**创智链区块链客户端软件**

**用户手册**

**北京新创智链科技有限公司**

目录

[1、引言 3](#_Toc30103702)

[1.1编写目的 3](#_Toc30103703)

[1.2概念解释 3](#_Toc30103704)

[2、gic客户端 4](#_Toc30103705)

[2.1使用方法 5](#_Toc30103706)

[2.2 命令详解 7](#_Toc30103707)

[2.2.1 registerOrg 7](#_Toc30103708)

[2.2.2 setOrgSigners 8](#_Toc30103709)

[2.2.3 setOrgDeployer 9](#_Toc30103710)

[2.2.4 deployContract 10](#_Toc30103711)

[2.2.5 registerToken 11](#_Toc30103712)

[2.2.6 transfer 13](#_Toc30103713)

[2.2.7 call 14](#_Toc30103714)

[2.2.8 contractInfo 15](#_Toc30103715)

[2.2.9 blockHeight 18](#_Toc30103716)

[2.2.10 block 18](#_Toc30103717)

[2.2.11 transaction 21](#_Toc30103718)

[2.2.12 balance 26](#_Toc30103719)

[2.2.13 nonce 27](#_Toc30103720)

[2.2.14 commitTx 27](#_Toc30103721)

[2.2.15 version 28](#_Toc30103722)

[2.2.16 runAsRPCService 29](#_Toc30103723)

[2.2.17 query 29](#_Toc30103724)

[2.2.18 solDeploy 30](#_Toc30103725)

[2.2.19 solCall 32](#_Toc30103726)

[2.3接口详解 33](#_Toc30103727)

[2.3.1 registerOrg 33](#_Toc30103728)

[2.3.2 setOrgSigners 34](#_Toc30103729)

[2.3.3 setOrgDeployer 36](#_Toc30103730)

[2.3.4 deployContract 37](#_Toc30103731)

[2.3.5 registerToken 39](#_Toc30103732)

[2.3.6 transfer 40](#_Toc30103733)

[2.3.7 call 42](#_Toc30103734)

[2.3.8 contractInfo 44](#_Toc30103735)

[2.3.9 blockHeight 46](#_Toc30103736)

[2.3.10 block 47](#_Toc30103737)

[2.3.11 transaction 50](#_Toc30103738)

[2.3.12 balance 55](#_Toc30103739)

[2.3.13 nonce 56](#_Toc30103740)

[2.3.14 commitTx 57](#_Toc30103741)

[2.3.15 version 58](#_Toc30103742)

[2.3.16 query 59](#_Toc30103743)

[3、giw客户端 60](#_Toc30103744)

[3.1使用方法 61](#_Toc30103745)

[3.2命令详解 62](#_Toc30103746)

[3.2.1 create 62](#_Toc30103747)

[3.2.2 export 62](#_Toc30103748)

[3.2.3 import 63](#_Toc30103749)

[3.2.4 signFile 63](#_Toc30103750)

[3.2.5 signData 64](#_Toc30103751)

# 1、引言

## 1.1编写目的

本文档提供的对象主要是开发，运营用户。本文档对区块链客户端软件主要功能和操作进行介绍，为用户操作、使用区块链客户端软件提供参考和指导。

## 1.2概念解释

区块链客户端软件主要包括gic以及giw工具。

gic客户端主要用于部署智能合约、调用智能合约、查询区块信息等操作。

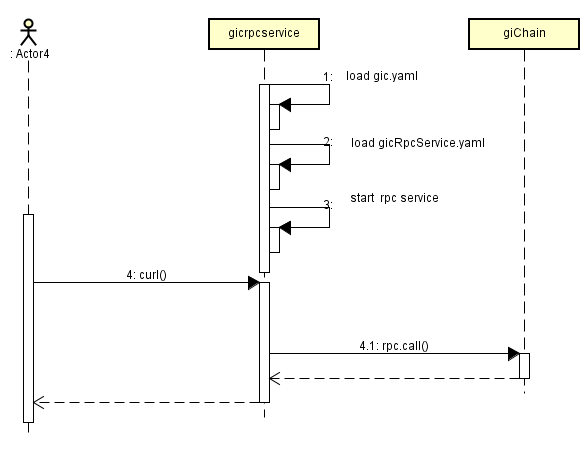
giw 客户端主要用于用于创建、导入、导出钱包、对文件进行签名

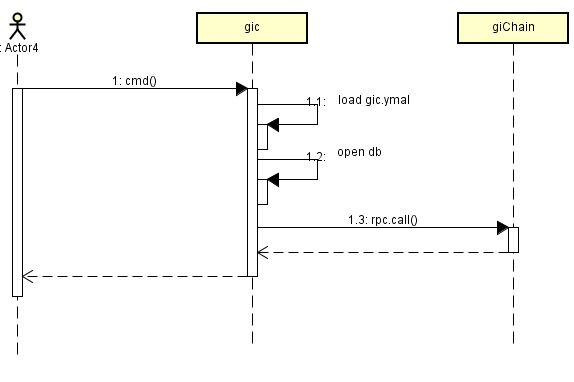
# 2、gic客户端

在使用客户端工具之前需要修改配置文件.config/gic.yaml和.config/gicRpcService.yaml

.keystore目录下存放账户的钱包文件。

以下为gic客户端软件的部分技术设计流程图。



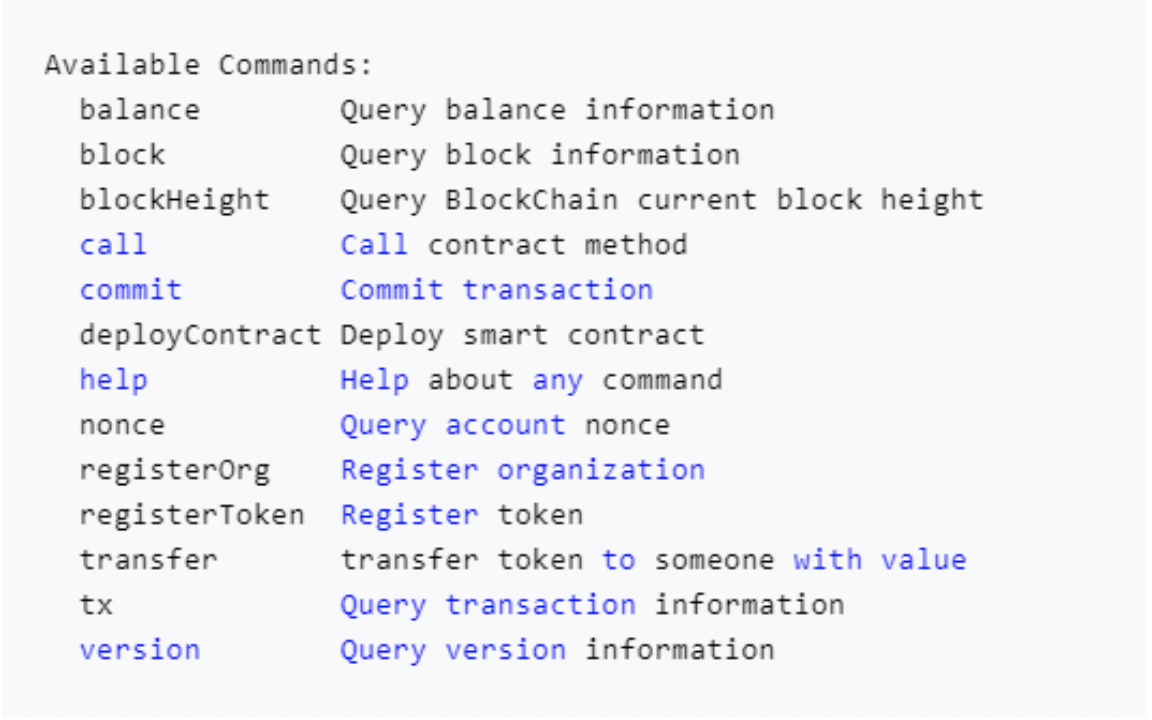


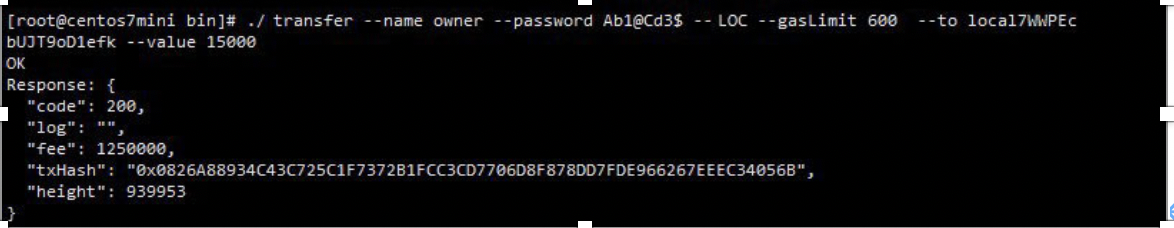
## 2.1使用方法

命令运行格式如下：

Usage:

gic [command]











