用于转移合约token

**(1) 调用方法**

Chain.Operation transferFrom(Ctp10TokenTransferFromOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| fromAddress | String | 必填，待转移的合约token持有者账户地址 |
| destAddress | String | 必填，待转移的目标账户地址 |
| tokenAmount | String | 必填，待转移的ctp10Token数量，大小[1, int(64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| INVALID\_FROMADDRESS\_ERROR | 11041 | Invalid fromAddress |
| FROMADDRESS\_EQUAL\_  DESTADDRESS\_ERROR | 11042 | FromAddress cannot be equal to destAddress |
| INVALID\_TOKEN\_AMOUNT\_ERROR | 11039 | Token amount must between 1 and max(int64) |
| METADATA\_NOT\_HEX\_STRING\_ERROR | 11008 | Metadata must be a hex string |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.1.4 approve

该接口用于授权从交易发送者账户转出指定数量的合约token

**(1) 调用方法**

Chain.Operation approve(Ctp10TokenApproveOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，合约token的持有者账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| spender | String | 必填，授权的账户地址 |
| tokenAmount | String | 必填，被授权的待转移的ctp10Token数量，大小[1, int(64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| INVALID\_SPENDER\_ERROR | 11043 | Invalid spender |
| INVALID\_TOKEN\_AMOUNT\_ERROR | 11039 | Token amount must between 1 and max(int64) |
| METADATA\_NOT\_HEX\_STRING\_ERROR | 11008 | Metadata must be a hex string |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.1.5 assign

该接口用于分配合约token

**(1) 调用方法**

Chain.Operation assign(Ctp10TokenAssignOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，合约token的拥有者账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| destAddress | String | 必填，待分配的目标账户地址 |
| tokenAmount | String | 必填，待分配的ctp10Token数量，大小[1, int(64)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| INVALID\_TOKEN\_AMOUNT\_ERROR | 11039 | Token amount must between 1 and max(int64) |
| METADATA\_NOT\_HEX\_STRING\_ERROR | 11008 | Metadata must be a hex string |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.1.6 changeOwner

该接口用于转移合约token拥有权

**(1) 调用方法**

Chain.Operation changeOwner(Ctp10TokenChangeOwnerOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，合约token的拥有者账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| tokenOwner | String | 必填，待分配的目标账户地址 |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 对应操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| INVALID\_TOKENOWNER\_ERRPR | 11035 | Invalid token owner |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| METADATA\_NOT\_HEX\_STRING\_ERROR | 11008 | Metadata must be a hex string |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 6.2 查询及验证方法

### 6.2.1 checkValid

该接口用于查询该合约token是否合法

**(1) 调用方法**

Ctp10TokenCheckValidResponse checkValid(Ctp10TokenCheckValidRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 必填，合约地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| isValid | boolean | 是否合法 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| INVALID\_TOKENOWNER\_ERROR | 11035 | Invalid token owner |
| INVALID\_SPENDER\_ERROR | 11043 | Invalid spender |
| GET\_ALLOWNANCE\_ERROR | 11036 | Get allowance failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.2 allowance

该接口用于获取spender仍然被允许从owner提取的金额

**(1) 调用方法**

Ctp10TokenAllowanceResponse allowance(Ctp10TokenAllowanceRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 必填，合约账户地址 |
| tokenOwner | String | 必填，合约token持有者账户地址 |
| spender | String | 必填，被授权账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| allowance | String | 允许提取的金额 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| INVALID\_TOKENOWNER\_ERRPR | 11035 | Invalid token owner |
| INVALID\_SPENDER\_ERROR | 11043 | Invalid spender |
| GET\_ALLOWNANCE\_ERROR | 11036 | 底层返回的具体错误 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.3 getInfo

该接口用于获取合约token的信息

**(1) 调用方法**

Ctp10TokenGetInfoResponse getInfo(Ctp10TokenGetInfoRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 合约账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| symbol | String | 合约Token符号 |
| decimals | int | 合约数量的精度 |
| totalSupply | String | 合约的总供应量 |
| name | String | 合约Token的名称 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.4 getName

该接口用于获取合约token的名称

**(1) 调用方法**

Ctp10TokenGetNameResponse getName(Ctp10TokenGetNameRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 合约账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| name | String | 合约Token的名称 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.5 getSymbol

该接口用于获取合约token的符号

**(1) 调用方法**

Ctp10TokenGetSymbolResponse getSymbol (Ctp10TokenGetSymbolRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 合约账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| symbol | String | 合约Token的符号 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.6 getDecimals

