MUSA Token

Total fix supply: 10 Billion coin.

Decimal: 18.

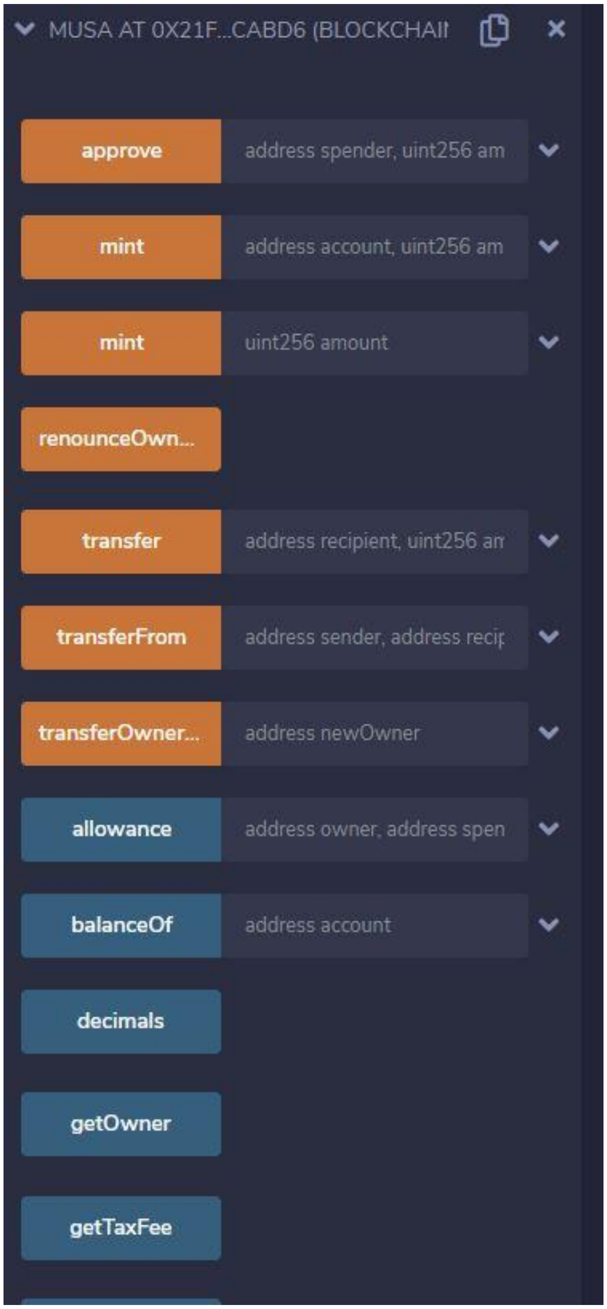
Supply: 1 Billion

Burnable: NO.

Mintable: YES.

Bep20 compliant: Yes.

Access type: Ownable.


MUSA AT 0X21F...CABD6 (BLOCKCHAII
approve
address spender, uint256 am
mint
address account, uint256 am
mint
uint256 amount
renounceOwn...
transfer
address recipient, uint256 an
transferFrom
address sender, address recip
transferOwner...
address newOwner
allowance
address owner, address spen
balanceOf
address account
decimals
getOwner
getTaxFee

Approve: require address of sender, and amount approve transaction.

Renounces ownership: For dev to transfer ownership to the owner

Transfer: Requirements:

* - `recipient` cannot be the zero address.

* - the caller must have a balance of at least `amount`.

* - Calls a unique reflection transfer

Transfer Ownership: For owner to transfer ownership to a new address/owner

Name: Returns the token name.

Symbol: Returns the token symbol.

Decimals: Returns the token decimals.

getOwner: Returns the bep token owner.

TotalSupply: See {BEP20-totalSupply}.

balanceOf: See {BEP20-balanceOf}.

Allowance: See {BEP20-allowance}.

TotalFees: The total fees generated using the transfer function

Approve: See {BEP20-approve}

* Requirements:

* - `spender` cannot be the zero address.

