

# EOS Token

Hexlant 이진호

# 토큰컨트랙트의 구성

## 1. 장부

각 계정별로, 토큰보유 물량을 기록해 두는 테이블

## 2. 토큰 전송기능

토큰의 전송가능 여부를 확인하고, 장부의 잔액정보를 변경해 주는 기능

## 3. 토큰 정책 제어

토큰의 발행량, 기능에 대한 권한 관리 필요한 규칙을 프로그램

# 이더리움 ERC-20 Token

<https://github.com/ethereum/eips/issues/20>

Ethereum Request for Comment

토큰 요청에 대한 표준 Interface

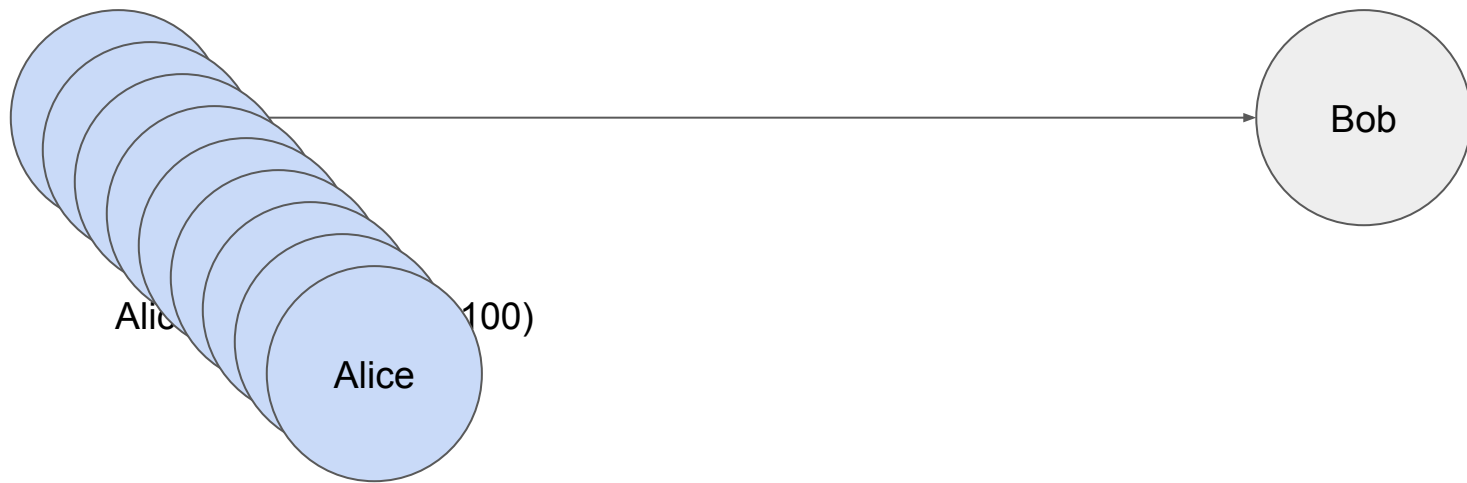
내부 코드는 각 컨트랙트 별로 다르게 구현 가능함.

(ex. transfer시 수수료 지불, 특정주소의 전송 거부)

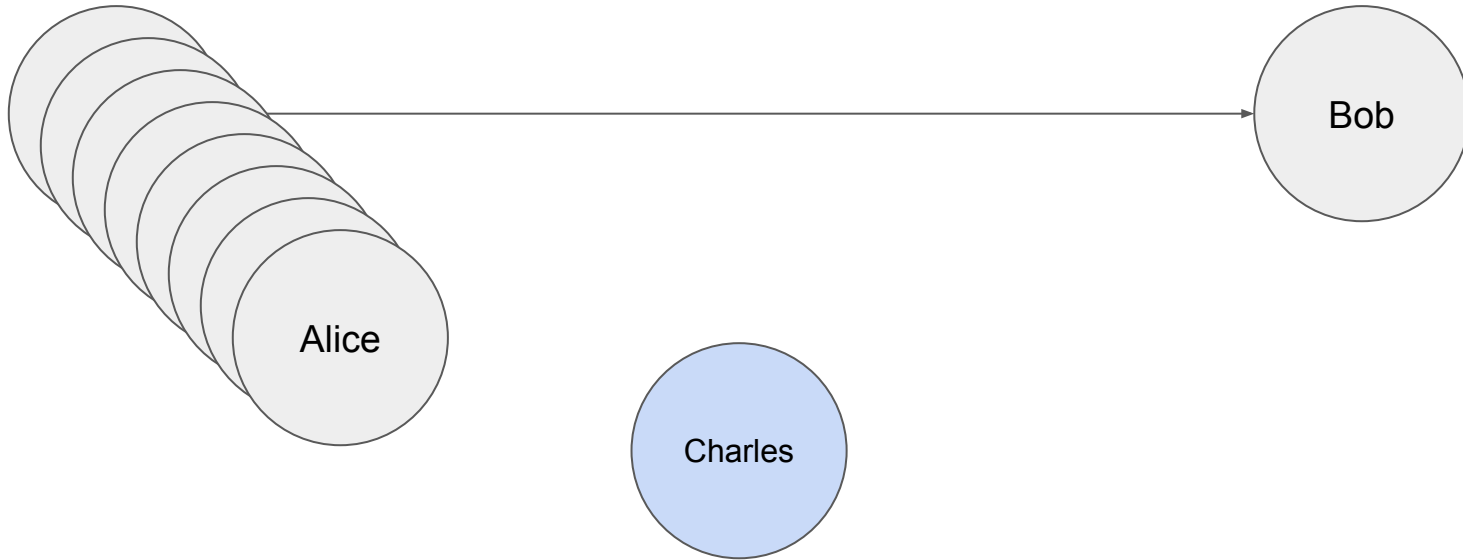
# ERC-20 Interface

1. `totalSupply()` public view returns (uint256 totalSupply)  
전체 토큰의 발행량을 가져옵니다.
2. `balanceOf(address _owner)` public view returns (uint256 balance)  
\_owner가 가진 토큰의 잔액을 가져옵니다.
3. `transfer(address _to, uint256 _value)` public returns (bool success)  
\_value만큼의 토큰을 \_to 주소로 전송합니다
4. `transferFrom(address _from, address _to, uint256 _value)` public returns (bool success)  
\_from의 주소에서 \_to주소로 \_value만큼의 토큰을 보냅니다.
5. `approve(address _spender, uint256 _value)` public returns (bool success)  
\_spender에게 \_value만큼의 토큰사용을 허용합니다.
6. `allowance(address _owner, address _spender)` public view returns (uint256 remaining)  
\_owner가 \_spender에게 사용을 허락한 토큰 양을 가져옵니다.

# Transfer



# TransferFrom



Charles : `transferFrom( alice ,bob, 100)`

# TransferFrom을 쓰는 이유는?

여러 개의 지갑의 자산을 관리해야 할 때.

이더리움은 **Transaction**을 처리 할 때 **Gas**를 지불할 **ETH**를 필요로 함.  
따라서 **ERC-20**토큰만 있는 경우, 토큰 전송을 할 수 없음.

여러개의 지갑을 관리한다고 했을때 각 지갑에 **ETH**를 일정량 보관해야 하며, 매번 토큰 전송시마다 **ETH** 수량을 체크해야함.

만일 여러개의 지갑들이 관리자 지갑에 **Approve**를 최대치로 주게되면, 이후부터는 각지갑이 토큰을 전송하지 않고, 관리자 지갑을 통해 자산 이동이 가능해짐.

# eosio.token

## 1. Create

새로운 토큰을 생성한다

## 2. Issue

생성된 토큰을 발행하여 공급한다

## 3. transfer

토큰을 **from**에서 **to**에게 전송한다



# 다른기능들은?

잔액확인은?

-> **accounts Table** 조회를 통해 계정이 가진 잔액확인이 가능

토큰의 발행량은?

-> **stats Table** 조회를 통해 해당 토큰의 정보를 확인 가능

## 다른기능들은?

	ERC-20	eosio.token
토큰 생성	기능 없음. 한개 컨트랙트는 한개의 토큰만 관리	<b>Create</b> 하나의 컨트랙트에서 여러개의 토큰을 관리
토큰의 발행	기능 없음. 일반적으로 <b>Fixed supply</b>	<b>Issue</b> <b>Maximum Supply</b> 범위 안에 추가발행 가능
토큰의 전송	<b>transfer</b> <b>transferFrom</b> <b>approve</b>	<b>Transfer</b>
잔액 확인	<b>balanceOf</b>	<b>get table accounts</b>
토큰의 발행량	<b>totalSupply</b>	<b>get table stats</b>

# EOSIO.TOKEN 컨트랙트를 보기에 앞서

eos/contracts/eosio.token 경로에 있는 소스코드를  
그대로 사용할 경우 빌드와 배포에 문제가 있습니다.