该接口用于获取合约token的精度

**(1) 调用方法**

Ctp10TokenGetDecimalsResponse getDecimals (Ctp10TokenGetDecimalsRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** |  |
| contractAddress | String | 合约账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| decimals | String | 合约Token的精度 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.7 getTotalSupply

该接口用于获取合约token的总供应量

**(1) 调用方法**

Ctp10TokenGetTotalSupplyResponse getTotalSupply(Ctp10TokenGetTotalSupplyRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 合约账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| totalSupply | String | 合约Token的总供应量 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 6.2.8 getBalance

该接口用于获取合约token拥有者的账户余额

**(1) 调用方法**

Ctp10TokenGetBalanceResponse getBalance(Ctp10TokenGetBalanceRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 必填，合约账户地址 |
| tokenOwner | String | 必填，合约Token持有者的账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| balance | String | 账户余额 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| INVALID\_TOKENOWNER\_ERRPR | 11035 | Invalid token owner |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 7. contract类

包括所有与合约相关的操作与查询

## 7.1 生成交易操作的方法

### 7.1.1 create

该方法用于合约的创建

**(1) 调用方法**

Chain.Operation create(ContractCreateOperation;

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| initBalance | int64 | 必填，给合约账户的初始化资产，大小[1, max(64)] |
| type | Integer | 选填，合约的语种，默认是0 |
| payload | String | 必填，对应语种的合约代码 |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 对应操作的protobuf对象 |

**(5) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_INITBALANCE\_ERROR | 11004 | InitBalance must between 1 and max(int64) |
| INVALID\_CONTRACT\_TYPE\_ERROR | 11047 | Type must be equal or bigger than 0 |
| PAYLOAD\_EMPTY\_ERROR | 11044 | Payload cannot be empty |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 7.1.2 invokeByAsset

该接口实现资产发行并触发合约，或仅触发合约

**(1) 调用方法**

Chain.Operation invokeByAsset(ContractInvokeByAssetOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| code | String | 选填，资产编码，长度[0, 1024]，当为null时，仅触发合约 |
| issuer | String | 选填，资产发行账户地址，当为null时，仅触发合约 |
| amount | int64 | 选填资产数量，大小[0, max(int64)]，当是0时，仅触发合约 |
| input | String | 选填，待触发的合约的main()入参 |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 对应操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| INVALID\_ASSET\_CODE \_ERROR | 11023 | The length of asset code must between 0 and 64 |
| INVALID\_ASSET\_AMOUNT\_ERROR | 11024 | Amount must between 0 and max(int64) |
| INVALID\_ISSUER\_ADDRESS\_ERROR | 11027 | Invalid issuer address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 7.1.3 invokeByBU

该接口实现BU资产的发送和触发合约，或仅触发合约

**(1) 调用方法**

Chain.Operation send(ContractInvokeByBUOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| contractAddress | String | 必填，合约账户地址 |
| amount | int64 | 选填，资产发行数量，大小[0, max(int64)]，当0时仅触发合约 |
| input | String | 选填，待触发的合约的main()入参 |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 对应操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_  CONTRACTACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| INVALID\_BU\_AMOUNT\_ERROR | 11026 | Amout must between 0 and max(int64) |
| INVALID\_ISSUER\_ADDRESS\_ERROR | 11027 | Invalid issuer address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 7.2 查询及验证方法

### 7.2.1 getInfo

该方法用于查询合约代码

**(1) 调用方法**

ContractGetInfoResponse getInfo (ContractGetInfoRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 必填，合约账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contract | ContractInfo | 合约信息 |
| type | int | 合约类型，默认0 |
| payload | String | 合约代码 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_CONTRACT  ACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| CONNECTNETWORK \_ERROR | 19999 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 7.2.2 getAddress

该方法用于查询合约代码

**(1) 调用方法**

ContractGetAddressResponse getAddress (ContractGetAddressRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| hash | String | 必填，创建合约交易的hash |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 合约地址 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_HASH\_ERROR | 11055 | Invalid transaction hash |
| CONNECTNETWORK \_ERROR | 19999 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 7.2.3 checkValid

检测账户地址的有效性

**(1) 调用方法**

ContractCheckValidResponse checkValid(ContractCheckValidRequest)

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| contractAddress | String | 必填，账户地址 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **result类型** | **说明** |
| isValid | boolean | 是否有效 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

### 7.2.4 call

调试合约代码

**(1) 调用方法**

ContractCallesponse call(ContractCallRequest)