## 2.2 命令详解

### 2.2.1 registerOrg

功能描述：注册组织

**Command：**

gic registerOrg --name hotwal001 --password aBig62\_123 --orgName example --gasLimit 1000000 [--note ...] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore" |

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"orgID": "orgBtjfCSPCAJ84uQWcpNr74NLMWYm5SXzer",

"height": 234389

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希。 |
| orgID | String | 组织ID。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.2 setOrgSigners

功能描述：设置组织公钥

**Command：**

gic setOrgSigners --name hotwal001 --password aBig62\_123 --orgName example --pubKeys ["0x..."] --gasLimit 1000000 [--note ...] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| gasLimit | String | 交易的燃料限制。 |
| pubKeys | String | 公钥列表。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example**

**OK**

**Response: {**

**"code": 200,**

**"log": "Check tx succeed",**

**"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",**

**"height": 234389**

**}**

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希,以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.3 setOrgDeployer

设置组织下部署合约权限

**Command：**

gic setOrgDeployer --name hotwal001 --password aBig62\_123 --orgName example --deployer local... --gasLimit 1000000 [--note ...] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| gasLimit | String | 交易的燃料限制。 |
| deployer | Address | 发布账号地址。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example**

**OK**

**Response: {**

**"code": 200,**

**"log": "Check tx succeed",**

**"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",**

**"height": 234389**

**}**

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希,以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.4 deployContract

功能描述：部署合约

**Command：**

gic deployContract --name hotwal001 --password aBig62\_123 --contractName dice2win --version 2.0 --orgName genesis --codeFile ./example.tar.gz --effectHeight 100 --owner giES5d6kwoX4vMeNLENMee2Mnsf2KL9ZpWo --gasLimit 10000 [--chainid gi] [--note ...] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| contractName | String | 合约名称。 |
| version | String | 合约版本。 |
| orgName | String | 组织名称。 |
| codeFile | String | 合约源码文件路径。 |
| effectHeight | Uint64 | 合约生效高度。 |
| owner | Address | 所有者地址。 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包文件访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example**

**OK**

**Response: {**

**"code": 200,**

**"log": "Check tx succeed",**

**"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",**

**"smcAddress": "giLVgb3odTfKC9Y9GeFnNWL9wmR4pwWiqwe",**

**"height": 234389**

**}**

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希,以“0x”开头。 |
| smcAddress | Address | 部署的合约的合约地址。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.5 registerToken

功能描述：注册代币

**Command：**

gic registerToken --name hotwal001 --password aBig62\_123 --tokenName xt --tokenSymbol xt --totalSupply 1000000000000 --addSupplyEnabled true --burnEnabled true --gasPrice 2500 --gasLimit 10000 [--note ...] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| tokenName | String | 代币名称。 |
| tokenSymbol | String | 代币符号。 |
| totalSupply | Number | 代币发行总量。 |
| addSupplyEnabled | Bool | 代币是否可增发标志。 |
| burnEnabled | Bool | 代币是否可燃烧标志。 |
| gasPrice | Int64 | 燃料价格。 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore" |

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"tokenAddress": "giLVgb3odTfKC9Y9GeFnNWL9wmR4pwWiqwe",

"height": 234389

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希。，以“0x”开头。 |
| tokenAddress | Address | 代币地址。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.6 transfer

功能描述：转账交易

**Command：**

gic transfer --name hotwal001 --password aBig62\_123 --token gi

--gasLimit 600 [--note hello] --to giLocFJG5Q792eLQXhvNkG417kwiaaoPH5a --value 1500000000 [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| token | String | 交易的资产（本币或代币）对应的名称。 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注（最长256字符），默认为空。 |
| to | Address | 接收转账的账户地址。 |
| value | String | 转账的资产数量（单位：Cong）。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584"

"height": 234389

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.7 call

功能描述：根据组织名、合约名调取合约方法

**Command：**

gic call --name hotwal001 --password aBig62\_123 --orgName example --contract dice2win --method placeBet --gasLimit 5000 [--splitBy @] [--params @ --paramsFile ./params] [--note ...] [--pay 1.003(loc)] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| contract | String | 合约名称。 |
| gasLimit | Int64 | 交易的燃料限制。 |
| note | String | [选填]交易备注，内容自定义。 |
| method | String | 方法名称。 |
| params | String | [与paramsFile二选一]输入参数值列表，规则见输入参数格式规定。 |
| paramsFile | String | [与params二选一]输入参数值列表所在文件。 |
| splitBy | String | [选填]分隔符，默认为"#"。 |
| pay | String | [选填]如遇调用合约方法前需要转账时，可使用该选项标注需要转账的数量和币种，该选项标注的值只向合约账户转账，默认为空。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

输入参数格式规定

1. 参数顺序必须与方法的输入参数顺序一致并且使用选定的分隔符分隔；
2. 如果输入参数中包含"\"，则需要输入双反斜杠"\\"；
3. 当输入类型为字节/字节切片时，将字节/字节切片转换为0x开头的十六进制字符串后输入；
4. 当输入类型为map时，按json串格式输入，且双引号前加反斜线，如：{\"key1\":\"value1\",\"key2\":\"value2\"}（key和value之间不能有额外的空格，如{"test": 123}则为错误格式）;
5. 如果输入字符串值中含有双引号，则需要在双引号前加反斜线，否则命令行无法正确输入双引号；
6. 如果输入参数过长导致超过命令行长度限制，则按如上规则将输入参数列表写入文件，并使用--file选项标注输入参数文件路径。

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"height": 234389E

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.8 contractInfo

功能描述：查询合约信息

模式：(orgName+contractName=指定合约信息);(orgID+contractName=指定合约信息)  
(contractAddr=指定合约信息);(无参数=打印所有合约地址)

**Command：**

gic contractInfo [--orgName example] [--orgID orgBtjf...] [--contractName mydice2win]

[--contractAddr localKkpg...]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| orgName | String | 组织名称。 |
| contractName | String | 合约名称。 |
| orgID | String | 组织ID |
| contractAddr | Address | 合约地址 |

**Output SUCCESS Example：**

OK

Response:

Version: "1.0"

Name: "mydice2win"

OrgID: "orgBtjfCSPCAJ84uQWcpNr74NLMWYm5SXzer"

Address: "local2H5hxpzfuDqvd2uc8DqhF5sgwe7GyDEGP"

Account: "localKkpgMAtPE9wUvtdY9xnN31zRVP3AaoJwA"

Owner: "localL9BzYNYns5VCRaJgfHEBJLzS1bhpHjx7j"

CodeHash: "61BEDF2AFB9CAGIAF0FC104C9C675DD4A522AA39DE07E696F40E84317711B9A"

EffectHeight: 34

LoseHeight: 0

KeyPrefix: "/orgBtjfCSPCAJ84uQWcpNr74NLMWYm5SXzer/mydice2win"

Token: ""

Interfaces: [

{

"methodId": "d517c92",

"gas": 400,

"prototype": "PlaceBet(bn.Number,int64,int64,[]byte,[]byte,types.Address)"

}

]

Method:

SetSecretSigner(types.PubKey)

Params： 01bd6c29d63f5f32aa33955f26a28459988edea4de517f77372e77db33958e6e

SetSettings(string)

Params： example

SetRecvFeeInfos(string)

Params： example

WithdrawFunds(string,types.Address,bn.Number)

Params： example@localL9BzYNYns5VCRaJgfHEBJLzS1bhpHjx7j@1000000000000

PlaceBet(bn.Number,int64,int64,[]byte,[]byte,types.Address)