- 컨트랙트 코드의 규칙위반 (12 글자를 넘어가는 구조체 이름)
- Table에 대한 ABI 부재

[https://github.com/Hexlant/eosio.token\\_example/commit/61a9cc22cef1dcc9272a14b04696ce623fd8926a](https://github.com/Hexlant/eosio.token_example/commit/61a9cc22cef1dcc9272a14b04696ce623fd8926a)

# 예제 소스

[https://github.com/Hexlant/eosio.token\\_example](https://github.com/Hexlant/eosio.token_example)

The screenshot shows the GitHub repository page for `Hexlant/eosio.token_example`. The repository has 3 watchers, 0 stars, and 0 forks. The main content area shows the commit history and the README file.

**Repository Information:**

- Watch: 3
- Star: 0
- Fork: 0

**Navigation:** Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, Settings

**Commit History:**

Commit	Branch	Releases	Contributor
4 commits	1 branch	0 releases	1 contributor

**Branch: master** | [New pull request](#) | [Create new file](#) | [Upload files](#) | [Find file](#) | [Clone or download](#)

**Commit Log:**

Commit	Message	Time
7cea2ba	Update README.md	25 minutes ago
7cea2ba	Update README.md	25 minutes ago
7cea2ba	modify source code for compile/deploy	33 minutes ago
7cea2ba	modify source code for compile/deploy	33 minutes ago

**README.md**

### eosio.token\_example

Compile

```
$ eosio.cpp -o eosio.token_example.wasm eosio.token.cpp
```

```
$ eosio.cpp -g eosio.token_example.abi eosio.token.hpp
```

Deploy

```
$ ./cleos.sh set contract [Contract Account] ~/path/eosio.token_example -p [Contract Account]
```

컨트랙트 배포를 위한 계정 생성

<https://monitor.jungletestnet.io/#home>

# EOS Jungle Testnet Monitor (GitHub)

Ver: 2.1 | TPS: 10008483 / All time high: 3 / 1222 | APS Live/All time high: 3/1222 | Connected Users: 124

[Register](#) | [Create Account](#) | [Account Info](#) | [Account History](#) | [Faucet](#) | [Get TX](#) | [API](#) | [P2P List](#) | [Telegram Bot](#) | [Map](#) | [Downtime](#)

## 10 008 483

irreversible: 10 008 162 [-321]

2018-08-16 22:37:00

Produced by: **komododragon**

Total -- Transactions in -- Blocks / 10008483 Blocks

```
10008483 benchbenchbp::runcpu2 benchbenchbp ""
10008483 benchbenchbp::runcpu2 benchbenchbp ""
10008482 bigertestabc::create bigertestabc Issue undefined undefined
10008465 bigertestabc::updateblk bigertestabc {"account":"bigertestabc","dealld":"1534426602886","eosRealblock":"10008455","blktme":"2018-08-16T13:36:46.000","blockSize":8892,"blockId":"0098b787fc1186a70672dc4c807d14597bfad0c9ffd4"}
10008464 benchbenchbp::runcpu2 benchbenchbp ""
10008460 bigertestabc::updaterefbk bigertestabc {"account":"bigertestabc","dealld":"1534426602886","eosRefblock":46654,"eosTxHash":"2332"}
10008458 benchbenchbp::runcpu2 benchbenchbp ""
```

## Network Status

Vote	System	All Nodes	Transactions	Legend					
#	Name	Answered	Block	Produced	Block	Blocks	TXs	Organisation	Votes
<input type="checkbox"/> 1 [5ms]	lioninjungle	3sec	10008476	1min55sec	10008279	375937	108642	CryptoLions	267 856 053 EOS <sup>23.66%</sup>
<input type="checkbox"/> 2 [331ms]	blocksmithio	1sec	10008478	36sec	10008410	211065	34048	EOS BlockSmith	42 277 983 EOS <sup>3.73%</sup>
<input type="checkbox"/> 3 [58ms]	wearegeckooo	1sec	10008480	3min27sec	10008095	352507	109548	EOSGen	41 987 809 EOS <sup>3.7%</sup>
<input checked="" type="checkbox"/> 4 [5ms]	komododragon	2sec	10008478	0 sec	10008483	384478	112191	mixbytes.io	39 945 720 EOS <sup>3.52%</sup>
<input type="checkbox"/> 5 [32ms]	mosquitometa	10sec	10008463	1min48sec	10008293	280840	78879	EOSMetal.io	39 100 895 EOS <sup>3.45%</sup>
<input type="checkbox"/> 6 [88ms]	eosdacserval	1sec	10008480	24sec	10008434	167196	42454	eosDAC	38 960 796 EOS <sup>3.44%</sup>
<input type="checkbox"/> 7 [7ms]	okapitestnet	1sec	10008479	1min36sec	10008317	236840	45545	mixbytes.io	38 761 300 EOS <sup>3.42%</sup>
<input type="checkbox"/> 8 [48ms]	galapagoloco	8sec	10008467	6sec	10008470	154905	28346	tortuomachines	38 693 735 EOS <sup>3.41%</sup>

## EOS Jungle Testnet Monitor (GitHub)

## Create account



Account name

(a-z,1-5 are allowed only. Length 12)

Owner Public Key

Active Public Key

## Network Status

System

All Nodes

Transactions

Legend

Name	Answered	Block	Produced	Block	Blocks	TXs	Organisation	Votes
lioninjungle	7sec	10008669	1min43sec	10008505	375949	108644	CryptoLions	267 856 053 EOS
blocksmith	6sec	10008671	24sec	10008625	211075	21050	EOS BlockSmith	12 277 222 EOS



# 테스트에 필요한 EOS토큰 받기

**EOS Jungle Testnet Monitor (GitHub)**

v1.2.1 | TPS Live/All time high: 3 / 1222 | TPS High: 3/1222 | Connected Users: 124

[Register](#) | [Create Account](#) | [Account Info](#) | [Account History](#) | [Faucet](#) | [Test TX](#) | [API](#) | [P2P List](#) | [Telegram Bot](#) | [Map](#) | [Downtime](#)

**10 008 483**

irreversible: 10 008 162 [-321]

2018-08-16 22:37:00

Produced by: [komododragon](#)

**Network Status**

#	Name	Answered	Block	Produced	Block	Blocks	TXs	Organisation	Votes
<input type="checkbox"/> 1[5ms]	lioninjungle	3sec	10008476	1min55sec	10008279	375937	108642	CryptoLions	267 856 053 EOS <sup>23.66%</sup>
<input type="checkbox"/> 2[331ms]	blocksmithio	1sec	10008478	36sec	10008410	211065	34048	EOS BlockSmith	42 277 983 EOS <sup>3.73%</sup>
<input type="checkbox"/> 3[58ms]	wearegeckooo	1sec	10008480	3min27sec	10008095	352507	109548	EOSGen	41 987 809 EOS <sup>3.7%</sup>
<input checked="" type="checkbox"/> 4[5ms]	komododragon	2sec	10008478	0 sec	10008483	384478	112191	mixbytes.io	39 945 720 EOS <sup>3.52%</sup>
<input type="checkbox"/> 5[32ms]	mosquitometa	10sec	10008463	1min48sec	10008293	280840	78879	EOSMetal.io	39 100 895 EOS <sup>3.45%</sup>
<input type="checkbox"/> 6[88ms]	eosdacserver	1sec	10008480	24sec	10008434	167196	42454	eosDAC	38 960 796 EOS <sup>3.44%</sup>
<input type="checkbox"/> 7[7ms]	okapitestnet	1sec	10008479	1min36sec	10008317	236840	45545	mixbytes.io	38 761 300 EOS <sup>3.42%</sup>
<input type="checkbox"/> 8[48ms]	galanadoloco	8sec	10008467	6sec	10008470	154905	28346	tortuoomachines	38 693 735 EOS <sup>3.41%</sup>



# 컨트랙트 빌드 및 배포

```
$ eosio-cpp -o eosio.token.wasm eosio.token.cpp
```

```
$ eosio-cpp -I include -o eosio.token.wasm eosio.token.cpp --abigen
```

```
Jinhoui-MacBook-Pro:eosio.token jinholee$ eosio-cpp -o eosio.token.wasm eosio.token.cpp
Jinhoui-MacBook-Pro:eosio.token jinholee$ eosio-cpp -I include -o eosio.token.wasm eosio.token.cpp --abigen
Warning, empty ricardian clause file
Warning, empty ricardian clause file
Warning, action <create> does not have a ricardian contract
Warning, action <issue> does not have a ricardian contract
Warning, action <retire> does not have a ricardian contract
Warning, action <transfer> does not have a ricardian contract
Warning, action <open> does not have a ricardian contract
Warning, action <close> does not have a ricardian contract
Warning, action <create> does not have a ricardian contract
Warning, action <issue> does not have a ricardian contract
Warning, action <retire> does not have a ricardian contract
Warning, action <transfer> does not have a ricardian contract
Warning, action <open> does not have a ricardian contract
Warning, action <close> does not have a ricardian contract
Jinhoui-MacBook-Pro:eosio.token jinholee$
```

# Account 구조체

// 계정별 잔액 정보를 저장하는 구조체

```
struct [[eosio::table]] account {  
    asset    balance;  
    uint64_t primary_key()const { return balance.symbol.code().raw(); }  
};  
  
typedef eosio::multi_index< "accounts"_n, account > accounts;
```

KEY	balance
balance.symbol   == EOS	....
JUNGLE	....
	....
	.....

