EOS开发入门之cleos命令解读(未完待续)

在前面的文章《EOS简介与环境搭建》里面有提到过，EOS有三个主要的组件，分别是keosd、nodeos和cleos。cleos是操作keosd和nodeos的命令行工具，因此了解cleos的命令非常重要。于是，有了这篇内容，对cleos命令进行翻译和解读。

本文是基于2018年6月1日Git克隆下来的EOSIO Dawn4.0上实验的。EOS版本更新太快，可谓一天一版本，才过了周末两天的时间，EOS就已经发布了v1.0.0和v1.0.1两个版本。在每个版本上实验可能会出现不同的结果很正常，所以要稍微留意一下版本问题。

运行前提：假设你已经安装并配置好了EOS的环境。

注意：以下操作均在MacOS上面操作

1、查看cloes全部命令

通过在终端中执行"cloes"命令，可以查看cloes全部命令以及子命令以及对应说明，

```
wenzildeiMac:~ wenzil$ cleos
ERROR: RequiredError: Subcommand required
Command Line Interface to EOSIO Client
Usage: cleos [OPTIONS] SUBCOMMAND

Options:
  -h,--help                   Print this help message and exit
  -u,--url TEXT=http://localhost:8888/
                              the http/https URL where nodeos is running
  --wallet-url TEXT=http://localhost:8900/
                              the http/https URL where keosd is running
  -r,--header                 pass specific HTTP header; repeat this option
to pass multiple headers
  -n,--no-verify              don't verify peer certificate when using HTTPS
  -v,--verbose                output verbose actions on error

Subcommands:
  version                     Retrieve version information
  create                      Create various items, on and off the blockchai
n
  get                         Retrieve various items and information from th
e blockchain
  set                         Set or update blockchain state
  transfer                    Transfer EOS from account to account
  net                         Interact with local p2p network connections
  wallet                      Interact with local wallet
  sign                        Sign a transaction
  push                        Push arbitrary transactions to the blockchain
  multisig                    Multisig contract commands
  system                      Send eosio.system contract action to the block
chain.
```

简单翻译和解读：

```
使用方法：cleos [OPTIONS] SUBCOMMAND
例子："cleos version"命令查看版本信息子命令

[OPTIONS]：为可选参数
-h,--help：打印帮助信息并退出
-u,--url TEXT=http://localhost:8888/：节点组件nodeos运行的URL地址
--wallet-url TEXT=http://localhost:8900/：钱包组件keosd运行的钱包URL地址
-r,--header：传递特定的HTTP请求头，传递多个请求头时重复该参数
-n,--no-verify：使用HTTPS协议的时候不验证节点证书
-v,--verbose：输出详细的错误操作信息

SUBCOMMAND：子命令
version：返回版本信息
create：基于区块链或者非区块链创建不同的项目
get：从区块链中获取不同的项目和信息
set：设置或者更新区块链状态
transfer：账户之间转移EOS
net：与本地P2P网络之间交互
wallet：与本地钱包互动
sign：签署交易
push：向区块链推送任何事物
multisig：对合同进行多重签名操作
system：发送EOS自带的eosio.system合约动作到区块链

#####额外说明#####
在EOS中，action是一个动作，部署智能合约需要将合约绑定到账户，账户和合约交互式是通过action
的，可以单独发送一个action。
Transaction（事务）是一组action，所有的action成功，该Transaction才会成功。
类似于数据库中的事务，要么全部成功，要么全部失败。
```

以下对各个主要子命令进行解读

2、version命令

version目前只有一个子命令"client"，作用是查看客户端的版本信息。

```
wenzildeiMac:~ wenzil$ cleos version client
```

```
Build version: cbf28a23
```

3、create命令

create命令目前主要用于创建账户和密钥。

```
wenzildeiMac:~ wenzil$ cleos create
ERROR: RequiredError: Subcommand required
Create various items, on and off the blockchain
Usage: cleos create SUBCOMMAND

Subcommands:
  key                         Create a new keypair and print the public and
private keys
  account                     Create an account, buy ram, stake for bandwidt
h for the account
```

（1）cleos create key

简单翻译和解读：

```
使用方法：cleos create SUBCOMMAND
例子："cleos create key"创建一对新的密钥，对应一个私钥和一个公钥。
```

操作演示如下：

```
wenzildeiMac:study wenzil$ cleos create key
Private key: 5KcUM9PkRLqRq6KtrGT4ek8dy9mLfPh4cTGR6Pt8fU5tH6Q7y4c
Public key: EOS7caVJ2iDpM7edncLbTGtQqXtTZqyMYxhUP8e1xPzv3mVyL7Gxu
```