Params： 1000000000000@200000@200000@0x01bd6c29d63f5f32aa33955f26a28459988edea4de517f77372e77db33958e6e@0x01bd6c29d63f5f32aa33955f26a28459988edea4de517f77372e77db33958e6e@localL9BzYNYns5VCRaJgfHEBJLzS1bhpHjx7j

SettleBet([]byte)

Params： 01bd6c29d63f5f32aa33955f26a28459988edea4de517f77372e77db33958e6e

RefundBet([]byte)

Params： 01bd6c29d63f5f32aa33955f26a28459988edea4de517f77372e77db33958e6e

PS: If the string is just a string, Example: "example"

If the string is a special string, Example: "recvFeeRatio":[500,500], "recvFeeAddr":["localKrHJUVGAt4R9gcfsBthu3dWJR7bAYq1c8","localNwdwjpDotDDLGiB9pARk1CcSM71bdgTef"]

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| Version | String | 合约所有迭代版本 |
| Name | String | 合约名称 |
| OrgID | String | 组织ID。 |
| Address | types.Address | 合约地址。 |
| Account | types.Address | 合约的账户地址 |
| Owner | types.Address | 合约拥有者的账户地址 |
| CodeHash | types.Hash | 合约代码的哈希。 |
| EffectHeight | int64 | 合约生效的区块高度 |
| LoseEffect | int64 | 合约失效的区块高度 |
| KeyPrefix | String | 合约在状态数据库中KEY值的前缀 |
| Token | types.Address | 合约代币地址 |
| Interfaces | []Method | 合约提供的跨合约调用的方法列表 |
| Method | []Method | 合约对外提供接口的方法列表 |
| SetSecretSigner(types.PubKey) | String | SetSecretSigner方法原型。 |
| SetSettings(string) | String | SetSettings方法原型。 |
| SetRecvFeeInfos(string) | String | SetRecvFeeInfos方法原型。 |
| WithdrawFunds(string, types.Address,bn.Number) | String | WithdrawFunds方法原型。 |
| PlaceBet(bn.Number,int64, int64,[]byte,[]byte,types.Address) | String | PlaceBet方法原型。 |
| SettleBet([]byte) | String | SettleBet方法原型。 |
| RefundBet([]byte) | String | RefundBet方法原型。 |

### 2.2.9 blockHeight

功能描述：查询区块高度

**Command：**

gic blockHeight [--chainid gi]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |

**Output SUCCESS Example：**

OK

Response: {

"lastBlock": 2500

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| lastBlock | Int64 | 最新区块高度。 |

### 2.2.10 block

功能描述：查询指定区块信息

**Command：**

gic block [--height 2500]|[--rt "2019-05-10 09:37:14"] [--chainid gi]

gic block [--height 2500]|[--rt "2019-05-10 09:37:14"] [--num 3] [--chainid gi]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| height | int64 | 指定区块高度。 |
| rt | String | 指定区块时间，采用UTC时间，返回的区块时间大于并且最接近输入时间的区块。 |
| num | int64 | 指定区块数量，为空的话返回一个区块详细信息，否则返回多个区块的简要信息。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |

注：时间和高度是互斥参数；如果高度和时间为空，返回最新区块信息。

**Output SUCCESS Example** **1** （详细信息）

OK

Response: {

"blockHeight": 2500,

"blockHash": "0x476dc319f4d3ad9b8c4c4ab03edcc100bb7c90da",

"parentHash": "0x95880ac3be0a89aa2c82fa5e9dadf95a0e99dc3c",

"chainID": "test",

"validatorHash": "0x771543e3773dfebdcc5f3d4ce42c39c29a543218",

"consensusHash": "0xf66ef1df8ba6dac7a1ecce40cc84e54a1cebc6a5",

"blockTime": "2019-05-10 09:37:14.923456881 +0000 UTC",

"blockSize": 988,

"proposerAddress": "testHs1N4Yudisgo5pqZw8AXE66Taq776H3PK",

"txs": [

{

"txHash": "0x4E456161A6580A1D34D86F1560DCFE6034F5E08FA31D7DCEBCCCC72A0DC73654",

"txTime": "2018-12-27T14:26:19.251820644Z",

"code": 200,

"log": "Deliver tx succeed"

"blockHash": "0x583E820E58D2FD00B1A7D66CDBB6B7C26B207925",

"blockHeight": 2495461,

"from": "giAkTDzHLf5udamub38GdepKe7nek66EHqY",

"nonce": 117510,

"gasLimit": 2500,

"fee": 1500000,

"note":"hello",

"messages": [

{

"smcAddress": "giLVgb3odTfKC9Y9GeFnNWL9wmR4pwWiqwe",

"smcName": "token-basic",

"method": "Transfer(smc.Address,big.Int)smc.Error",

"to": "giKuqW1qdsnD7mRsRooXMEkCBj2s9GLF9pn",

"value": "683000000000"

}

]

}

]

}

**Output SUCCESS Example2** （简要信息）

OK

Response: {

"result": [

{

"blockHeight": 2500,

"blockHash": "0x95880AC3BE0A89AA2C82FA5E9DADF95A0E99DC3C",

"blockTime": "2019-05-10T09:37:14.923456881Z",

},

{

"blockHeight": 2501,

"blockHash": "0x476DC319F4D3AD9B8C4C4AB03EDCC100BB7C90DA",

"blockTime": "2019-05-10T09:39:16.234023422Z",

},

{

"blockHeight": 2502,

"blockHash": "0xCF499509BF839A3083B954F409742C3C9AF93367",

"blockTime": "2019-05-10T09:41:17.544988709Z",

}

]

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| blockHeight | Int64 | 区块高度。 |
| blockHash | HexString | 区块哈希值，以“0x”开头。 |
| parentHash | HexString | 父区块哈希值，以“0x”开头。 |
| chainID | String | 链ID。 |
| validatorsHash | HexString | 验证者列表哈希值，以“0x”开头。 |
| consensusHash | HexString | 共识信息哈希值，以“0x”开头。 |
| blockTime | String | 区块打包时间 |
| blockSize | Int | 当前区块大小。 |
| proposerAddress | Address | 提案人地址。 |
| txs [ | Object Array | 交易列表。 |
| { | Object | 交易参数。 |
| txHash | HexString | 交易哈希值，以“0x”开头。 |
| txTime | String | 交易时间 |
| code | Uint32 | 交易结果码，200表示交易成功，其它值表示失败。 |
| log | String | 交易结果描述 |
| blockHash | HexString | 交易所在区块哈希值，以“0x”开头。 |
| blockHeight | Int64 | 交易所在区块高度。 |
| from | Address | 交易签名人地址。 |
| nonce | Uint64 | 交易签名人交易计数值。 |
| gasLimit | Uint64 | 最大燃料数量。 |
| fee | Uint64 | 交易手续费（单位cong）。 |
| note | String | 备注。 |
| messages [ | Object Array | 消息列表。 |
| smcAddress | Address | 合约地址。 |
| smcName | String | 合约名称。 |
| method | String | 方法原型。 |
| to | Address | 转账目的账户地址，仅当交易是BRC20标准转账时有效。 |
| value | String | 转账金额（单位cong），仅当交易是BRC20标准转账时有效。 |
| } |  |  |
| ] |  |  |
|  |  |  |

### 2.2.11 transaction

功能描述：查询交易信息

**Command：**

gic tx --txHash 0x4E456161... [--chainid gi]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| txHash | HexString | 指定交易哈希值，以“0x”开头。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |

**Output SUCCESS Example：**

OK

Response: {

"txHash": "0xc2d613b000c56b53099b3668664c44f1395406a6e8b8f288fb0e045f72a3c83c",

"txTime": "2019-05-08 09:38:05.521987701 +0000 UTC",

"code": 200,

"log": "",

"blockHash": "0xd4f0257eb95de4c2584013f84acf6ae5311bca91",

"blockHeight": 86,

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"nonce": 11,

"gasLimit": 600,

"fee": 1250000,

"note": "",

"messages": [

{

"smcAddress": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"smcName": "token-basic",

"method": "Transfer(types.Address,bn.Number)",

"to": "local3pw2Lgbb83JV5HpoRhM5eCCYS4jtoxxJx",

"value": "150000000000000"

}

],

"tags": {

"/0/std::fee": {

"Name": "std::fee",

"ContractAddr": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidmFsdWUiOjEyNTAwMDB9",