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，合约触发账户地址 |
| contractAddress | String | 选填，合约账户地址，与code不能同时为空 |
| code | String | 选填，合约代码，与contractAddress不能同时为空 |
| input | String | 选填，合约入参 |
| contractBalance | String | 选填，赋予合约的初始 BU 余额 |
| optType | Integer | 必填，0: 调用合约的读写接口 init, 1: 调用合约的读写接口 main, 2 :调用只读接口 query |
| feeLimit | Long | 手续费 |
| gasPrice | Long | 打包费用 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **result类型** | **说明** |
| logs | JSONObject | 日志信息 |
| queryRets | JSONArray | 查询结果集 |
| stat | ContractStat | 合约资源占用信息 |
| applyTime | Long | 接收时间 |
| memoryUsage | Long | 内存占用量 |
| stackUsage | Long | 堆栈占用量 |
| step | Long | 几步完成 |
| txs | TransactionEnvs[] | 交易集 |
| transactionEnv | TransactionEnv | 交易 |
| transaction | TransactionInfo | 交易内容 |
| sourceAddress | String | 交易发起的源账户地址 |
| feeLimit | int64 | 交易费用 |
| gasPrice | int64 | 交易打包费用 |
| nonce | int64 | 交易序列号 |
| operations | Operations[] | 操作列表 |
| type | int | 操作类型 |
| sourceAddress | String | 操作发起源账户地址 |
| metadata | String | 备注 |
| createAccount | OperationCreateAccount | 创建账户操作 |
| destAddress | String | 目标账户地址 |
| contract | Contract | 合约信息 |
| type | Integer | 约的语种，默认不赋值 |
| payload | String | 对应语种的合约代码 |
| priv | Priv | 账户权限 |
| masterWeight | int64 | 账户自身权重 |
| signers | Signer[] | 签名者权重 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重 |
| threshold | Threshold | 门限 |
| txThreshold | int64 | 交易默认门限 |
| typeThresholds | TypeThreshold[] | 不同类型交易的门限 |
| type | int64 | 操作类型 |
| threshold | int64 | 门限 |
| metadata | Metadata[] | 账户 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |
| initBalance | int64 | 账户资产 |
| initInput | String | 合约init函数的入参 |
| issueAsset | OperationIssueAsset | 发行资产操作 |
| code | String | 资产编码 |
| amount | int64 | 资产数量 |
| payAsset | OperationPayAsset | 转移资产操作 |
| destAddress | String | 待转移的目标账户地址 |
| asset | Asset | 账户资产 |
| key | Key | 资产惟一标识 |
| code | String | 资产编码 |
| issuer | String | 资产发行账户地址 |
| amount | int64 | 资产数量 |
| input | String | 合约main函数入参 |
| payCoin | OperationPayCoin | 发送BU操作 |
| destAddress | String | 待转移的目标账户地址 |
| amount | int64 | 待转移的BU数量 |
| input | String | 合约main函数入参 |
| setMetadata | OperationSetMetadata | 设置metadata操作 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |
| deleteFlag | boolean | 是否删除metadata |
| setPrivilege | OperationSetPrivilege | 设置账户权限操作 |
| masterWeight | String | 账户自身权重，大小[0, max(uint32)] |
| signers | Signer[] | 签名者权重列表 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重，大小[0, max(uint32)] |
| txThreshold | String | 交易门限，大小[0, max(int64)] |
| typeThreshold | TypeThreshold | 指定类型交易门限 |
| type | int64 | 操作类型，大小[1, 9] |
| threshold | int64 | 门限，大小[0, max(int64)] |
| log | OperationLog | 记录日志 |
| topic | String | 日志主题 |
| data | String[] | 日志内容 |
| trigger | ContractTrigger | 合约触发者 |
| transaction | TriggerTransaction | 触发交易 |
| hash | String | 交易hash |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_CODE\_BOTH  \_NULL\_ERROR | 11063 | ContractAddress and code cannot be empty at the same time |
| INVALID\_OPTTYPE\_ERROR | 11064 | OptType must between 0 and 2 |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 8. log类

包括日志的创建，将日志写到链上

## 8.1 生成交易操作的方法

### 8.1.1 create

在区块链上写日志信息

**(1) 调用方法**

Chain.Operation create (LogCreateOperation);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 选填，发起该操作的源账户地址 |
| topic | String | 必填，日志主题，长度[1, 128] |
| data | String[] | 必填，日志内容，每个字符串长度[1, 1024] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | Chain.Operation | 操作的protobuf对象 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_LOG\_TOPIC \_ERROR | 11045 | The length of key must between 1 and 128 |
| INVALID\_LOG\_DATA \_ERROR | 11046 | The length of value must between 1 and 1024 |
| INVALID\_FEELIMIT\_ERROR | 11050 | FeeLimit must between 1 and max(int64) |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 9. transaction类