# stat 구조체

```
struct [[eosio::table]] currency_stats {  
    asset    supply;  
    asset    max_supply;  
    name     issuer;  
    uint64_t primary_key()const { return supply.symbol.code().raw(); }  
};  
  
typedef eosio::multi_index< "stat"_n, currency_stats > stats;
```

KEY	supply	max_supply	issuer
EOS	...	...	...
JUNGLE	...	...	...
	...	...	...

# Table 확인해 보기

```
$ ./cleos.sh get table eosio.token clubmateshot accounts
```

```
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$ ./cleos.sh get table eosio.token clubmateshot accounts
{
  "rows": [{
    "balance": "200.0000 EOS"
  }, {
    "balance": "200.0000 JUNGLE"
  }
],
  "more": false
}
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$
```

# Table 확인해 보기

```
$ ./cleos.sh get table eosio.token EOS stat
$ ./cleos.sh get table eosio.token JUNGLE stat
```

```
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$ ./cleos.sh get table eosio.token EOS stat
{
  "rows": [{
    "supply": "8098125107.8281 EOS",
    "max_supply": "100000000000.0000 EOS",
    "issuer": "eosio"
  }
],
  "more": false
}
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$ ./cleos.sh get table eosio.token JUNGLE stat
{
  "rows": [{
    "supply": "10000020000.0000 JUNGLE",
    "max_supply": "100000000000.0000 JUNGLE",
    "issuer": "eosio"
  }
],
  "more": false
}
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$
```

# Create

새로운 토큰을 만들어 주는 기능.

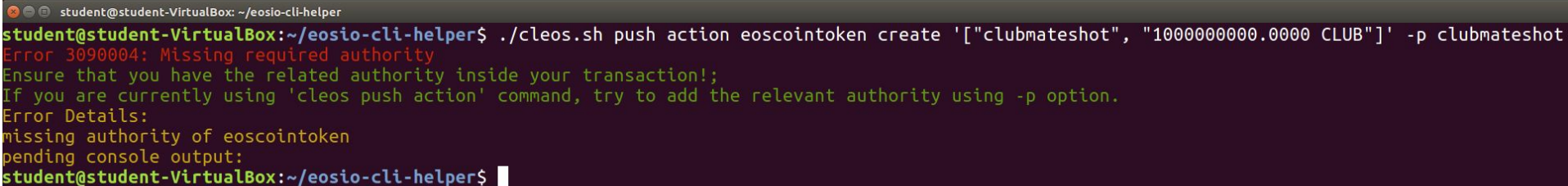
이를 활용하면 매번 컨트랙트를 배포 할 필요없이,  
하나의 컨트랙트로 여러개의 토큰을 관리.

```
void create( name          issuer,          <-토큰을 발행할 수 있는 계정  
            asset         maximum_supply); <-최대로 발행 할 수 있는 토큰
```

# Create Code 분석

```
void create( name          issuer,
             asset         maximum_supply){

    // action을 푸시한 계정이 _self(컨트랙트 계정) 인지 확인
    require_auth( _self);
```



The screenshot shows a terminal window with the title bar 'student@student-VirtualBox: ~/eosio-cli-helper'. The user has executed the command `./cleos.sh push action eoscointoken create '["clubmateshot", "1000000000.0000 CLUB"]' -p clubmateshot`. The output shows an error: `Error 3090004: Missing required authority`. The message continues: `Ensure that you have the related authority inside your transaction!;` and `If you are currently using 'cleos push action' command, try to add the relevant authority using -p option.` The error details are: `missing authority of eoscointoken` and `pending console output:`. The prompt `student@student-VirtualBox:~/eosio-cli-helper$` is visible at the bottom.

```
student@student-VirtualBox: ~/eosio-cli-helper
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken create '["clubmateshot", "1000000000.0000 CLUB"]' -p clubmateshot
Error 3090004: Missing required authority
Ensure that you have the related authority inside your transaction!;
If you are currently using 'cleos push action' command, try to add the relevant authority using -p option.
Error Details:
missing authority of eoscointoken
pending console output:
student@student-VirtualBox:~/eosio-cli-helper$
```

Create를 사용하려면 반드시 컨트랙트의 계정으로만 생성 해야함

student@student-VirtualBox: ~/eosio-cli-helper

```
student@student-VirtualBox:~/eosio-cli-helper$
student@student-VirtualBox:~/eosio-cli-helper$
student@student-VirtualBox:~/eosio-cli-helper$
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken create '["clubmateshot", "1000000000.0000 CLUB"]' -p eoscointoken
executed transaction: 822f7201b00bbe453f743309d000021e3a3257e8153ec6f142dd2679a0ce7139 120 bytes 470 us
# eoscointoken <= eoscointoken::create {"issuer":"clubmateshot","maximum_supply":"1000000000.0000 CLUB"}
warning: transaction executed locally, but may not be confirmed by the network yet ]
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get table eoscointoken CLUB stat
{
  "rows": [{
    "supply": "0.0000 CLUB",
    "max_supply": "1000000000.0000 CLUB",
    "issuer": "clubmateshot"
  }],
  "more": false
}
student@student-VirtualBox:~/eosio-cli-helper$
```

```
Jinhui-MacBook-Pro: eosio-cli-helper jinholee$ ./cleos.sh push action hellotestman create '["clubmateshot", "10000.0000 TEST"]' -p hellotestman
executed transaction: 94a5b5986e7a6701193bc9add85f38e0a19375bbfb59a52b60c9cea72c29df3c 120 bytes 137 us
# hellotestman <= hellotestman::create {"issuer":"clubmateshot","maximum_supply":"10000.0000 TEST"}
warning: transaction executed locally, but may not be confirmed by the network yet ]
```



# Create Code 분석

// 심볼이 올바른 심볼인지 확인

`eosio_assert( sym.is_valid(), "invalid symbol name" );`

```
student@student-VirtualBox: ~/eosio-cli-helper
student@student-VirtualBox:~/eosio-cli-helper$
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eosointoken create '['clubmateshot',
"1000000000.0000 club"]' -p eosointoken
Error 3010014: Invalid symbol
Error Details:
invalid character in symbol name
rethrow
rethrow
rethrow
'["clubmateshot","1000000000.0000 club"]' is invalid args for action 'create' code 'eosointoken'. expected
'[{ "name": "issuer", "type": "name"}, { "name": "maximum_supply", "type": "asset"}]'
code: eosointoken, action: create, args: ["clubmateshot","1000000000.0000 club"]
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eosointoken create '['clubmateshot',
"1000000000.0000 TATKSJIHOIHSJHKJHKJDKL FHKJ"]' -p eosointoken
Error 3010014: Invalid symbol
Error Details:
invalid symbol: _K^_[NO
rethrow
rethrow
'["clubmateshot","1000000000.0000 TATKSJIHOIHSJHKJHKJDKL FHKJ"]' is invalid args for action 'create' code 'e
osointoken'. expected ' [{ "name": "issuer", "type": "name"}, { "name": "maximum_supply", "type": "asset"}] '
code: eosointoken, action: create, args: ["clubmateshot","1000000000.0000 TATKSJIHOIHSJHKJHKJDKL FHKJ"]
student@student-VirtualBox:~/eosio-cli-helper$
```

# sym.is\_valid()

eoslib/symbol.hpp

심볼의 조건

- 영문 대문자
- 7글자 이내

```
/**
 * uint64_t representation of a symbol name
 */
typedef uint64_t symbol_name;

/**
 * Checks if provided symbol name is valid.
 *
 * @param sym - symbol name of type symbol_name
 * @return true - if symbol is valid
 */
static constexpr bool is_valid_symbol( symbol_name sym ) {
    sym >>= 8;
    for( int i = 0; i < 7; ++i ) {
        char c = (char)(sym & 0xff);
        if( !('A' <= c && c <= 'Z') ) return false;
        sym >>= 8;
        if( !(sym & 0xff) ) {
            do {
                sym >>= 8;
                if( (sym & 0xff) ) return false;
                ++i;
            } while( i < 7 );
        }
    }
    return true;
}
```