"ReceiptHash": "18B3A7A65CA224B94C90200EC59029B3E94D8A74D853052E67F8F657BD95FB66",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": null,

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 1250000

}

},

"/1/std::transfer": {

"Name": "std::transfer",

"ContractAddr": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbDNwdzJMZ2JiODNKVjVIcG9SaE01ZUNDWVM0anRveHhKeCIsInZhbHVlIjoxNTAwMDAwMDAwMDAwMDB9",

"ReceiptHash": "BB599A08525D3F8293B3695E1C79B05316BE0C0B5B2FE24F4E67CB5C686A27C8",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "local3pw2Lgbb83JV5HpoRhM5eCCYS4jtoxxJx",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 150000000000000

}

},

"/2/totalFee": {

"Name": "totalFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidmFsdWUiOjEyNTAwMDB9",

"ReceiptHash": "F1359BC1665022CB77DA69FCCFA4AC60FA0FD00CE09CB897830083E76F7E31E6",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": null,

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 1250000

}

},

"/3/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbDhMdFQ4QW9uV2dKOG5NQ0VkQVI1VUdyYlJmVW11b2VpeiIsInZhbHVlIjoyNTAwMDB9",

"ReceiptHash": "D872750193DA93BA345DE11A546CBA0BD6A7AE3F68745C537CECE432A047ADDD",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "local8LtT8AonWgJ8nMCEdAR5UGrbRfUmuoeiz",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 250000

}

},

"/4/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbEVtbUxrNEdpSkNrWk5aY0hoVENYaWhFWm9MSmRENG9RRCIsInZhbHVlIjoyNTAwMDB9",

"ReceiptHash": "21AA7212F6F9GI38C01FDC8C0F2CD82A47068DCD535389F34E7493F9CED9571",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "localEmmLk4GiJCkZNZcHhTCXihEZoLJdD4oQD",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 250000

}

},

"/5/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbEhZRHdSU29ORXlDc1o4S1lhdnlSQlpzR0hEN2lMeGpuWCIsInZhbHVlIjozNzUwMDB9",

"ReceiptHash": "90B62B8DCBA1E351452D0C0FD430E83C84FDF89CC580022C93B0476B3CD86068",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "localHYDwRSoNEyCsZ8KYavyRBZsGHD7iLxjnX",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 375000

}

},

"/6/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbHpWdDNwZmU3NWN1VmR2d3paTGtBU1FzS2MzYmY5M2ZpIiwidmFsdWUiOjM3NTAwMH0=",

"ReceiptHash": "8E95324BA76072616AFE77B99A9CB7BD477BFEFAAFB3C1162DC77689EAA95560",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "localzVt3pfe75cuVdvwzZLkASQsKc3bf93fi",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 375000

}

}

}

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| txHash | HexString | 交易哈希值，以“0x”开头。 |
| txTime | String | 交易时间。 |
| code | Uint32 | 交易结果码，200表示交易成功，其它值表示失败。 |
| log | String | 交易结果描述。 |
| blockHash | HexString | 交易所在区块哈希值，以“0x”开头。 |
| blockHeight | Int64 | 交易所在区块高度。 |
| from | Address | 交易签名人地址。 |
| nonce | Uint64 | 交易签名人交易计数值。 |
| gasLimit | Uint64 | 最大燃料数量。 |
| fee | Uint64 | 交易手续费（单位cong）。 |
| note | String | 备注。 |
| messages | Object Array | 消息列表。 |
| smcAddress | Address | 合约地址。 |
| smcName | String | 合约名称。 |
| method | String | 方法原型。 |
| to | Address | 转账目的账户地址，仅当交易是BRC20标准转账时有效。 |
| value | String | 转账金额（单位cong），仅当交易是BRC20标准转账时有效。 |
| Tags | []string | 收据列表 。 |
| Name | String | 收据名称。 |
| ContractAddr | String | 合约地址。 |
| ReceiptBytes | []byte | 收据数据原型。 |
| ReceiptHash | String | 收据hash。 |
| Receipt | map[string]interface{} | 收据解析结果。 |
| from | String | 交易签名人地址。 |
| to | String | 每笔收据的目的账户地址。 |
| token | String | 代币地址。 |
| value | String | 每笔收据的金额。 |

### 2.2.12 balance

功能描述：查询账户余额

**Command：**

gic balance [--address giAkTD... | --name hotwal-001 --password aBig62\_123] [--token gi] [--all false] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| address | Address | [与钱包信息二选一]账户地址，默认为空。。 |
| name | String | [与账户地址二选一]钱包名称，默认为空。 |
| password | String | [与账户地址二选一]钱包密码，默认为空。 |
| token | String | [选填]代币名称，默认为基础代币。 |
| all | Bool | [选填]是否查询所有资产余额标记，默认为false。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]访问钱包路径，默认为"./.keystore" |

**Output SUCCESS Example：**

OK

Response: [

{

"tokenAddress": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"tokenName": "LOC",

"balance": "119997371250000"

}

]

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| token | String | 代币名称。 |
| tokenAddress | String | 代币地址。 |
| balance | String | 账户余额（单位：Cong）。 |

### 2.2.13 nonce

功能描述：查询nonce值

**Command：**

gic nonce [--address giAkTD... | --name hotwal-001 --password aBig62\_123] [--chainid local] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| address | Address | [与钱包二选一]账户地址。 |
| name | String | [与账户地址二选一]钱包地址。 |
| password | String | [与账户地址二选一]钱包密码 |
| chainid | String | [选填]链ID，默认值以配置文件中的配置为准。 |
| keystorepath | String | [选填]访问钱包路径，默认"./.keystore" |

**Output SUCCESS Example：**

OK

Response: {

"nonce": 5000

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| nonce | Uint64 | 指定地址在区块链上可用的下一个交易计数值。 |

### 2.2.14 commitTx

功能描述：提交交易数据

**Command：**

gic commitTx [--tx gi<tx>.v1.AetboY...] [--file xxxxx] [--chainid gi]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| tx | String | 交易数据。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| file | String | 交易数据文件（文本文件），一行一个交易。 |

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "Deliver tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"height": 234389

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易校验/背书结果代码，200表示成功。 |
| log | String | 对交易校验/背书结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.2.15 version

功能描述：查询版本信息

**Command：**

gic version [--chainid gi]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |

**Output SUCCESS Example：**

OK

Response: {

version: 1.0.7.9636

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| version | String | 当前程序版本号。 |

### 2.2.16 runAsRPCService

功能描述：启动RPC服务

**Command：**

gic runAsRPCService

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| -- | -- | -- |

**Output SUCCESS Example：**

OK

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| -- | -- | -- |

### 2.2.17 query

功能描述：根据指令查询指定信息

**Command：**

gic query [--path /contract/all/0] [--chainid gi]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| path | String | [必填]查询指定信息的key。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |

**Output SUCCESS Example：**

OK

response: {

"code" : 200

"key" : "L2NvbnRyYWN0L2FsbC8w"

"value" : "WyJ0ZXN0NVpYaG8zdUw1WlNCV3BIMmlmbmJzQmFRY3k1NTNnWDhGIiwidGVzdEFSUlZm

na3V4ekE1VlpGaGhvNjUxYkZGZ21ncyIsInRlc3Q2cnlWS1NnZFpCZ2NnUjNhV2djbWRa

WEpLaDFRSnoiLCJ0ZXN0QWlXdXNuc0ZzVVFXa2ZIR1NLNXpoNjNGb3pDUFY1UGlCIiwid

Es2a2pnQ2tBQm5TVFMxTUI4NkNpeEVRV005Nlpic1A5IiwidGVzdFA4ZEMyUHVCcHNVQ

Dhqb0t1RjFFdXpaTFhGbXg1NyJd"

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易校验/背书结果代码，200表示成功。 |
| key | String | base64编码格式的String，由输入的参数（path）编码而成。 |
| value | String | base64编码格式的String，由输出的结果编码而成。 |

### 2.2.18 solDeploy

功能描述：根据指令查询指定信息

**Command：**

gic solDeploy --name hotwal001 --password aBig62\_123 --gasLimit 1000000 [--tokenAddr giaabdf...]|[--tokenName gi] [--abiFile ./test.abi] [--binFile ./test.bin]|[--sourceFile ./test.sol] [--note ...] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| tokenName | String | 合约绑定的代币名称。 |
| tokenAddr | String | [与tokenName二选一]合约绑定的代币名称。 |
| abiFile | Number | 合约abi文件的路径 |
| addSupplyEnabled | Bool | 合约binCode文件的路径 |
| burnEnabled | String | [与binFile二选一]合约源码文件路径 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "",