包括所有与交易相关的操作与查询

## 9.2 buildBlob

该接口实现交易blob的生成

**(1) 调用方法**

TransactionBuildBlobResponse buildBlobObject(TransactionBuildBlobRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 必填，发起该操作的源账户地址 |
| nonce | int64 | 必填，待发起的交易序列号，函数里+1，大小[1, max(int64)] |
| gasPrice | int64 | 必填，交易打包费用，单位MO，1 BU = 10^8 MO，大小[1000, max(int64)] |
| feeLimit | int64 | 必填，交易手续费，单位MO，1 BU = 10^8 MO，大小[1000000, max(int64)] |
| operation[] | OperationBase的某个子类 | 必填，待提交的操作列表，不能为空 |
| ceilLedgerSeq | long | 选填，区块高度限制，大于等于0，是0时不限制 |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| transactionBlob | String | Transaction序列化后的16进制字符串 |
| hash | String | 交易hash |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_NONCE \_ERROR | 11048 | Nonce must between 1 and max(int64) |
| INVALID\_ GASPRICE\_ERROR | 11049 | Amount must between gasPrice in block and max(int64) |
| INVALID\_FEELIMIT \_ERROR | 11050 | FeeLimit must between 1000000 and max(int64) |
| INVALID\_OPERATION\_ERROR | 11051 | Operation cannot be resolved |
| INVALID\_CEILLEDGERSEQ\_ERROR | 11052 | CeilLedgerSeq must be equal or bigger than 0 |
| SYSTEM\_ERROR | 20000 | System error |

## 9.3 evaluationFee

该接口实现交易的费用评估

**(1) 调用方法**

TransactionEvaluationFeeResponse evaluateFee (TransactionEvaluationFeeRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| sourceAddress | String | 必填，发起该操作的源账户地址 |
| nonce | int64 | 必填，待发起的交易序列号，大小[1, max(int64)] |
| operation | OperationBase[] | 必填，待提交的操作列表，不能为空 |
| signtureNumber | int32 | 选填，待签名者的数量，默认是1，大小[1, max(int32)] |
| metadata | String | 选填，备注 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| txs | TestTx[] | 评估交易集 |
| testTransactionFees | TestTransactionFees | 评估交易费用 |
| transactionFees | TransactionFees | 交易费用 |
| gasPrice | int64 | 打包费用 |
| feeLimit | int64 | 交易费用 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_NONCE \_ERROR | 11045 | Nonce must between 1 and max(int64) |
| INVALID\_OPERATION\_ERROR | 11051 | Operation cannot be resolved |
| OPERATIONS\_ONE\_ERROR | 11053 | One of operations error |
| INVALID\_SIGNATURENUMBER\_ERROR | 11054 | SignagureNumber must between 1 and max(int32) |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 9.4 sign

该接口实现交易的签名

**(1) 调用方法**

TransactionSignResponse signObject (TransactionSignRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blob | String | 必填，发起该操作的源账户地址 |
| privateKeys | String[] | 必填，私钥列表 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| signatures | Signature[] | 签名后的数据列表 |
| signData | int64 | 签名后数据 |
| publicKey | int64 | 公钥 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOB\_ERROR | 11056 | Invalid blob |
| PRIVATEKEY\_NULL\_ERROR | 11057 | PrivateKeys cannot be empty |
| PRIVATEKEY\_ONE\_ERROR | 11058 | One of privateKeys error |
| SYSTEM\_ERROR | 20000 | System error |

## 9.5 submit

该接口实现交易的提交，只有提交交易中有失败的，就抛出Exception，提示有交易失败，具体哪个交易失败，遍历返回数组中TransactionResult的错误码和错误描述。

**(1) 调用方法**

TransactionSubmitResponse submitObject(TransactionSubmitRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blob | String | 必填，交易blob |
| signature | Signature[] | 必填，签名列表 |
| signData | int64 | 必填，签名后的数据 |
| publicKey | int64 | 必填，公钥 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| hash | String | 交易hash |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOB\_ERROR | 11052 | Invalid blob |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 9.6 getInfo