# eoslib/asset.hpp

```
eosio_assert( maximum_supply.is_valid(), "invalid supply");
eosio_assert( maximum_supply.amount > 0, "max-supply must be positive");
static constexpr int64_t max_amount = (1LL << 62) - 1;

/**
 * Construct a new asset given the symbol name and the amount
 *
 * @brief Construct a new asset object
 * @param a - The amount of the asset
 * @param s - The name of the symbol, default to CORE_SYMBOL
 */
explicit asset( int64_t a = 0, symbol_type s = CORE_SYMBOL )
: amount(a), symbol{s}
{
    eosio_assert( is_amount_within_range(), "magnitude of asset amount must be less than 2^62" );
    eosio_assert( symbol.is_valid(), "invalid symbol name" );
}

/**
 * Check if the amount doesn't exceed the max amount
 *
 * @brief Check if the amount doesn't exceed the max amount
 * @return true - if the amount doesn't exceed the max amount
 * @return false - otherwise
 */
bool is_amount_within_range()const { return -max_amount <= amount && amount <= max_amount; }

/**
 * Check if the asset is valid. %A valid asset has its amount <= max_amount and its symbol name valid
 *
 * @brief Check if the asset is valid
 * @return true - if the asset is valid
 * @return false - otherwise
 */
bool is_valid()const { return is_amount_within_range() && symbol.is_valid(); }
```

```
stats statstable( _self, sym.name() );  
// 동일한 심볼을 갖는 stst이 있는지 확인  
auto existing = statstable.find( sym.name() );  
eosio_assert( existing == statstable.end(), "token with symbol already exists" );
```

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken create '["clubmateshot",  
"1000000000.0000 EOSCOIN"]' -p eoscointoken  
executed transaction: 0784d944bdf977e4254fdc4806a70d718cba4df5976a0ddd8d890c7b25f9e8f5 120 bytes 493 us  
# eoscointoken <= eoscointoken::create {"issuer":"clubmateshot","maximum_supply":"1000000000.0000  
EOSCOIN"}  
warning: transaction executed locally, but may not be confirmed by the network yet  
student@student-VirtualBox:~/eosio-cli-helper$  
student@student-VirtualBox:~/eosio-cli-helper$  
student@student-VirtualBox:~/eosio-cli-helper$  
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken create '["clubmateshot",  
"1000000000.0000 EOSCOIN"]' -p eoscointoken  
Error 3050003: eosio_assert_message assertion failure  
Error Details:  
assertion failure with message: token with symbol already exists  
pending console output:  
student@student-VirtualBox:~/eosio-cli-helper$
```

# QUIZ

이더리움 ERC-20 토큰은 심볼 중복이 가능 할까요?

# Issue

## 토큰의 발행

처음 생성된 토큰은 정의만 되었을 뿐, 실제로 발행된 수량이 없음

**Issue**함수를 통해 토큰을 발행하여, 공급할 수 있음

```
void issue( account_name to,    // <- 새로 발행된 토큰을 받을 계정
            asset quantity,     // <- 발행할 토큰의 수량
            string memo );      // <- 메모 : 토큰발행/전송시 부가적인 정보를 전달 할 수 있음
```

```
// 심볼을 스코프로 하여 테이블 로드
```

```
auto sym_name = sym.name();
```

```
stats statstable( _self, sym_name );
```

```
auto existing = statstable.find( sym_name );
```

```
// 해당 심볼을 갖는 stat정보가 있는지 확인
```

```
eosio_assert( existing != statstable.end(), "token with symbol does not exist, create token before issue" );
```

```
const auto& st = *existing;
```

```
student@student-VirtualBox:~/eosio-cli-helper$
```

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken issue '["clubmateshot",  
"100000.0000 NOMONEY", "xx"]' -p clubmateshot
```

```
Error 3050003: eosio_assert_message assertion failure
```

```
Error Details:
```

```
assertion failure with message: token with symbol does not exist, create token before issue
```

```
pending console output:
```

```
student@student-VirtualBox:~/eosio-cli-helper$
```

# Issue

// action을 push한 계정이 해당 토큰의 발행자 인지 체크

**require\_auth( st.issuer );**

// 발행하려는 토큰의 수량이 0보다 큰지 확인

**eosio\_assert( quantity.is\_valid(), "invalid quantity" );**

**eosio\_assert( quantity.amount > 0, "must issue positive quantity" );**

// 발행하려는 토큰의 심볼이 stat정보과 맞는지 확인

**eosio\_assert( quantity.symbol == st.supply.symbol, "symbol precision mismatch" );**

// 발행했을때의 발행량이 최대 발행량을 넘지는 않는지 확인

**eosio\_assert( quantity.amount <= st.max\_supply.amount - st.supply.amount, "quantity exceeds available supply");**



# Issue -Issuer 권한 검증

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get table eoscointoken CLUB stat
{
  "rows": [{
    "supply": "0.0000 CLUB",
    "max_supply": "1000000000.0000 CLUB",
    "issuer": "clubmateshot"
  }
],
"more": false
}
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken issue '["clubmateshot",
"10.00 CLUB", "xx"]' -p hellotestman
Error 3090004: Missing required authority
Ensure that you have the related authority inside your transaction!;
If you are currently using 'cleos push action' command, try to add the relevant authority using -p option.
Error Details:
missing authority of clubmateshot
pending console output:
student@student-VirtualBox:~/eosio-cli-helper$ █
```

# Issue - quantity 검증

```
student@student-VirtualBox: ~/eosio-cli-helper
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken issue '["clubmateshot",
"10.00 CLUB", "xx"]' -p clubmateshot
Error 3050003: eosio_assert_message assertion failure
Error Details:
assertion failure with message: symbol precision mismatch
pending console output:
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken issue '["clubmateshot",
"1000000000000.0000 CLUB", "xx"]' -p clubmateshot
Error 3050003: eosio_assert_message assertion failure
Error Details:
assertion failure with message: quantity exceeds available supply
pending console output:
student@student-VirtualBox:~/eosio-cli-helper$ █
```

# Issue - 발행과 전송

// 발행권자 계좌에 발행한 금액 추가

```
add_balance( st.issuer, quantity, st.issuer );
```

// 만일 발행된 토큰을 받을 사람이 발행자가 아니라면, 토큰 전송

```
if( to != st.issuer ) {
```

```
    SEND_INLINE_ACTION( *this, transfer, {st.issuer,N(active)},  
                        {st.issuer, to, quantity, memo} );
```

```
}
```

# Issue - 발행과 전송

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken issue '{"clubmateshot",
"100.0000 CLUB", "xx"}' -p clubmateshot
executed transaction: 0794b6a0defdcc367c12d2700a1ca1574f7c3d249a2e9d7b877c11cb185deff8 120 bytes 598 us
# eoscointoken <= eoscointoken::issue {"to":"clubmateshot","quantity":"100.0000 CLUB","memo":"xx"}
}
warning: transaction executed locally, but may not be confirmed by the network yet ]
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken issue '{"lioninjungle",
"100.0000 CLUB", "xx"}' -p clubmateshot
executed transaction: 66f57acce2c41296c017728f6af3dab001fdffde58fe4995fb94f26811ec94a2 120 bytes 1212 us
# eoscointoken <= eoscointoken::issue {"to":"lioninjungle","quantity":"100.0000 CLUB","memo":"xx"}
}
# eoscointoken <= eoscointoken::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"100.
0000 CLUB","memo":"xx"}
# clubmateshot <= eoscointoken::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"100.
0000 CLUB","memo":"xx"}
# lioninjungle <= eoscointoken::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"100.
0000 CLUB","memo":"xx"}
warning: transaction executed locally, but may not be confirmed by the network yet ]
student@student-VirtualBox:~/eosio-cli-helper$
```

# transfer

## 토큰의 전송

가지고 있는 토큰의 수량 중 일부를 다른 계정으로 보내는 기능

```
void token::transfer( account_name from,    // <- 토큰은 보내는 계정
                     account_name to,      // <- 토큰을 받을 계정
                     asset      quantity,  // <- 보내려는 토큰의 종류와 수량
                     string      memo )    // <- 메모
```

# From / To 계정에 대한 검증

// 자기자신에게 보내는 전송이면 **assert**

**eosio\_assert**( from != to, "cannot transfer to self" );

// action을 push한 계정이 from인지 확인

**require\_auth**( from );

// 받을 계정이 EOS네트워크에 존재하는 계정인지 확인

**eosio\_assert**( **is\_account**( to ), "to account does not exist");

# 토큰 전송에 대한 전파

// from과 to 계정에 action 내용 전파

```
require_recipient( from );
```

```
require_recipient( to );
```