"fee": 1039902500,

"txHash": "0x0041656D5CA3DC79451B8FAB431902E3EDF1A7913840A2CFDEBAB049A2717288",

"height": 48980,

"data": "localGP7ktCj3HwY14EQQZPXxNTWhkL3G4QvDP"

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |
| data | Address | 合约地址 |

### 2.2.19 solCall

功能描述：根据指令查询指定信息

**Command：**

gic solCall --name hotwal001 --password aBig62\_123 --gasLimit 1000000 --contractAddr giGP7ktCj3HwY1... --method getValue [--paramsArr hello] [--abiFile ./test.abi] [--value 1000] [--note ...] [--chainid gi] [--keystorepath ./.keystore]

**Input Parameters：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| contractAddr | String | 合约地址。 |
| method | String | [与tokenName二选一]合约绑定的代币名称。 |
| abiFile | Number | 合约abi文件的路径 |
| addSupplyEnabled | Bool | 合约参数 |
| burnEnabled | String | 转账的资产数量 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example：**

OK

Response: {

"code": 200,

"log": "",

"fee": 3607500,

"txHash": "0x7B57ECDD0744618881C84B7E8951FD1EC8CFC2743A97082E05A8FDA9A93B1278",

"height": 49092,

"data": "0000000000000000000000000000000000000000000000000000000000000040"

}

**Output SUCCESS Result：**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |
| data | Address | 函数返回值 |

## ****2.3接口详解****

### 2.3.1 registerOrg

**Request URI over HTTPS:**

https://localhost:37658/gic\_registerOrg?name=\_&password=\_&gicParams=\_

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_registerOrg",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"gasLimit": "1000000",

"note": "",

"orgName": "example",

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"orgID": "orgBtjfCSPCAJ84uQWcpNr74NLMWYm5SXzer",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| orgID | String | 组织ID。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.2 setOrgSigners

**Request URI over HTTPS:**

[https://localhost:37658/gic\_setOrgSigners?name=\_&password=\_&gicParams=\_](https://localhost:37658/bcc_setOrgSigners?name=_&password=_&bccParams=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_setOrgSigners",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"gasLimit": "1000000",

"note": "",

"orgName": "example",

"pubKeys": ["0x..."],

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| gasLimit | String | 交易的燃料限制。 |
| pubKeys | String | 公钥列表。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.3 setOrgDeployer

**Request URI over HTTPS:**

[https://localhost:37658/gic\_setOrgDeployer?name=\_&password=\_&gicParams=\_](https://localhost:37658/bcc_setOrgDeployer?name=_&password=_&bccParams=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_setOrgDeployer",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"gasLimit": "1000000",

"note": "",

"orgName": "example",

"deployer": local...,

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| gasLimit | String | 交易的燃料限制。 |
| deployer | Address | 发布账号地址。 |
| note | String | [选填]交易备注。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.4 deployContract

**Request URI over HTTPS:**

[https://localhost:37658/gic\_deployContract?name=\_&password=\_&gicParams=\_](https://localhost:37658/bcc_deployContract?name=_&password=_&bccParams=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_deployContract",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"gasLimit": "100000",

"note": "",

"contractName": "dice2win",

"version": "1.0",

"orgName": "genesis",

"codeFile": "./example.tar.gz",

"effectHeight": 100,

"owner": "giES5d6kwoX4vMeNLENMee2Mnsf2KL9ZpWo",

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| contractName | String | 合约名称。 |
| version | String | 合约版本。 |
| orgName | String | 组织名称。 |
| codeFile | String | 合约源码文件路径。 |
| effectHeight | Uint64 | 合约生效高度。 |
| owner | Address | 所有者地址。 |
| gasLimit | int64 | 交易的燃料限制。 |
| note | String | [选填]交易备注。 |
| chainID | String | [选填]链ID。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"smcAddress": "giLVgb3odTfKC9Y9GeFnNWL9wmR4pwWiqwe",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| smcAddress | Address | 部署的合约的合约地址。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.5 registerToken

**Request URI over HTTPS:**

[https://localhost:37658/gic\_registerToken?name=\_&password=\_&gicParams=\_](https://localhost:37658/bcc_registerToken?name=_&password=_&bccParams=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_registerToken",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"tokenName": "tx",

"tokenSymbol": "tx",

"totalSupply": "1000000000",

"addSupplyEnabled": true,

"burnEnabled": true,

"gasPrice": 2500,

"gasLimit": "100000",

"note": "",

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| tokenName | String | 代币名称。 |
| tokenSymbol | String | 代币符号。 |
| totalSupply | Number | 代币发行总量。 |
| addSupplyEnabled | Bool | 代币是否可增发标志。 |
| burnEnabled | Bool | 代币是否可燃烧标志。 |
| gasPrice | Int64 | 燃料价格。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"tokenAddress": "giLVgb3odTfKC9Y9GeFnNWL9wmR4pwWiqwe",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| tokenAddress | String | 代币地址。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.6 transfer

**Request URI over HTTPS:**

[https://localhost:37658/gic\_transfer?name=\_&password=\_&gicParams=\_](https://localhost:37658/bcc_transfer?name=_&password=_&bccParams=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_transfer",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"token": "gi",

"gasLimit": "1000",

"note": "",

"to": "giLocFJG5Q792eLQXhvNkG417kwiaaoPH5a",

"value": "150000000",

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| token | String | 交易的资产（本币或代币）对应的名称。 |
| gasLimit | String | 交易的燃料限制。 |
| note | String | [选填]交易备注（最长256字符），默认为空。 |
| to | Address | 接收转账的账户地址。 |
| value | String | 转账的资产数量（单位：Cong）。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584"

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.7 call

**Request URI over HTTPS:**

[https://localhost:37658/gic\_call?name=\_&password=\_&gicParams=\_](https://localhost:37658/bcc_call?name=_&password=_&bccParams=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_call",

"params": {

"name": "hotwal001",

"password": "aBig62\_123",

"gicParams": {

"orgName": "example",

"contract": "dice2win",

"method": "placeBet",

"gasLimit": "10000",

"splitBy": ";",

"params": "1,2,100,0x234...,0xjsdj...,giLocF...",

"paramsFile": "./params",

"note": "",

"pay": "1.002gi",

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 钱包密码。 |
| orgName | String | 组织名称。 |
| contract | String | 合约名称。 |
| gasLimit | Int64 | 交易的燃料限制。 |
| note | String | [选填]交易备注，内容自定义。 |
| method | String | 方法名称。 |
| params | String | [与paramsFile二选一]输入参数值列表，规则见输入参数格式规定。 |
| paramsFile | String | [与params二选一]输入参数值列表所在文件。 |
| splitBy | String | [选填]分隔符，默认为";"。 |
| pay | String | [选填]如遇调用合约方法前需要转账时，可使用该选项标注需要转账的数量和币种，该选项标注的值只向合约账户转账，默认为空。 |
| keystorepath | String | [选填]钱包访问路径，默认为"./.keystore"。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Check tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易执行结果代码，200表示成功。 |
| log | String | 对交易执行结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.8 contractInfo

**Request URI over HTTPS:**

[https://localhost:37658/gic\_contractInfo?orgName=\_&contractName=\_&orgID=\_&contractAddr=\_&chainID=\_](https://localhost:37658/bcc_contractInfo?orgName=_&contractName=_&orgID=_&contractAddr=_&chainID=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_contractInfo",

"params": {

"orgName": "jiujiu"

"contractName": "excellencies"

"orgID": "orgCZkw5xz9DYa3h5pJ2CzZSuGHRCj2ot5xq"

"contractAddr": "devtest7rXz2j8Vv1ZnawYMi3vswQ1ra2Lt33LTs"

"chainID": "devtest"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| orgName | String | 组织名称。 |
| contractName | String | 合约名称。 |
| orgID | String | 组织ID |
| contractAddr | Address | 合约地址 |
| chainid | String | 选填]链ID，默认以配置文件中的配置为准。 |