该接口实现根据交易hash查询交易

**(1) 调用方法**

TransactionGetInfoResponse getInfo (TransactionGetInfoRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| hash | String | 交易hash |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| totalCount | int64 | 返回的总交易数 |
| transactions | TransactionHistory[] | 交易内容 |
| actualFee | String | 交易实际费用 |
| closeTime | int64 | 交易关闭时间 |
| errorCode | int64 | 交易错误码 |
| errorDesc | String | 交易描述 |
| hash | String | 交易hash |
| ledgerSeq | int64 | 区块序列号 |
| transaction | TransactionInfo | 交易内容列表 |
| sourceAddress | String | 交易发起的源账户地址 |
| feeLimit | int64 | 交易费用 |
| gasPrice | int64 | 交易打包费用 |
| nonce | int64 | 交易序列号 |
| metadata | String | 备注，读取出来是16字符串，需要反转换为用户的字符串。 |
| operations | Operations[] | 操作列表 |
| type | int | 操作类型 |
| sourceAddress | String | 操作发起源账户地址 |
| metadata | String | 备注 |
| createAccount | OperationCreateAccount | 创建账户操作 |
| destAddress | String | 目标账户地址 |
| contract | Contract | 合约信息 |
| type | ContractType | 约的语种，默认不赋值 |
| payload | String | 对应语种的合约代码 |
| priv | Priv | 账户权限 |
| masterWeight | int64 | 账户自身权重 |
| signers | Signer[] | 签名者权重 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重 |
| threshold | Threshold | 门限 |
| txThreshold | int64 | 交易默认门限 |
| typeThresholds | TypeThreshold[] | 不同类型交易的门限 |
| type | int64 | 操作类型 |
| threshold | int64 | 门限 |
| metadata | Metadata[] | 账户 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |
| initBalance | int64 | 账户资产 |
| initInput | String | 合约init函数的入参 |
| issueAsset | OperationIssueAsset | 发行资产操作 |
| code | String | 资产编码 |
| amount | int64 | 资产数量 |
| payAsset | OperationPayAsset | 转移资产操作 |
| destAddress | String | 待转移的目标账户地址 |
| asset | Asset | 账户资产 |
| key | Key | 资产惟一标识 |
| code | String | 资产编码 |
| issuer | String | 资产发行账户地址 |
| amount | int64 | 资产数量 |
| input | String | 合约main函数入参 |
| payCoin | OperationPayCoin | 发送BU操作 |
| destAddress | String | 待转移的目标账户地址 |
| amount | int64 | 待转移的BU数量 |
| input | String | 合约main函数入参 |
| setMetadata | OperationSetMetadata | 设置metadata操作 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |
| deleteFlag | boolean | 是否删除metadata |
| setPrivilege | OperationSetPrivilege | 设置账户权限操作 |
| masterWeight | String | 账户自身权重，大小[0, max(uint32)] |
| signers | Signer[] | 签名者权重列表 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重，大小[0, max(uint32)] |
| txThreshold | String | 交易门限，大小[0, max(int64)] |
| typeThreshold | TypeThreshold | 指定类型交易门限 |
| type | int64 | 操作类型，大小[1, 9] |
| threshold | int64 | 门限，大小[0, max(int64)] |
| log | OperationLog | 记录日志 |
| topic | String | 日志主题 |
| data | String[] | 日志内容 |
| signatures | Signature[] | 签名列表 |
| signData | int64 | 签名后数据 |
| publicKey | int64 | 公钥 |
| txSize | int64 | 交易大小 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_HASH\_ERROR | 11055 | Invalid transaction hash |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 10. block类

包括所有与区块相关的操作与查询

## 10.1 getNumber

查询最新的区块高度

**(1) 调用方法**

BlockGetNumberResponse getNumber();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| header | BlockHeader | 区块头 |
| blockNumber | int64 | 最新的区块高度，对应底层字段seq |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.2 checkStatus

检查本地节点区块是否同步完成

**(1) 调用方法**

BlockCheckStatusResponse checkStatus();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| isSynchronous | boolean | 区块是否同步 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.3 getTransactions

查询指定区块高度下的所有交易

**(1) 调用方法**

BlockGetTransactionsResponse getTransactions (BlockGetTransactionRequest);