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eoscointoken transfer '["clubmateshot","lioninjungle", "1.0000 CLUB", "xx"]' -p clubmateshot
executed transaction: c20065714afa3f50b496d01af9ab98cadb9099340f44ec3753acf299c8b5ece5 128 bytes 730 us
# eoscointoken <= eoscointoken::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"1.0000 CLUB","memo":"xx"}
# clubmateshot <= eoscointoken::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"1.0000 CLUB","memo":"xx"}
# lioninjungle <= eoscointoken::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"1.0000 CLUB","memo":"xx"}
warning: transaction executed locally, but may not be confirmed by the network yet
student@student-VirtualBox:~/eosio-cli-helper$
```



# 토큰 전송에 대한 전파

// from과 to 계정에 action 내용 전파

// require\_recipient( from );

require\_recipient( to );

student@student-VirtualBox: ~/eosio-cli-helper

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$ ./cleos.sh push action eosio.token transfer '["clubmateshot","lioninjungle", "1.0000 CLUB", "xx"]' -p clubmateshot

executed transaction: 05c0be82d2d769adb2607db41d2356ac087071e277c1616f019855d42cc49cfe 128 bytes 685 us

# eosio.token <= eosio.token::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"1.0000 CLUB","memo":"xx"}

# lioninjungle <= eosio.token::transfer {"from":"clubmateshot","to":"lioninjungle","quantity":"1.0000 CLUB","memo":"xx"}

warning: transaction executed locally, but may not be confirmed by the network yet ]

student@student-VirtualBox:~/eosio-cli-helper\$



# Accounts Table Update

```
void token::add_balance( account_name owner, asset value, account_name ram_payer )
{
    accounts to_acnts( _self, owner );
    auto to = to_acnts.find( value.symbol.name() );
    if( to == to_acnts.end() ) {
        to_acnts.emplace( ram_payer, [&]( auto& a ){
            a.balance = value;
        });
    } else {
        to_acnts.modify( to, 0, [&]( auto& a ) {
            a.balance += value;
        });
    }
}
```

# Ram Payer!

새로운 **Account** 정보가 테이블에 저장되면서 발생하는 **RAM** 사용 이슈

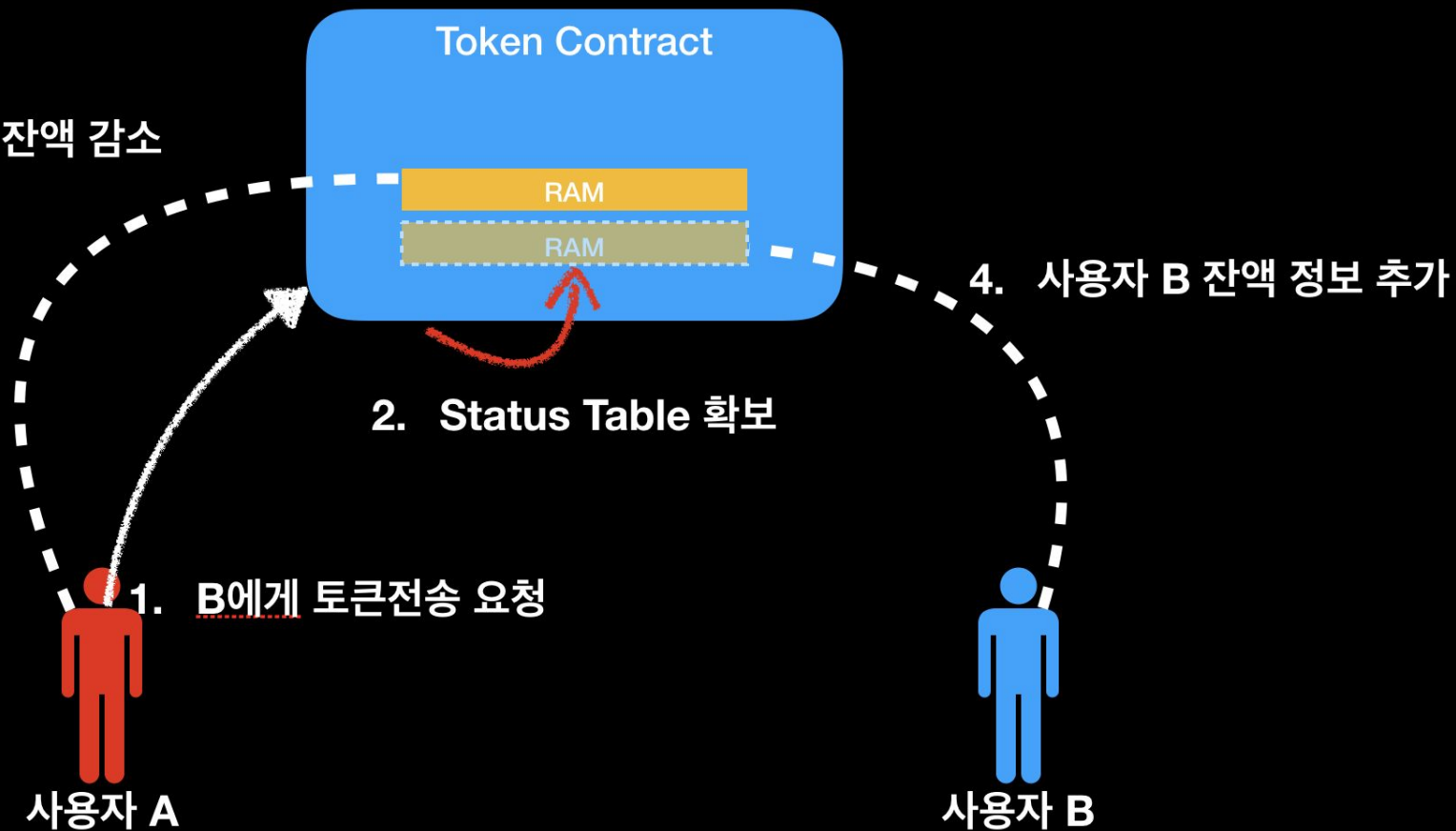
왜 컨트랙트 계정이 부담하지 않을까?

- **SPAM Attack**으로 인한 **Contract RAM** 비용 증가

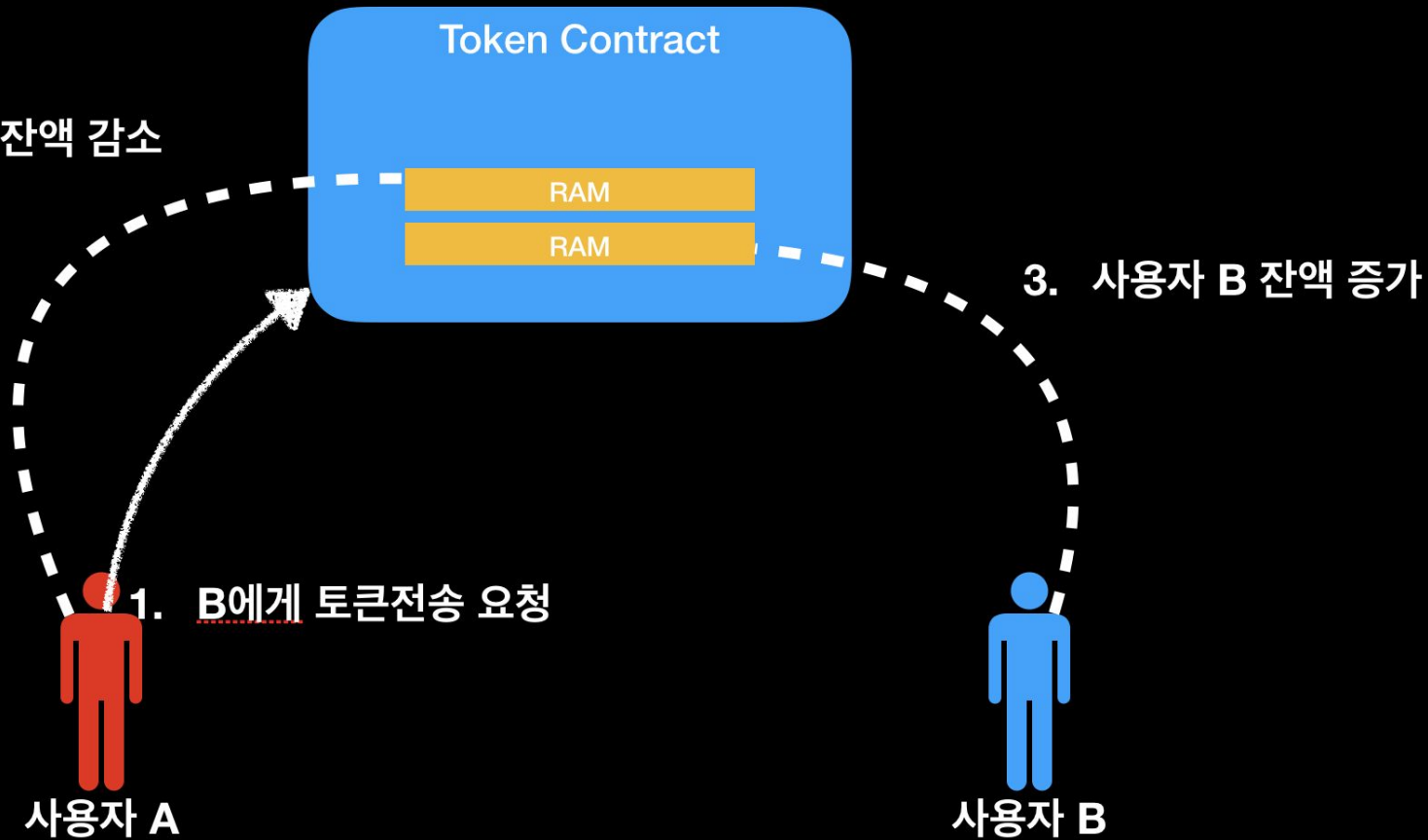
누군가가 컨트랙트를 사보타주 할 목적으로 무작위 다수에게 토큰을 보낼 경우,

컨트랙트가 확보해야 할 **RAM** 용량 증가 -> 컨트랙트 무력화

3. 사용자 A 잔액 감소



2. 사용자 A 잔액 감소



3. 사용자 B 잔액 증가

1. B에게 토큰전송 요청

사용자 A

사용자 B

# Accounts Table Update