**Output SUCCESS Example:**

{

"jsonrpc": "2.0",

"id": "",

"result": {

"address": "devtest7rXz2j8Vv1ZnawYMi3vswQ1ra2Lt33LTs",

"account": "devtestR2FF1eA9KrdPP3HyD1B8ZoA6Mun272de",

"owner": "devtestLve7gJUkNkNYV8uVs2zW3UBZWEbVThnND",

"name": "excellencies",

"version": "4.0",

"codeHash": "0x1A78E2F6F6A87B1563480EA41B2046D0813F6D929D0A05D748C7D4F85B1F941D",

"effectHeight": 119,

"loseHeight": 0,

"keyPrefix": "/orgCZkw5xz9DYa3h5pJ2CzZSuGHRCj2ot5xq/excellencies",

"methods": [

{

"methodId": "e786362c",

"gas": 5000,

"prototype": "SetManager([]types.Address)"

},

{

"methodId": "d373a935",

"gas": 5000,

"prototype": "SetSecretSigner(types.PubKey)"

},

{

"methodId": "44f1b25c",

"gas": 5000,

"prototype": "SetSettings(string)"

},

{

"methodId": "9fe1c79c",

"gas": 5000,

"prototype": "SetRecFeeInfo(string)"

},

{

"methodId": "ec4a2a7e",

"gas": 500,

"prototype": "PlaceBet(string,int64,[]byte,[]byte,types.Address,types.Address)"

},

{

"methodId": "2316ef3a",

"gas": 500,

"prototype": "SettleBet([]byte,int64)"

},

{

"methodId": "d32a5fe6",

"gas": 500,

"prototype": "WithdrawWin([]byte)"

},

{

"methodId": "ac8151c9",

"gas": 500,

"prototype": "RefundBets([]byte,int64)"

}

],

"interfaces": [],

"mine": null,

"token": "",

"orgID": "orgCZkw5xz9DYa3h5pJ2CzZSuGHRCj2ot5xq",

"chainVersion": 2

}

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| Version | String | 合约所有迭代版本 |
| Name | String | 合约名称 |
| OrgID | String | 组织ID。 |
| Address | types.Address | 合约地址。 |
| Account | types.Address | 合约的账户地址 |
| Owner | types.Address | 合约拥有者的账户地址 |
| CodeHash | types.Hash | 合约代码的哈希。 |
| EffectHeight | int64 | 合约生效的区块高度 |
| LoseEffect | int64 | 合约失效的区块高度 |
| KeyPrefix | String | 合约在状态数据库中KEY值的前缀 |
| Token | types.Address | 合约代币地址 |
| Interfaces | []Method | 合约提供的跨合约调用的方法列表 |
| Method | []Method | 合约对外提供接口的方法列表 |
| SetSecretSigner(types.PubKey) | String | SetSecretSigner方法原型。 |
| SetSettings(string) | String | SetSettings方法原型。 |
| SetRecvFeeInfos(string) | String | SetRecvFeeInfos方法原型。 |
| WithdrawFunds(string, types.Address,bn.Number) | String | WithdrawFunds方法原型。 |
| PlaceBet(bn.Number,int64, int64,[]byte,[]byte,types.Address) | String | PlaceBet方法原型。 |
| SettleBet([]byte) | String | SettleBet方法原型。 |
| RefundBet([]byte) | String | RefundBet方法原型。 |

### 2.3.9 blockHeight

**Request URI over HTTPS:**

[https://localhost:37658/gic\_blockHeight?chainID=\_](https://localhost:37658/bcc_blockHeight?chainID=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_blockHeight",

"params": {

"chainID": "gi"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| -- | -- | -- |

**Output SUCCESS Example:**

OK

Response: {

"lastBlock": 2500

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| lastBlock | Int64 | 最新区块高度。 |

### 2.3.10 block

**Request URI over HTTPS:**

[https://localhost:37658/gic\_block?height=\_&rt=\_&num=\_&chainID=\_](https://localhost:37658/bcc_block?height=_&rt=_&num=_&chainID=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_block",

"params": {

"height": 2500,

"chainID": "gi"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| height | int64 | 指定区块高度。 |
| rt | String | 指定区块时间，采用UTC时间，返回的区块时间大于并且最接近输入时间的区块。 |
| num | int64 | 指定区块数量，为空的话返回一个区块详细信息，否则返回多个区块的简要信息。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |

注：时间和高度是互斥参数；如果高度和时间为空，返回最新区块信息。

**Output SUCCESS Example1** （详细信息结果）

OK

Response: {

"blockHeight": 2500,

"blockHash": "0x476dc319f4d3ad9b8c4c4ab03edcc100bb7c90da",

"parentHash": "0x95880ac3be0a89aa2c82fa5e9dadf95a0e99dc3c",

"chainID": "test",

"validatorHash": "0x771543e3773dfebdcc5f3d4ce42c39c29a543218",

"consensusHash": "0xf66ef1df8ba6dac7a1ecce40cc84e54a1cebc6a5",

"blockTime": "2019-05-10 09:37:14.923456881 +0000 UTC",

"blockSize": 988,

"proposerAddress": "testHs1N4Yudisgo5pqZw8AXE66Taq776H3PK",

"txs": [

{

"txHash": "0x4E456161A6580A1D34D86F1560DCFE6034F5E08FA31D7DCEBCCCC72A0DC73654",

"txTime": "2018-12-27T14:26:19.251820644Z",

"code": 200,

"log": "Deliver tx succeed"

"blockHash": "0x583E820E58D2FD00B1A7D66CDBB6B7C26B207925",

"blockHeight": 2495461,

"from": "giAkTDzHLf5udamub38GdepKe7nek66EHqY",

"nonce": 117510,

"gasLimit": 2500,

"fee": 1500000,

"note":"hello",

"messages": [

{

"smcAddress": "giLVgb3odTfKC9Y9GeFnNWL9wmR4pwWiqwe",

"smcName": "token-basic",

"method": "Transfer(smc.Address,big.Int)smc.Error",

"to": "giKuqW1qdsnD7mRsRooXMEkCBj2s9GLF9pn",

"value": "683000000000"

}

]

}

]

}

**Output SUCCESS Example2** （简要信息结果）

OK

Response: {

[

{

"blockHeight": 2500,

"blockHash": "0x95880AC3BE0A89AA2C82FA5E9DADF95A0E99DC3C",

"blockTime": "2019-05-10T09:37:14.923456881Z",

},

{

"blockHeight": 2501,

"blockHash": "0x476DC319F4D3AD9B8C4C4AB03EDCC100BB7C90DA",

"blockTime": "2019-05-10T09:39:16.234023422Z",

},

{

"blockHeight": 2502,

"blockHash": "0xCF499509BF839A3083B954F409742C3C9AF93367",

"blockTime": "2019-05-10T09:41:17.544988709Z",

}

]

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| blockHeight | Int64 | 区块高度。 |
| blockHash | HexString | 区块哈希值，以“0x”开头。 |
| parentHash | HexString | 父区块哈希值，以“0x”开头。 |
| chainID | String | 链ID。 |
| validatorsHash | HexString | 验证者列表哈希值，以“0x”开头。 |
| consensusHash | HexString | 共识信息哈希值，以“0x”开头。 |
| blockTime | String | 区块打包时间 |
| blockSize | Int | 当前区块大小。 |
| proposerAddress | Address | 提案人地址。 |
| txs [ | Object Array | 交易列表。 |
| { | Object | 交易参数。 |
| txHash | HexString | 交易哈希值，以“0x”开头。 |
| txTime | String | 交易时间 |
| code | Uint32 | 交易结果码，200表示交易成功，其它值表示失败。 |
| log | String | 交易结果描述 |
| blockHash | HexString | 交易所在区块哈希值，以“0x”开头。 |
| blockHeight | Int64 | 交易所在区块高度。 |
| from | Address | 交易签名人地址。 |
| nonce | Uint64 | 交易签名人交易计数值。 |
| gasLimit | Uint64 | 最大燃料数量。 |
| fee | Uint64 | 交易手续费（单位cong）。 |
| note | String | 备注。 |
| messages [ | Object Array | 消息列表。 |
| smcAddress | Address | 合约地址。 |
| smcName | String | 合约名称。 |
| method | String | 方法原型。 |
| to | Address | 转账目的账户地址，仅当交易是BRC20标准转账时有效。 |
| value | String | 转账金额（单位cong），仅当交易是BRC20标准转账时有效。 |
| } |  |  |
| ] |  |  |