（2）cleos create account

```
wenzildeiMac:~ wenzil$ cleos create account
ERROR: RequiredError: creator
Create an account, buy ram, stake for bandwidth for the account
Usage: cleos create account [OPTIONS] creator name OwnerKey [ActiveKey]

Positionals:
  creator TEXT                The name of the account creating the new accou
nt (required)
  name TEXT                   The name of the new account (required)
  OwnerKey TEXT               The owner public key for the new account (requ
ired)
  ActiveKey TEXT                The active public key for the new account

Options:
  -h,--help                     Print this help message and exit
  -x,--expiration               set the time in seconds before a transaction e
xpires, defaults to 30s
  -f,--force-unique             force the transaction to be unique. this will
consume extra bandwidth and remove any protections against accidently issuin
g the same transaction multiple times
  -s,--skip-sign                Specify if unlocked wallet keys should be used
 to sign transaction
  -j,--json                     print result as json
  -d,--dont-broadcast           don't broadcast transaction to the network (ju
st print to stdout)
  -r,--ref-block TEXT           set the reference block num or block id used f
or TAPOS (Transaction as Proof-of-Stake)
  -p,--permission TEXT ...      An account and permission level to authorize,
as in 'account@permission'
  --max-cpu-usage-ms UINT       set an upper limit on the milliseconds of cpu
usage budget, for the execution of the transaction (defaults to 0 which mean
s no limit)
  --max-net-usage UINT          set an upper limit on the net usage budget, in
 bytes, for the transaction (defaults to 0 which means no limit)
```

简单翻译和解读：

```
使用方法：cleos create account [OPTIONS] creator name OwnerKey [ActiveKey]

[OPTIONS]：为可选参数
creator TEXT：创建新账户动作的账户名（必填）
name TEXT：新账户的名字（必填）
OwnerKey TEXT：新帐户的所有者公钥（必填），[作用是掌握该账户所有控制器]
ActiveKey TEXT ：新帐户的有效公钥，[作用是只掌握该账户资金访问权限]

SUBCOMMAND：子命令
-h,--help ：打印帮助信息并退出
-x,--expiration ：设置交易到期之前的秒数，默认为30秒
-f,--force-unique ：确证交易唯一。这将消耗额外的带宽，并消除对多次意外发布相同事务的任何
保护。
-s,--skip-sign ：指定是否应使用解锁的钱包密钥来签署交易
-j,--json ：以JSON格式打印结果
-d,--dont-broadcast ：不向网络广播事务（仅打印标准输出）
-r,--ref-block TEXT：设置用于TAPOS的参考块号或块ID（作为证明的交易）
-p,--permission TEXT ... ：授权的帐户和权限级别，如'account@permission'
--max-cpu-usage-ms UINT：设置交易执行的CPU使用预算的毫秒数上限（默认为0，表示无限制）
```

```
--max-net-usage UINT：设置交易的净使用预算的上限（以字节为单位）（默认为0，表示无限制）

操作演示如下：
wenzildeiMac:study wenzil$ cleos create account eosio user1 EOS6Ps9FM3A6yi5J
gVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra
9EoP6JLTi51zzat
executed transaction: c0b7ff03972c5829a6fb3b8a3d422f5f08b8213c95c1883fd5a641
e5c55c029b  200 bytes  250 us
#         eosio <= eosio::newaccount            {"creator":"eosio","name":"u
ser1","owner":{"threshold":1,"keys":[{"key":"EOS6Ps9FM3A6yi5JgVZ7hPzQU4g...
warning: transaction executed locally, but may not be confirmed by the netwo
rk yet
```

4、get命令

```
wenzildeiMac:~ wenzil$ cleos get
ERROR: RequiredError: Subcommand required
Retrieve various items and information from the blockchain
Usage: cleos get SUBCOMMAND

Subcommands:
  info                        Get current blockchain information
  block                       Retrieve a full block from the blockchain
  account                     Retrieve an account from the blockchain
  code                        Retrieve the code and ABI for an account
  abi                         Retrieve the ABI for an account
  table                       Retrieve the contents of a database table
  currency                    Retrieve information related to standard curre
ncies
  accounts                    Retrieve accounts associated with a public key
  servants                    Retrieve accounts which are servants of a give
n account
  transaction                 Retrieve a transaction from the blockchain
  actions                     Retrieve all actions with specific account nam
e referenced in authorization or receiver
```