**(3) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockNumber | int64 | 区块高度 |

**(4) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| totalCount | int64 | 返回的总交易数 |
| transactionHistory | TransactionHistory[] | 交易内容 |
| actualFee | String | 交易实际费用 |
| closeTime | int64 | 交易关闭时间 |
| errorCode | int64 | 交易错误码 |
| errorDesc | String | 交易描述 |
| hash | String | 交易hash |
| ledgerSeq | int64 | 区块序列号 |
| transactions | TransactionInfo[] | 交易内容列表 |
| sourceAddress | String | 交易发起的源账户地址 |
| feeLimit | int64 | 交易费用 |
| gasPrice | int64 | 交易打包费用 |
| nonce | int64 | 交易序列号 |
| operations | Operations[] | 操作列表 |
| type | int | 操作类型 |
| sourceAddress | String | 操作发起源账户地址 |
| metadata | String | 备注 |
| createAccount | OperationCreateAccount | 创建账户操作 |
| destAddress | String | 目标账户地址 |
| contract | Contract | 合约信息 |
| type | ContractType | 约的语种，默认不赋值 |
| payload | String | 对应语种的合约代码 |
| priv | Priv | 账户权限 |
| masterWeight | int64 | 账户自身权重 |
| signers | Signer[] | 签名者权重 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重 |
| threshold | Threshold | 门限 |
| txThreshold | int64 | 交易默认门限 |
| typeThresholds | TypeThreshold[] | 不同类型交易的门限 |
| type | int64 | 操作类型 |
| threshold | int64 | 门限 |
| metadata | Metadata[] | 账户 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |
| initBalance | int64 | 账户资产 |
| initInput | String | 合约init函数的入参 |
| issueAsset | OperationIssueAsset | 发行资产操作 |
| code | String | 资产编码 |
| amount | int64 | 资产数量 |
| payAsset | OperationPayAsset | 转移资产操作 |
| destAddress | String | 待转移的目标账户地址 |
| asset | Asset | 账户资产 |
| key | Key | 资产惟一标识 |
| code | String | 资产编码 |
| issuer | String | 资产发行账户地址 |
| amount | int64 | 资产数量 |
| input | String | 合约main函数入参 |
| payCoin | OperationPayCoin | 发送BU操作 |
| destAddress | String | 待转移的目标账户地址 |
| amount | int64 | 待转移的BU数量 |
| input | String | 合约main函数入参 |
| setMetadata | OperationSetMetadata | 设置metadata操作 |
| key | String | metadata的关键词 |
| value | String | metadata的内容 |
| version | int64 | metadata的版本 |
| deleteFlag | boolean | 是否删除metadata |
| setPrivilege | OperationSetPrivilege | 设置账户权限操作 |
| masterWeight | String | 账户自身权重，大小[0, max(uint32)] |
| signers | Signer[] | 签名者权重列表 |
| address | String | 签名账户地址 |
| weight | int64 | 签名账户权重，大小[0, max(uint32)] |
| txThreshold | String | 交易门限，大小[0, max(int64)] |
| typeThreshold | TypeThreshold | 指定类型交易门限 |
| type | int64 | 操作类型，大小[1, 9] |
| threshold | int64 | 门限，大小[0, max(int64)] |
| log | OperationLog | 记录日志 |
| topic | String | 日志主题 |
| data | String[] | 日志内容 |
| signatures | Signature[] | 签名列表 |
| signData | int64 | 签名后数据 |
| publicKey | int64 | 公钥 |
| txSize | int64 | 交易大小 |

**(5) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOCKNUMBER\_ERROR | 11060 | BlockNumber must bigger than 0 |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.4 getInfo

获取区块信息

**(1) 调用方法**

BlockGetInfoResponse getInfo(BlockGetInfoRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockNumber | int64 | 待查询的区块高度 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| closeTime | int64 | 区块关闭时间 |
| number | int64 | 区块高度 |
| txCount | int64 | 交易总量 |
| version | String | 区块版本 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOCKNUMBER\_ERROR | 11060 | BlockNumber must bigger than 0 |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.5 getLatestInfo

获取最新区块信息

**(1) 调用方法**

BlockGetLatestInfoResponse getLatestInfo();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| closeTime | int64 | 区块关闭时间 |
| number | int64 | 区块高度，对应底层字段seq |
| txCount | int64 | 交易总量 |
| version | String | 区块版本 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.6 getValidators

获取指定区块中所有验证节点数

**(1) 调用方法**

BlockGetValidatorsResponse getValidators(BlockGetValidatorsRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockNumber | int64 | 待查询的区块高度 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| validators | ValidatorInfo[] | 验证节点列表 |
| address | String | 共识节点地址 |
| plegeCoinAmount | int64 | 验证节点押金 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOCKNUMBER\_ERROR | 11060 | BlockNumber must bigger than 0 |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.7 getLatestValidators

获取最新区块中所有验证节点数

**(1) 调用方法**

BlockGetLatestValidatorsResponse getLatestValidators();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| validators | Validator[] | 验证节点列表 |
| address | String | 共识节点地址 |
| plegeCoinAmount | int64 | 验证节点押金 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.8 getReward