```
void token::sub_balance( account_name owner, asset value ) {  
    accounts from_acnts( _self, owner );  
  
    const auto& from = from_acnts.get( value.symbol.name(), "no balance object found" );  
    eosio_assert( from.balance.amount >= value.amount, "overdrawn balance" ); //<-잔액 확인  
  
    if( from.balance.amount == value.amount ) {  
        from_acnts.erase( from ); // <-보유잔액이 0이 되면, 메모리 해제  
    } else {  
        from_acnts.modify( from, owner, [&]( auto& a ) {  
            a.balance -= value;  
        });  
    }  
}
```

# RAM 회수

토큰을 받은 계정이, 해당 토큰을 사용 할 경우,  
기존에 RAM을 제공한 계정은 RAM자원을 돌려 받음  
ram\_payer 가 변경되면서, 토큰을 사용한 계정이 RAM 자원을 지원.

# 실제 EOS 토큰전송에 따른 RAM 사용량

	사용자 A RAM 사용량	사용자 B RAM 사용량	사용자 C RAM 사용량
사용자 A만 토큰을 가지고 있을때	5.33 KiB	2.926 KiB	3.365 KiB
A가 B에게 토큰을 전송한 후	5.564 KiB (+0.234)	2.926 KiB	3.365 KiB
B가 C에게 토큰을 전송한 후	5.439 KiB (-0.125)	3.051 KiB (+0.125)	3.365 KiB

# Token Sale

token을 팔아보자.



# Tokensale

토큰을 확산 시키는 방법

Air-drop

ICO



# 컨트랙트를 통한 토큰 세일

컨트랙트에 EOS를 입금하면, Token을 지급하는 컨트랙트

- 프로그램된 세일 정책
- 수기로 전송하는것에 비해 수월한 판매 관리
- 공개 원장장부를 통한 투명한 판매 이력

[https://github.com/Hexlant/eos\\_tokensale\\_example](https://github.com/Hexlant/eos_tokensale_example)

The screenshot shows the GitHub repository page for `Hexlant / eos_tokensale_example`. The repository has 3 watchers, 0 stars, and 0 forks. The main content area displays the commit history and the README file. The README file contains instructions for building and deploying the EOSIO contract.

Repository: Hexlant / eos\_tokensale\_example

4 commits, 1 branch, 0 releases, 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

Latest commit: 328d6eb 26 seconds ago

Update README.md 26 seconds ago

Tokensale example an hour ago

README.md

### eos\_tokensale\_example

```
Build $ eosiocpp -o eos_tokensale_example.wast eos_tokensale_example.cpp
$ eosiocpp -g eos_tokensale_example.abi eos_tokensale_example.cpp
Deploy Contract $ ./cleos.sh set contract [@Contract Account] ~/([path])/eos_tokensale_example -p [@Contract Account]
Usage Create Sale
$ ./cleos.sh push action hellotestman salestart '["eosointoken","100.0000 CLUB";*1]' -p hellotestman
$ ./cleos.sh push action [@sale Contract] startsale '["[@token Contract], [sale cap], [exchange rate]]' -p [@sale Contract]
Buy Token Transfer EOS token to sale contract
$ ./cleos.sh push action eosio.token transfer '["clubmateshot", "hellotestman", "1.0000 EOS", ""]' -p clubmateshot
```

© 2018 GitHub, Inc. | Terms | Privacy | Security | Status | Help | Contact GitHub | Pricing | API | Training | Blog | About

student@student-VirtualBox: ~/eosio-cli-helper

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$

student@student-VirtualBox:~/eosio-cli-helper\$ ./cleos.sh push action eosio.token transfer '["clubmateshot","hellotestman","1.00 EOS",""]' -p clubmateshot

Error 3050003: eosio\_assert\_message assertion failure

Error Details:

assertion failure with message: There isn't sale information

pending console output:

student@student-VirtualBox:~/eosio-cli-helper\$

# 컨트랙트 세일 결과

student@student-VirtualBox: ~/eosio-cli-helper

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action hellotestman salestart '["eosointoken","100.0000 CLUB","1"]' -p hellotestman
executed transaction: b1674f126c132c0feb1eace1f29115bc632b93217add02eb85800c421db00243 128 bytes 486 us
# hellotestman <= hellotestman::salestart {"contraccount":"eosointoken","salecap":"100.0000 CLUB","exchangerate":1}
warning: transaction executed locally, but may not be confirmed by the network yet ]
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get currency balance eosointoken hellotestman
9998.0000 CLUB
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get currency balance eosointoken clubmateshot
10.0000 TET
10099.0000 CLUB
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action eosio.token transfer '["clubmateshot","hellotestman","1.0000 EOS",""]' -p clubmateshot
executed transaction: 2d5b03080264cc293e51e905d26e4219ba65d1ea6cb0745e6c14239b9b419b58 128 bytes 1605 us
# eosio.token <= eosio.token::transfer {"from":"clubmateshot","to":"hellotestman","quantity":"1.0000 EOS","memo":""}
# clubmateshot <= eosio.token::transfer {"from":"clubmateshot","to":"hellotestman","quantity":"1.0000 EOS","memo":""}
# hellotestman <= eosio.token::transfer {"from":"clubmateshot","to":"hellotestman","quantity":"1.0000 EOS","memo":""}
>> clubmateshot paid: 1.0000 EOS exchanged to : 1.0000 CLUBI got EOS!!!
# eosointoken <= eosointoken::transfer {"from":"hellotestman","to":"clubmateshot","quantity":"1.0000 CLUB","memo":""}
# clubmateshot <= eosointoken::transfer {"from":"hellotestman","to":"clubmateshot","quantity":"1.0000 CLUB","memo":""}
warning: transaction executed locally, but may not be confirmed by the network yet ]
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get currency balance eosointoken hellotestman
9997.0000 CLUB
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get currency balance eosointoken clubmateshot
10.0000 TET
10100.0000 CLUB
student@student-VirtualBox:~/eosio-cli-helper$
```

# Sale Information

```
/// @abi table sale i64
```

```
struct sale {  
    account_name contraccount;    // 판매할 토큰의 컨트랙트 계정  
    asset salecap;                // 판매 하려는 토큰의 수량  
    asset sold;                  // 현재까지 판매된 토큰의 수량  
    uint64_t exchangerate;       // 1 EOS 당 지금할 토큰의 수량  
    uint64_t primary_key() const {return contraccount;}  
    EOSLIB_SERIALIZE(sale,(contraccount)(salecap)(sold)(exchangerate))  
};  
typedef multi_index<N(sale),sale> sales;
```