简单翻译和解读：

```
使用方法：cleos get SUBCOMMAND

SUBCOMMAND：子命令
info：获取当前的区块链信息
block：从区块链中检索完整的区块
account：从区块链中检索账户
```

```
code：检索账户的代码和ABI
abi：检索账户的ABI
table：检索数据库表的内容
currency：检索与标准货币相关的信息
accounts：检索与公钥相关的帐户
servants：检索指定账户的下属账户
transaction：从区块链中检索交易
actions：检索授权或者接收者中指定账户名的所有操作
```

以下分别对cleos get子命令解读和演示

4.1、get info命令

获取当前的区块链信息

```
wenzildeiMac:~ wenzil$ cleos get info
{
  "server_version": "cbf28a23",
  "chain_id": "a4fe43c897279a677025624555f3835b949f995f87923c97f0392dcff835b
fd0",
  "head_block_num": 35375,
  "last_irreversible_block_num": 35374,
  "last_irreversible_block_id": "00008a2e4c52794e19fa77ae5e1304819ba10a9348c
f241195de2b991ee77066",
  "head_block_id": "00008a2f34b872fbbf17f994cefac9dd7d6e4632b621894ab0e59239
c8e625b8",
  "head_block_time": "2018-06-04T00:08:15",
  "head_block_producer": "eosio",
  "virtual_block_cpu_limit": 200000000,
  "virtual_block_net_limit": 1048576000,
  "block_cpu_limit": 199900,
  "block_net_limit": 1048576
}
```

4.2、get block命令

```
wenzildeiMac:~ wenzil$ cleos get block
ERROR: RequiredError: block
Retrieve a full block from the blockchain
Usage: cleos get block block

Positionals:
  block TEXT                    The number or ID of the block to retrieve (req
uired)
```

简单翻译和解读：

```
使用方法：cleos get block block

Positionals:（参数）
block TEXT：要检索的块的编号或ID（必填）
```

操作演示如下：

根据区块number（也即区块编号）查询

```
wenzildeiMac:~ wenzil$ cleos get block 35375
{
  "timestamp": "2018-06-04T00:08:15.000",
  "producer": "eosio",
  "confirmed": 0,
  "previous": "00008a2e4c52794e19fa77ae5e1304819ba10a9348cf241195de2b991ee77
066",
  "transaction_mroot": "0000000000000000000000000000000000000000000000000000
000000000000",
  "action_mroot": "a48de6a284f75061440fddacd885a87df0b104b2727fec1b1a56082eb
8797d53",
  "schedule_version": 0,
  "new_producers": null,
  "header_extensions": [],
  "producer_signature": "SIG_K1_K5N3V6r6kw9V6euJWyaHMw3KTF2iLgwooQPQH7KhKtbj
NFBNmafyL9ZQB7RW991qp7EVfTkX18j6JVHqwjdPTVQ4b7CmGG",
  "transactions": [],
  "block_extensions": [],
  "id": "00008a2f34b872fbbf17f994cefac9dd7d6e4632b621894ab0e59239c8e625b8",
  "block_num": 35375,
  "ref_block_prefix": 2499352511
}
```

根据区块ID（也即区块哈希值）查询

```
wenzildeiMac:~ wenzil$ cleos get block 00008a2f34b872fbbf17f994cefac9dd7d6e4
632b621894ab0e59239c8e625b8
{
  "timestamp": "2018-06-04T09:08:15.000",
  "producer": "eosio",
  "confirmed": 0,
  "previous": "00008a2e4c52794e19fa77ae5e1304819ba10a9348cf241195de2b991ee77
066",
  "transaction_mroot": "0000000000000000000000000000000000000000000000000000
000000000000",
  "action_mroot": "a48de6a284f75061440fddacd885a87df0b104b2727fec1b1a56082eb
8797d53",
  "schedule_version": 0,
  "new_producers": null,
  "header_extensions": [],
  "producer_signature": "SIG_K1_K5N3V6r6kw9V6euJWyaHMw3KTF2iLgwooQPQH7KhKtbj
NFBNmafyL9ZQB7RW991qp7EVfTkX18j6JVHqwjdPTVQ4b7CmGG",
  "transactions": [],
  "block_extensions": [],
  "id": "00008a2f34b872fbbf17f994cefac9dd7d6e4632b621894ab0e59239c8e625b8",
  "block_num": 35375,
  "ref_block_prefix": 2499352511
}
```

4.4、get account命令

```
wenzildeiMac:~ wenzil$ cleos get account
ERROR: RequiredError: name
Retrieve an account from the blockchain
Usage: cleos get account [OPTIONS] name

Positionals:
  name TEXT                   The name of the account to retrieve (required)

Options:
  -j,--json                   Output in JSON format
```

简单翻译和解读：

```
使用方法：cleos get account [OPTIONS] name

Positionals:（必填参数）
name TEXT：要检索的账户名（必填）

Options:（可选参数）
-j,--json ：以JSON格式打印结果
```