获取指定区块中的区块奖励和验证节点奖励

**(1) 调用方法**

BlockGetRewardResponse getReward(BlockGetRewardRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockNumber | int64 | 待查询的区块高度 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockReward | int64 | 区块奖励数 |
| validatorsReward | ValidatorReward[] | 验证节点奖励情况 |
| validator | String | 验证节点地址 |
| reward | int64 | 验证节点奖励 |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOCKNUMBER\_ERROR | 11060 | BlockNumber must bigger than 0 |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.9 getLatestReward

获取最新区块中的区块奖励和验证节点奖励

**(1) 调用方法**

BlockGetLatestRewardResponse getLatestReward();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockReward | int64 | 区块奖励数 |
| validatorsReward | ValidatorReward[] | 验证节点奖励情况 |
| validator | String | 验证节点地址 |
| reward | int64 | 验证节点奖励 |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.10 getFees

获取指定区块中的账户最低资产限制和打包费用

**(1) 调用方法**

BlockGetFeesResponse getFees(BlockGetFeesRequest);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| blockNumber | int64 | 待查询的区块高度 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| fees | Fees | 费用 |
| baseReserve | int64 | 账户最低资产限制 |
| gasPrice | int64 | 打包费用，单位MO，1 BU = 10^8 MO |

**(4) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| INVALID\_BLOCKNUMBER\_ERROR | 11060 | BlockNumber must bigger than 0 |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

## 10.11 getLatestFees

获取最新区块中的账户最低资产限制和打包费用

**(1) 调用方法**

BlockGetLatestFeesResponse getLatestFees();

**(2) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| fees | Fees | 费用 |
| baseReservce | int64 | 账户最低资产限制 |
| gasPrice | int64 | 打包费用，单位MO，1 BU = 10^8 MO |

**(3) 错误码**

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| CONNECTNETWORK \_ERROR | 11007 | Connect network failed |
| SYSTEM\_ERROR | 20000 | System error |

以及底层返回的错误码及错误描述

# 11. sdk

整合所有的类接口，这里仅提供初始化操作

## 11.1 getInstance

单例生成SDK对象，并初始化url

**(1) 调用方法**

SDK getInstance(String url);

**(2) 入参**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
| url | String | 待连接的低层节点 |

**(3) 返回值**

|  |  |  |
| --- | --- | --- |
| **变量名** | **类型** | **说明** |
|  | SDK | SDK对象 |

# 12. 错误码

注意：这里可自定义一些错误码，自定义的错误码, java从12001到13000，javascript从13001到14000，golang从14001到15000，ios从15001到16000，php从16001到17000，python从17001到18000，c++从18001到19000