### 2.3.11 transaction

**Request URI over HTTPS:**

[https://localhost:37658/gic\_transaction?txHash=\_&chainID=\_](https://localhost:37658/bcc_transaction?txHash=_&chainID=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_transaction",

"params": {

"txHash": "0x4E456161...",

"chainID": "gi"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| txHash | HexString | 指定交易哈希值，以“0x”开头。 |

**Output SUCCESS Example:**

OK

Response: {

"txHash": "0xc2d613b000c56b53099b3668664c44f1395406a6e8b8f288fb0e045f72a3c83c",

"txTime": "2019-05-08 09:38:05.521987701 +0000 UTC",

"code": 200,

"log": "",

"blockHash": "0xd4f0257eb95de4c2584013f84acf6ae5311bca91",

"blockHeight": 86,

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"nonce": 11,

"gasLimit": 600,

"fee": 1250000,

"note": "",

"messages": [

{

"smcAddress": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"smcName": "token-basic",

"method": "Transfer(types.Address,bn.Number)",

"to": "local3pw2Lgbb83JV5HpoRhM5eCCYS4jtoxxJx",

"value": "150000000000000"

}

],

"tags": {

"/0/std::fee": {

"Name": "std::fee",

"ContractAddr": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidmFsdWUiOjEyNTAwMDB9",

"ReceiptHash": "18B3A7A65CA224B94C90200EC59029B3E94D8A74D853052E67F8F657BD95FB66",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": null,

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 1250000

}

},

"/1/std::transfer": {

"Name": "std::transfer",

"ContractAddr": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbDNwdzJMZ2JiODNKVjVIcG9SaE01ZUNDWVM0anRveHhKeCIsInZhbHVlIjoxNTAwMDAwMDAwMDAwMDB9",

"ReceiptHash": "BB599A08525D3F8293B3695E1C79B05316BE0C0B5B2FE24F4E67CB5C686A27C8",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "local3pw2Lgbb83JV5HpoRhM5eCCYS4jtoxxJx",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 150000000000000

}

},

"/2/totalFee": {

"Name": "totalFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidmFsdWUiOjEyNTAwMDB9",

"ReceiptHash": "F1359BC1665022CB77DA69FCCFA4AC60FA0FD00CE09CB897830083E76F7E31E6",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": null,

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 1250000

}

},

"/3/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbDhMdFQ4QW9uV2dKOG5NQ0VkQVI1VUdyYlJmVW11b2VpeiIsInZhbHVlIjoyNTAwMDB9",

"ReceiptHash": "D872750193DA93BA345DE11A546CBA0BD6A7AE3F68745C537CECE432A047ADDD",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "local8LtT8AonWgJ8nMCEdAR5UGrbRfUmuoeiz",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 250000

}

},

"/4/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbEVtbUxrNEdpSkNrWk5aY0hoVENYaWhFWm9MSmRENG9RRCIsInZhbHVlIjoyNTAwMDB9",

"ReceiptHash": "21AA7212F6F9GI38C01FDC8C0F2CD82A47068DCD535389F34E7493F9CED9571",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "localEmmLk4GiJCkZNZcHhTCXihEZoLJdD4oQD",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 250000

}

},

"/5/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbEhZRHdSU29ORXlDc1o4S1lhdnlSQlpzR0hEN2lMeGpuWCIsInZhbHVlIjozNzUwMDB9",

"ReceiptHash": "90B62B8DCBA1E351452D0C0FD430E83C84FDF89CC580022C93B0476B3CD86068",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "localHYDwRSoNEyCsZ8KYavyRBZsGHD7iLxjnX",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 375000

}

},

"/6/transferFee": {

"Name": "transferFee",

"ContractAddr": "",

"ReceiptBytes": "eyJ0b2tlbiI6ImxvY2FsTlZVM0xlWDdTaFZVemFxRk51NUM4R1g4czhUaUxRVHNuIiwiZnJvbSI6ImxvY2FsRVRLN1poOWhOU1ByRUtkbUNnbkhEdEZQYXRjczlXd1ZMIiwidG8iOiJsb2NhbHpWdDNwZmU3NWN1VmR2d3paTGtBU1FzS2MzYmY5M2ZpIiwidmFsdWUiOjM3NTAwMH0=",

"ReceiptHash": "8E95324BA76072616AFE77B99A9CB7BD477BFEFAAFB3C1162DC77689EAA95560",

"Receipt": {

"from": "localETK7Zh9hNSPrEKdmCgnHDtFPatcs9WwVL",

"to": "localzVt3pfe75cuVdvwzZLkASQsKc3bf93fi",

"token": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"value": 375000

}

}

}

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| txHash | HexString | 交易哈希值，以“0x”开头。 |
| txTime | String | 交易时间。 |
| code | Uint32 | 交易结果码，200表示交易成功，其它值表示失败。 |
| log | String | 交易结果描述。 |
| blockHash | HexString | 交易所在区块哈希值，以“0x”开头。 |
| blockHeight | Int64 | 交易所在区块高度。 |
| from | Address | 交易签名人地址。 |
| nonce | Uint64 | 交易签名人交易计数值。 |
| gasLimit | Uint64 | 最大燃料数量。 |
| fee | Uint64 | 交易手续费（单位cong）。 |
| note | String | 备注。 |
| messages | Object Array | 消息列表。 |
| smcAddress | Address | 合约地址。 |
| smcName | String | 合约名称。 |
| method | String | 方法原型。 |
| to | Address | 转账目的账户地址，仅当交易是BRC20标准转账时有效。 |
| value | String | 转账金额（单位cong），仅当交易是BRC20标准转账时有效。 |
| Tags | []string | 收据列表 。 |
| Name | String | 收据名称。 |
| ContractAddr | String | 合约地址。 |
| ReceiptBytes | []byte | 收据数据原型。 |
| ReceiptHash | String | 收据hash。 |
| Receipt | map[string]interface{} | 收据解析结果。 |
| from | String | 交易签名人地址。 |
| to | String | 每笔收据的目的账户地址。 |
| token | String | 代币地址。 |
| value | String | 每笔收据的金额。 |

### 2.3.12 balance

**Request URI over HTTPS:**

[https://localhost:37658/gic\_balance?address=\_&name=\_&password=\_&token=\_&all=\_&chainID=\_&keyStorePath=\_](https://localhost:37658/bcc_balance?address=_&name=_&password=_&token=_&all=_&chainID=_&keyStorePath=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_balance",

"params": {

"address": "giAkTD...",

"name": "hotwal001",

"password": "aBig62\_123",

"token": "gi",

"all": true,

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| address | Address | [与钱包信息二选一]账户地址，默认为空。。 |
| name | String | [与账户地址二选一]钱包名称，默认为空。 |
| password | String | [与账户地址二选一]钱包密码，默认为空。 |
| token | String | [选填]代币名称，默认为基础代币。 |
| all | Bool | [选填]是否查询所有资产余额标记，默认为false。 |
| chainid | String | [选填]链ID，默认以配置文件中的配置为准。 |
| keystorepath | String | [选填]访问钱包路径，默认为"./.keystore" |

**Output SUCCESS Example:**

OK

Response: [

{

"tokenAddress": "localNVU3LeX7ShVUzaqFNu5C8GX8s8TiLQTsn",

"tokenName": "LOC",

"balance": "119997371250000"

}

]

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| token | String | 代币名称。 |
| tokenAddress | String | 代币地址。 |
| balance | String | 账户余额（单位：Cong）。 |

### 2.3.13 nonce

**Request URI over HTTPS:**

[https://localhost:37658/gic\_nonce?address=\_&name=\_&password=\_&chainID=\_&keyStorePath=\_](https://localhost:37658/bcc_nonce?address=_&name=_&password=_&chainID=_&keyStorePath=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_nonce",

"params": {

"address": "giAkTD...",

"name": "hotwal001",

"password": "aBig62\_123",

"chainID": "gi",

"keyStorePath": "./.keystore"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| address | Address | [与钱包二选一]账户地址。 |
| name | String | [与账户地址二选一]钱包地址。 |
| password | String | [与账户地址二选一]钱包密码 |
| chainid | String | [选填]链ID，默认值以配置文件中的配置为准。 |
| keystorepath | String | [选填]访问钱包路径，默认"./.keystore" |