\*실제 토큰세일에서는 기간, 보너스 비율등 다양한 옵션들이 있을 수 있음

# exchange

외부 코드로 부터 transfer action을 전달 받았을때,  
전송받은 transfer 정보를 바탕으로 토큰을 지급

```

#define EOSIO_ABI2( TYPE, MEMBERS ) \
extern "C" { \
    void apply( uint64_t receiver, uint64_t code, uint64_t action ) { \
        auto self = receiver; \
        if( action == N(onerror)) { \
            /* onerror is only valid if it is for the "eosio" code account and authorized by "eosio"'s "active permission */ \
            eosio_assert(code == N(eosio), "onerror action's are only valid from the \"eosio\" system account"); \
        } \
        if( code == self || action == N(onerror) ) { \
            TYPE thiscontract( self ); \
            switch( action ) { \
                EOSIO_API( TYPE, MEMBERS ) \
            } \
            /* does not allow destructor of thiscontract to run: eosio_exit(0); */ \
        } \
        else { \
            TYPE thiscontract( self ); \
            switch( action ) { \
                case N(transfer): return thiscontract.exchange(receiver, code); \ // <-외부 코드로부터 transfer 액션을 전달 받으면, exchange
실행
            } \
            /* does not allow destructor of thiscontract to run: eosio_exit(0); */ \
        } \
    } \
} \

```



```
// EOS토큰이 아닌경우, 이벤트 처리 없이 종료
```

```
if(code != N(eosio.token)) return;
```

```
// #아래 조건문은 EOS가 아닌경우, 트랜잭션을 리젝 시키기 때문에, 다른 토큰을 받을 수가 없음
```

```
// eosio_assert(code == N(eosio.token), "I reject your non-eosio.token deposit");
```

외부 코드가, eosio.token이 아닌경우 exchange 프로세스를 진행하지 않고 함수 종료

**\*만일 assert로 처리하면, 컨트랙트가 올라간 계정은 다른 토큰을 받을 수가 없다.**

```
sales saleinfo(_self,_self);  
// 세일 정보가 없다면 트랜잭션을 리젝시킴  
eosio_assert(saleinfo.begin() != saleinfo.end(), "There isn't sale information");  
  
auto info = *(saleinfo.begin());
```

## 테이블에 저장된 세일 정보 가져오기

saleinfo.begin()의 리턴값은 iterator 타입으로 pointer 처럼 사용할 수 있다.

```
// EOS를 보내준 계정에 전달할 토큰의 수량 계산
```

```
asset tokens;
```

```
tokens.symbol = info.salecap.symbol;
```

```
tokens.amount = data.quantity.amount * info.exchangerate;
```

```
// 만일 세일 캡 이상으로 토큰을 판매될 경우 트랜잭션 리젝.
```

```
eosio_assert(tokens + info.sold <= info.salecap, "This order exceed salecap");
```

입금된 EOS 수량과 교환비율을 이용하여 지급할 토큰 수량 계산

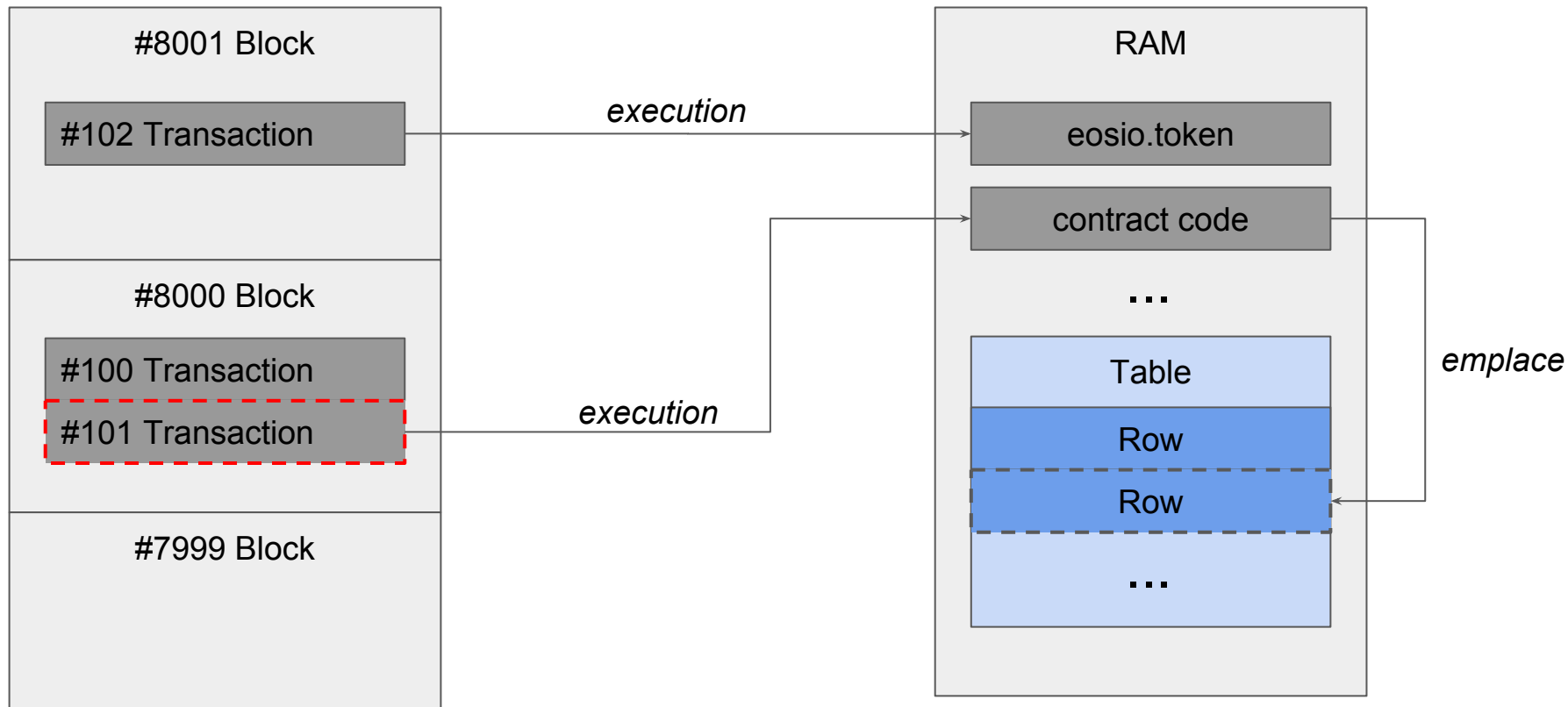
```
// 판매 수량 갱신
saleinfo.modify( saleinfo.begin(), 0,[&](auto& data){
    data.sold += tokens;
});

// 현재 컨트랙트 계정이 가지고 있는 토큰을 전달
action(
    permission_level{ _self, N(active) },
    info.contraccount, N(transfer),
    std::make_tuple(_self, data.from, tokens, std::string(""))
).send();
```

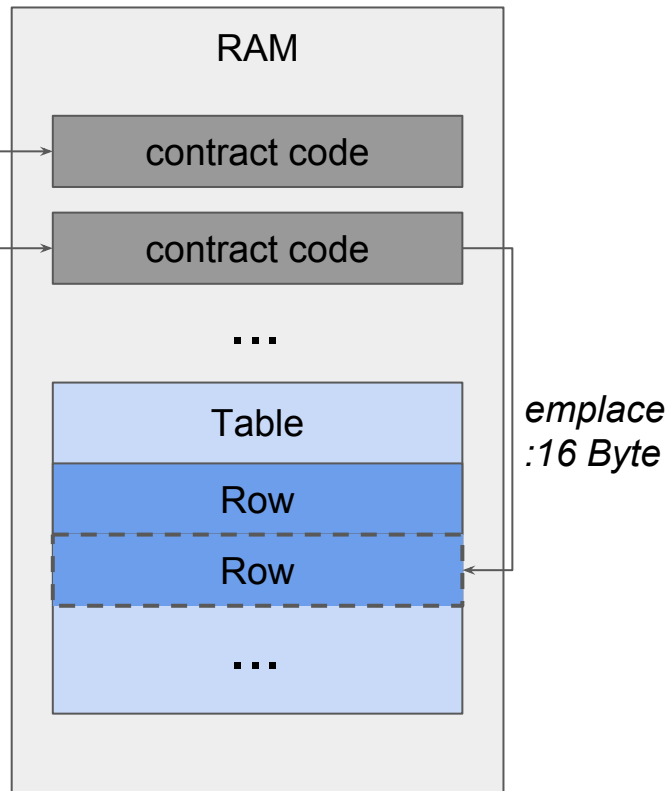
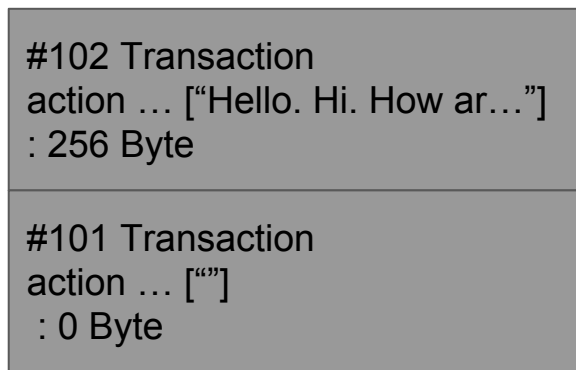
현재 판매 수량을 업데이트 하고, **transfer**를 통해 구매자에게 토큰 지급