## 12.1 统一错误码

目前java，javascript，go共同的错误码，底层错误码直接返回底层码。

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| ACCOUNT\_CREATE\_ERROR | 11001 | Create account failed |
| INVALID\_SOURCEADDRESS\_ERROR | 11002 | Invalid sourceAddress |
| INVALID\_DESTADDRESS\_ERROR | 11003 | Invalid destAddress |
| INVALID\_INITBALANCE\_ERROR | 11004 | InitBalance must between 1 and max(int64) |
| SOURCEADDRESS\_EQUAL  \_DESTADDRESS\_ERROR | 11005 | SourceAddress cannot be equal to destAddress |
| INVALID\_ADDRESS\_ERROR | 11006 | Invalid address |
| CONNECTNETWORK\_ERROR | 11007 | Connect network failed |
| INVALID\_ISSUE\_AMMOUNT\_ERROR | 11008 | AssetAmount this will be issued must between 1 and max(int64) |
| NO\_ASSET\_ERROR | 11009 | The account does not have the asset |
| NO\_METADATA\_ERROR | 11010 | The account does not have the metadata |
| INVALID\_DATAKEY\_ERROR | 11011 | The length of key must between 1 and 1024 |
| INVALID\_DATAVALUE\_ERROR | 11012 | The length of value must between 0 and 256000 |
| INVALID\_DATAVERSION\_ERROR | 11013 | The version must be equal or bigger than 0 |
| INVALID\_MASTERWEIGHT\_ERROR | 11015 | MasterWeight must between 0 and max(uint32) |
| INVALID\_SIGNER\_ADDRESS\_ERROR | 11016 | Invalid signer address |
| INVALID\_SIGNER\_WEIGHT\_ERROR | 11017 | Signer weight must between 0 and max(uint32) |
| INVALID\_TX\_THRESHOLD\_ERROR | 11018 | TxThreshold must between 0 and max(int64) |
| INVALID\_TYPETHRESHOLD \_TYPE\_ERROR | 11019 | Type of TypeThreshold is invalid |
| INVALID\_TYPE\_THRESHOLD\_ERROR | 11020 | TypeThreshold must between 0 and max(int64) |
| INVALID\_ASSET\_CODE\_ERROR | 11023 | The length of key must between 1 and 64 |
| INVALID\_ASSET\_AMOUNT\_ERROR | 11024 | AssetMount must between 0 and max(int64) |
| INVALID\_BU\_AMOUNT\_ERROR | 11026 | BuAmount must between 0 and max(int64) |
| INVALID\_ISSUER\_ADDRESS\_ERROR | 11027 | Invalid issuer address |
| NO\_SUCH\_TOKEN\_ERROR | 11030 | No such token |
| INVALID\_TOKEN\_NAME\_ERROR | 11031 | The length of token name must between 1 and 1024 |
| INVALID\_TOKEN\_SIMBOL\_ERROR | 11032 | The length of symbol must between 1 and 1024 |
| INVALID\_TOKEN\_DECIMALS\_ERROR | 11033 | Decimals must between 1 and 8 |
| INVALID\_TOKEN\_TOTALSUPPLY\_ERROR | 11034 | TotalSupply must between 1 and max(int64) |
| INVALID\_TOKENOWNER\_ERRPR | 11035 | Invalid token owner |
| INVALID\_CONTRACTADDRESS\_ERROR | 11037 | Invalid contract address |
| CONTRACTADDRESS\_NOT\_  CONTRACTACCOUNT\_ERROR | 11038 | contractAddress is not a contract account |
| INVALID\_TOKEN\_AMOUNT\_ERROR | 11039 | Amount must between 1 and max(int64) |
| SOURCEADDRESS\_EQUAL\_  CONTRACTADDRESS\_ERROR | 11040 | SourceAddress cannot be equal to contractAddress |
| INVALID\_FROMADDRESS\_ERROR | 11041 | Invalid fromAddress |
| FROMADDRESS\_EQUAL\_  DESTADDRESS\_ERROR | 11042 | FromAddress cannot be equal to destAddress |
| INVALID\_SPENDER\_ERROR | 11043 | Invalid spender |
| INVALID\_LOG\_TOPIC\_ERROR | 11045 | The length of log topic must between 1 and 128 |
| INVALID\_LOG\_DATA \_ERROR | 11046 | The length of one of log data must between 1 and 1024 |
| INVALID\_NONCE\_ERROR | 11048 | Nonce must between 1 and max(int64) |
| INVALID\_GASPRICE\_ERROR | 11049 | Amount must between gasPrice in block and max(int64) |
| INVALID\_FEELIMIT\_ERROR | 11050 | FeeLimit must between 1 and max(int64) |
| OPERATIONS\_EMPTY\_ERROR | 11051 | Operations cannot be empty |
| INVALID\_CEILLEDGERSEQ\_ERROR | 11052 | CeilLedgerSeq must be equal or bigger than 0 |
| OPERATIONS\_ONE\_ERROR | 11053 | One of operations cannot be resolved |
| INVALID\_SIGNATURENUMBER\_ERROR | 11054 | SignagureNumber must between 1 and max(int32) |
| INVALID\_HASH\_ERROR | 11055 | Invalid transaction hash |
| INVALID\_BLOB\_ERROR | 11056 | Invalid blob |
| PRIVATEKEY\_NULL\_ERROR | 11057 | PrivateKeys cannot be empty |
| PRIVATEKEY\_ONE\_ERROR | 11058 | One of privateKeys is invalid |
| URL\_EMPTY\_ERROR | 11062 | Url cannot be empty |
| CONTRACTADDRESS\_CODE\_BOTH  \_NULL\_ERROR | 11063 | ContractAddress and code cannot be empty at the same time |
| INVALID\_OPTTYPE\_ERROR | 11064 | OptType must between 0 and 2 |
| GET\_ALLOWANCE\_ERROR | 11065 | 底层错误描述 |
| GET\_TOKEN\_INFO\_ERROR | 11066 | 底层错误描述 |
| CONNECTN\_BLOCKCHAIN\_ERROR | 19999 | Connect blockchain failed |
| SYSTEM\_ERROR | 20000 | System error |

## 12.2 Java错误码

|  |  |  |
| --- | --- | --- |
| **异常** | **错误码** | **错误信息** |
| REQUEST\_NULL\_ERROR | 12001 | Request parameter cannot be null |

## 12.3 Javascript错误码

## 12.4 go错误码