**Output SUCCESS Example:**

OK

Response: {

"nonce": 5000

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| nonce | Uint64 | 指定地址在区块链上可用的下一个交易计数值。 |

### 2.3.14 commitTx

**Request URI over HTTPS:**

[https://localhost:37658/gic\_commitTx?tx=\_&chainID=\_](https://localhost:37658/bcc_commitTx?tx=_&chainID=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_commitTx",

"params": {

"tx": "gi<tx>.v1.AetboY...",

"chainID": "gi"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| tx | String | 交易数据。 |

**Output SUCCESS Example:**

OK

Response: {

"code": 200,

"log": "Deliver tx succeed",

"txHash": "0xA1C960B9D5DB633A6E45B45015A722A2C516B392F93C9BF41F5DAA1197030584",

"height": 234389

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易校验/背书结果代码，200表示成功。 |
| log | String | 对交易校验/背书结果进行的文字描述，当code不等于200时描述具体的错误信息。 |
| txHash | HexString | 交易的哈希，以“0x”开头。 |
| height | Int64 | 交易在哪个高度的区块被确认。 |

### 2.3.15 version

**Request URI over HTTPS:**

[https://localhost:37658/gic\_version?chainID=\_](https://localhost:37658/bcc_version?chainID=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_version",

"params": {

"chainID": "gi"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| -- | -- | -- |

**Output SUCCESS Example:**

OK

Response: {

version: 1.0.7.9636

}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| version | String | 当前程序版本号。 |

### 2.3.16 query

**Request URI over HTTPS:**

[https://localhost:37658/gic\_query?path=\_](https://localhost:37658/bcc_query?path=_)

**Request JSONRPC over HTTPS:**

{

"jsonrpc": "2.0",

"id": "dontcare/anything",

"method": "gic\_query",

"params": {

"path": "/contract/all/0"

}

}

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| path | string | [必填]查询指定信息的key。 |

**Output SUCCESS Example:**

OK

response: {

"code" : 200

"key" : "L2NvbnRyYWN0L2FsbC8w"

"value" : "WyJ0ZXN0NVpYaG8zdUw1WlNCV3BIMmlmbmJzQmFRY3k1NTNnWDhGIiwidGVzdEFSUlZm

na3V4ekE1VlpGaGhvNjUxYkZGZ21ncyIsInRlc3Q2cnlWS1NnZFpCZ2NnUjNhV2djbWRa

WEpLaDFRSnoiLCJ0ZXN0QWlXdXNuc0ZzVVFXa2ZIR1NLNXpoNjNGb3pDUFY1UGlCIiwid

Es2a2pnQ2tBQm5TVFMxTUI4NkNpeEVRV005Nlpic1A5IiwidGVzdFA4ZEMyUHVCcHNVQ

Dhqb0t1RjFFdXpaTFhGbXg1NyJd"

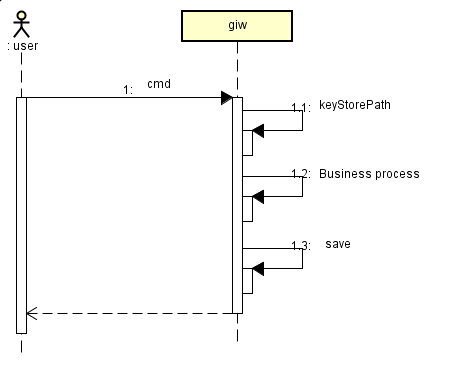
}

**Output SUCCESS Result:**

|  |  |  |
| --- | --- | --- |
| **语法** | **类型** | **注释** |
| code | Int | 交易校验/背书结果代码，200表示成功。 |
| key | String | base64编码格式的String，由输入的参数（path）编码而成。 |
| value | String | base64编码格式的String，由输出的结果编码而成。 |

# 3、giw客户端

以下为giw客户端的部分技术设计图。



## 3.1使用方法

命令运行格式如下：

Usage:

giw [command] [flags]

Available commands:

create create a new wallet

export export a wallet

import import a wallet

signFile sign a file

signData sign raw data

Global flags:

-h, --help help for giw

Use "giw <command> --help" for more information about a command.

## 3.2命令详解

### 3.2.1 create

**command**

giw create --name=hotwal001 [--password=aBig62\_123] [--keystorepath=/wallet/sdfsdfds]

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **选项** | **类型** | **注释** |
| name | String | 钱包名称，长度1-40字符，只能由 大写字母、小写字母、数字、@、\_、-、. `组成 |
| password | String | 可选项，钱包密码，长度8-20字符（可由任意可打印ASCII字符组成，要求至少有一个小写字母、至少一个大写字母、至少一个数字、至少一个非字母数字字符）。  如果命令行不指定密码，程序会提示用户输入两遍密码 |
| keystorepath | String | 可选项，钱包存储路径，如果不指定，系统就采用默认路径 |

**Output SUCCESS Result:**

OK

Pubkey: 0x01bd6c29d63f5f32aa33955f2...ea4de517e77db33958e6e

### 3.2.2 export

**command**

giw export --name=hotwal001 [--password=aBig62\_123] [--keystorepath=/wallet/sdfsdfds]

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **选项** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 可选项，钱包密码，长度8-20字符（可由任意可打印ASCII字符组成，要求至少有一个小写字母、至少一个大写字母、至少一个数字、至少一个非字母数字字符）。 如果命令行不指定密码，程序会提示用户输入密码 |
| keystorepath | String | 可选项，钱包存储路径，如果不指定，系统就采用默认路径 |

**Output SUCCESS Result:**

OK

PrivateKey: 0x832c3477b3a730e9601ed774a...ff7d0333c21721dd33697

PubKey: 0x01bd6c29d63f5f32aa33955f2...ea4de517e77db33958e6e

### 3.2.3 import

**command**

giw import --name=hotwal001 [--password=aBig62\_123] --privatekey=0xad3abc... [--keystorepath=/wallet/sdfsdfds]

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **选项** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 可选项，钱包密码，长度8-20字符（可由任意可打印ASCII字符组成，要求至少有一个小写字母、至少一个大写字母、至少一个数字、至少一个非字母数字字符）。 如果命令行不指定密码，程序会提示用户输入两遍密码。 |
| privatekey | HexString | 导入的私钥明文，可以为32位私钥，也可以是64位私钥+公钥,以“0x”开头 |
| keystorepath | String | 可选项，钱包存储路径，如果不指定，系统就采用默认路径 |

**Output SUCCESS Result:**

OK

PubKey: 0x01bd6c29d63f5f32aa33955f2...ea4de517e77db33958e6e

### 3.2.4 signFile

**command**

giw signFile --name=hotwal001 [--password=aBig62\_123] --file=genesis.json --mode=t [--keystorepath=/wallet/sdfsdfds]

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **选项** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 可选项，钱包密码，长度8-20字符（可由任意可打印ASCII字符组成，要求至少有一个小写字母、至少一个大写字母、至少一个数字、至少一个非字母数字字符）。 如果命令行不指定密码，程序会提示用户输入密码 |
| file | String | 要签名的文件全名 |
| mode | String | 签名模式：  t:文本文件；  b:二进制文件 |
| keystorepath | String | 可选项，钱包存储路径，如果不指定，系统就采用默认路径 |

**Output SUCCESS Result:**

OK

SignatureFile: genesis.json.sig

### 3.2.5 signData

**command**

giw signData --name=hotwal001 [--password=aBig62\_123] --data=abcd12... --file=a.sig [--keystorepath=/wallet/sdfsdfds]

**Input Parameters:**

|  |  |  |
| --- | --- | --- |
| **选项** | **类型** | **注释** |
| name | String | 钱包名称。 |
| password | String | 可选项，钱包密码，长度8-20字符（可由任意可打印ASCII字符组成，要求至少有一个小写字母、至少一个大写字母、至少一个数字、至少一个非字母数字字符）。 如果命令行不指定密码，程序会提示用户输入密码。 |
| data | HexString | 待签名数据的16进制形式 |
| file | String | 签名后输出的文件名 |
| keystorepath | String | 可选项，钱包存储路径，如果不指定，系统就采用默认路径 |

**Output SUCCESS Result:**

OK

SignatureFile: a.sig