操作演示如下：

```
wenzildeiMac:~ wenzil$ cleos get account user1
privileged: false
permissions:
     owner     1:     1 EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51zzat
```

```
        active     1:    1 EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQCgZ2mqra9EoP6JLTi51
zzat
memory:
     quota:        -1 bytes  used:     33.25 Kb

net bandwidth: (averaged over 3 days)
     used:                -1 bytes
     available:           -1 bytes
     limit:               -1 bytes

cpu bandwidth: (averaged over 3 days)
     used:                -1 us
     available:           -1 us
     limit:               -1 us
```

4.5、get code命令

"get code"命令只有一个参数name（账户名），作用是检索指定账户的代码和ABI，可以使用"cleos get code"命令查看，这里不作详细说明。

操作演示如下：

```
wenzildeiMac:~ wenzil$ cleos get code user1
code hash: c3b697beb6e972e35dcec00c215ef9adde37b0084cb9a43e2f571e3c7aaab4f7
```

4.6、get table命令

```
wenzildeiMac:~ wenzil$ cleos get table
ERROR: RequiredError: contract
Retrieve the contents of a database table
Usage: cleos get table [OPTIONS] contract scope table

Positionals:
  contract TEXT             The contract who owns the table (required)
  scope TEXT                The scope within the contract in which the tab
le is found (required)
  table TEXT                The name of the table as specified by the cont
ract abi (required)

Options:
  -b,--binary UINT          Return the value as BINARY rather than using a
bi to interpret as JSON
  -l,--limit UINT           The maximum number of rows to return
  -k,--key TEXT             The name of the key to index by as defined by
```

```
the abi, defaults to primary key
  -L,--lower TEXT             JSON representation of lower bound value of ke
y, defaults to first
  -U,--upper TEXT             JSON representation of upper bound value value
 of key, defaults to last
```

简单翻译和解读：

```
使用方法：cleos get table [OPTIONS] contract scope table

Positionals：(必填参数)
contract TEXT：拥有表格的合同名（必填）
scope TEXT：找到该表格的合同范围（必填）
table TEXT ：合同abi文件指定的表格名称（必填）

Options：(可选参数)
-b,--binary UINT ：将值以二进制格式返回，而不是解释为JSON
-l,--limit UINT：要返回的最大行数
-k,--key TEXT ：按abi定义的索引键的名称，缺省为主键
-L,--lower TEXT：键的下限值的JSON表示，默认为第一个
-U,--upper TEXT  ：键的上限值的JSON表示，默认为最后一个
```

操作演示如下：

```
wenzildeiMac:~ wenzil$ cleos get table user1 user1 accounts
{
  "rows": [{
      "balance": "80.0000 EOS"
    }
  ],
  "more": false
}

cleos --url http://localhost:8888 get transaction 35fb9cd316955e7004fe25f61c
6c583a663e14e12d0636833f179f980b22f6cd
```

EOS好几个命令都直接报错了，如下：

```
wenzildeiMac:~ wenzil$ cleos get transaction 35fb9cd316955e7004fe25f61c6c583
a663e14e12d0636833f179f980b22f6cd
1264429ms thread-0   main.cpp:2584                 main                 ] Fa
iled with error: Assert Exception (10)
status_code == 200: Error code 404
: {"code":404,"message":"Not Found","error":{"code":0,"name":"exception","wh
at":"unspecified","details":[{"message":"Unknown Endpoint","file":"http_plug
in.cpp","line_number":203,"method":"handle_http_request"}]}}

wenzildeiMac:cleos wenzil$ cleos get servants user1
2798076ms thread-0   main.cpp:2584                 main                 ] Fa
iled with error: Assert Exception (10)
status_code == 200: Error code 404
: {"code":404,"message":"Not Found","error":{"code":0,"name":"exception","wh
at":"unspecified","details":[{"message":"Unknown Endpoint","file":"http_plug
in.cpp","line_number":203,"method":"handle_http_request"}]}}

wenzildeiMac:cleos wenzil$ cleos get accounts EOS6Ps9FM3A6yi5JgVZ7hPzQU4gAgQ
CgZ2mqra9EoP6JLTi51zzat
2394312ms thread-0   main.cpp:2584                 main                 ] Fa
iled with error: Assert Exception (10)
status_code == 200: Error code 404
: {"code":404,"message":"Not Found","error":{"code":0,"name":"exception","wh
at":"unspecified","details":[{"message":"Unknown Endpoint","file":"http_plug
in.cpp","line_number":203,"method":"handle_http_request"}]}}
```

坑啊，写不下去了，待续