# FAQ

# EOS Memory에 대한 이해



# Transaction 매개변수



트랜잭션 **Data**는 RAM사용량에 영향 없음.  
트랜잭션 결과로 테이블이 할당되었을때 메모리  
사용량이 증가

# TEST CODE

```
#include <eosiolib/eosio.hpp>
```

```
using namespace eosio;
```

```
class hello : public eosio::contract {  
public:  
    using contract::contract;
```

```
    // @abi action
```

```
    void hi( account_name user, std::string data ) {  
        require_auth(user);  
        print( "Hello, ", name{user} );  
    }
```

```
    // @abi action
```

```
    void emplace() {  
        rows table(_self, _self);  
        table.emplace(_self, [&](auto& r){  
            r.account = _self;  
            r.data1 = r.data2 = r.data3 = r.data4 = 1;  
            r.data5 = r.data6 = r.data7 = r.data8 = 2;  
        });  
    }
```

```
}
```

```
/// @abi table row i64
```

```
struct row {  
    account_name account;  
    uint64_t data1;  
    uint64_t data2;  
    uint64_t data3;  
    uint64_t data4;  
    uint64_t data5;  
    uint64_t data6;  
    uint64_t data7;  
    uint64_t data8;
```

```
    uint64_t primary_key()const { return account; }
```

```
EOSLIB_SERIALIZE(row,(account)(data1)(data2)(data3)(data4)(data5)(data6)(data7)(data8))
```

```
};
```

```
typedef eosio::multi_index<N(row), row> rows;
```

```
};
```

```
EOSIO_ABI(hello,(hi)(emplace))
```



# 트랜잭션 데이터가 큰 함수를 호출했을 때

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get account hellotestman
permissions:
  owner      1:      1 EOS5GNcmvLr5XcXEWXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca
  active     1:      1 EOS5GNcmvLr5XcXEWXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca1 hellotestman@eosio.code,
memory:
  quota:     6.139 MiB   used:      71.03 KiB
net bandwidth:
  staked:     10.0000 EOS      (total stake delegated from account to self)
  delegated:  0.1000 EOS      (total staked delegated to account from others)
  used:       37.44 KiB
  available:  1.843 MiB
  limit:      1.879 MiB
cpu bandwidth:
  staked:     10.0000 EOS      (total stake delegated from account to self)
  delegated:  0.1000 EOS      (total staked delegated to account from others)
  used:       25.8 ms
  available:  346.4 ms
  limit:      372.1 ms
EOS balances:
  liquid:     110203.0000 EOS
  staked:     20.0000 EOS
  unstaking:  0.0000 EOS
  total:      110223.0000 EOS
producers:    <not voted>
```

# 트랜잭션 데이터가 큰 함수를 호출했을 때

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action hellotestman hi '{"hel
executed transaction: 8ddab3de57a074ca7811c67c061f518a21354f2612c1c3206ec1f71cfc129c7d 160
# hellotestman <= hellotestman::hi {"user":"hellotestman","data":"hello hello h
>> Hello, hellotestman
warning: transaction executed locally, but may not be confirmed by the network yet ]
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get account hellotestman
permissions:
  owner      1:      1 EOS5GNcmvLr5XcXEwXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca
  active     1:      1 EOS5GNcmvLr5XcXEwXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca1 hellotestm
memory:
  quota:      6.139 MiB  used:      71.03 KiB
net bandwidth:
  staked:      10.0000 EOS      (total stake delegated from account to self)
  delegated:    0.1000 EOS      (total staked delegated to account from others)
  used:         37.53 KiB
  available:    1.843 MiB
  limit:        1.879 MiB
cpu bandwidth:
  staked:      10.0000 EOS      (total stake delegated from account to self)
  delegated:    0.1000 EOS      (total staked delegated to account from others)
  used:         27.56 ms
```

# 테이블 할당을 하였을 때

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get account hellotestman
permissions:
  owner      1:      1 EOS5GNcmvLr5XcXEwXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca
  active     1:      1 EOS5GNcmvLr5XcXEwXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca1 hellotestman@eosio.code,
memory:
  quota:      6.139 MiB   used:      71.03 KiB
net bandwidth:
  staked:      10.0000 EOS      (total stake delegated from account to self)
  delegated:    0.1000 EOS      (total staked delegated to account from others)
  used:         37.53 KiB
  available:    1.843 MiB
  limit:        1.879 MiB
cpu bandwidth:
  staked:      10.0000 EOS      (total stake delegated from account to self)
  delegated:    0.1000 EOS      (total staked delegated to account from others)
  used:         27.56 ms
  available:    344.6 ms
  limit:        372.1 ms
EOS balances:
  liquid:      110203.0000 EOS
  staked:       20.0000 EOS
  unstaking:    0.0000 EOS
  total:       110223.0000 EOS
producers:    <not voted>
```

# 테이블 할당을 하였을 때

```
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh push action hellotestman emplace '[]' -p hellotestman
executed transaction: 884bf31fe5d70dce44c093ce300edf16726af54a4552a53b0c8a6a11238a3c35 96 bytes 458 us
hellotestman <= hellotestman::emplace ""
Warning: transaction executed locally, but may not be confirmed by the network yet
student@student-VirtualBox:~/eosio-cli-helper$ ./cleos.sh get account hellotestman
permissions:
  owner      1:      1 EOS5GNcmvLr5XcXEwXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca
  active     1:      1 EOS5GNcmvLr5XcXEwXpyJd9YdAPdTSTWot3YcVi2mypapeZfF45Ca1 hellotestman@eosio.code,
memory:
  quota:      6.139 MiB  used:      71.32 KiB
net bandwidth:
  staked:      10.0000 EOS      (total stake delegated from account to self)
  delegated:    0.1000 EOS      (total staked delegated to account from others)
  used:         37.42 KiB
  available:    1.843 MiB
  limit:        1.879 MiB
cpu bandwidth:
  staked:      10.0000 EOS      (total stake delegated from account to self)
  delegated:    0.1000 EOS      (total staked delegated to account from others)
  used:         28.18 ms
  available:    344 ms
  limit:        372.1 ms
```

# Transaction 매개변수

트랜잭션 Data는 RAM사용량에 영향 없음.

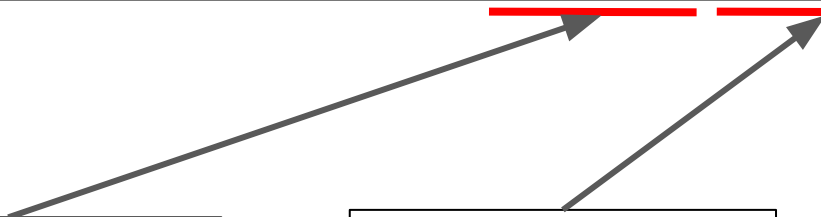
트랜잭션 결과로 테이블이 할당되었을때 메모리 사용량이 증가

# Multi Index Table

```
games gametable(_self, _self);
```

테이블이 있는 코드의  
계정

테이블의 스코프



RAM

```
games gametable(_self,_self);
```

contract code

Table

Row (PK:1111)

Row (PK:1112)

Row (PK:1113)

...

Row (PK:90000)

Row (PK:90001)

Row (PK:90002)

Self Scope 안에서 모든 데이터를 조회 해야함  
PK가 중복 될 수 없음.

-> 인덱싱 / 검색에서 비효율 발생

RAM

```
games gametable(_self,host);
```

contract code

Table

Scope : A

Row (PK:1111)

Row (PK:1112)

Row (PK:1113)

...

Scope : X

Row (PK:1112)

Row (PK:2001)

Row (PK:90002)

각 Scope 별로 조회가 가능  
PK가 같더라도, Scope가 다르면 충돌되지 않음  
->데이터 인덱싱 / 관리에 효율적



```
./cleos.sh system buyram ${account} ${account} 25 --kbytes
```

```
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$ ./cleos.sh set contract hellotestman eosio.token
Reading WASM from /Users/jinholee/eosio-cli-helper/eosio.token/eosio.token.wasm...
Publishing contract...
Error 3080001: Account using more than allotted RAM usage
Error Details:
account hellotestman has insufficient ram; needs 189245 bytes has 5468 bytes
Jinhui-MacBook-Pro:eosio-cli-helper jinholee$ █
```