Transfer: See {BEP20-transfer}

* Requirements:

* - `recipient` cannot be the zero address.

* - the caller must have a balance of at least `amount`.

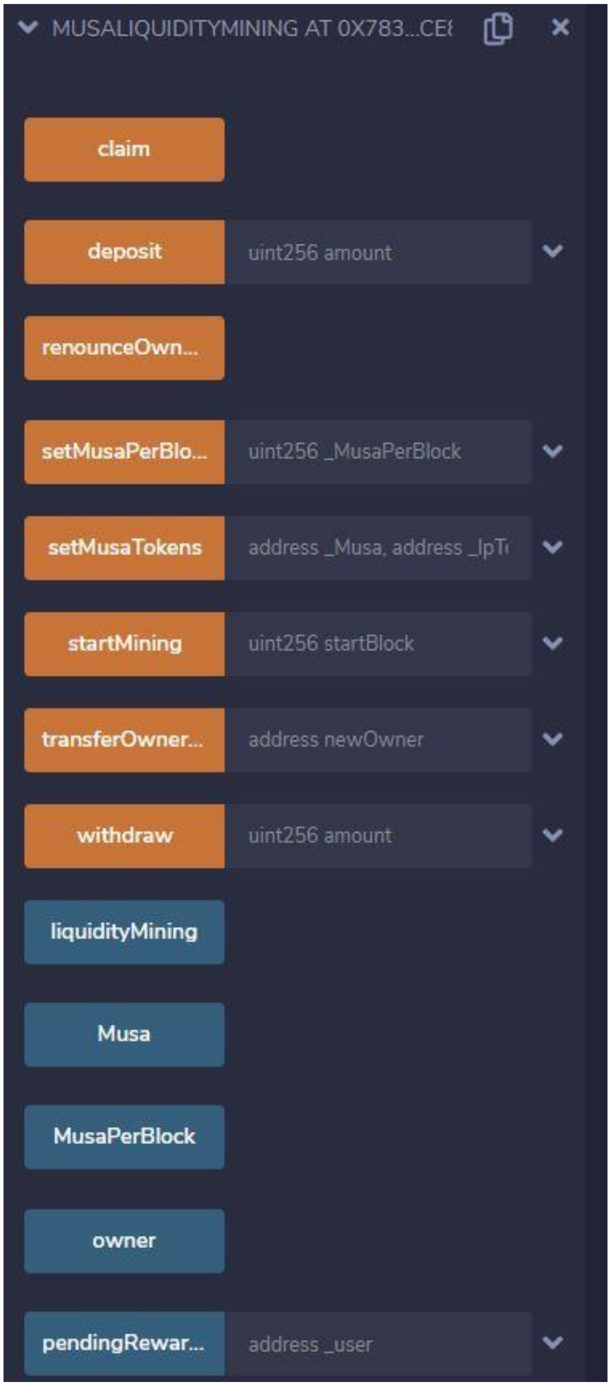
TransferFees: Transfers token with reflection fees

ReflectFee: Updates token's fee, burn, and total values

TaxFee: Returns the tax fee

Mint: Creates `amount` tokens and assigns them to `msg.sender`, increasing the total supply

Requirements: `msg.sender` must be the token owner.



Info of each user:

struct UserInfo {

uint256 amount;   // How many LP tokens the user has provided.

uint256 rewardDebt;   // Reward debt. See explanation below.

// Basically, any point in time, the amount of Musa entitled to a user but is pending to be distributed is pending reward = (user.amount * pool.accMusaPerShare) - user.rewardDebt

// Whenever a user deposits or withdraws LP tokens to a pool. Here's what happens:

1. The pool's `accMusaPerShare` (and `lastRewardBlock`) gets updated.
2. User receives the pending reward sent to his/her address.
3. User's `amount` gets updated.
4. User's `rewardDebt` gets updated.

```
// Info of each pool.
struct PoolInfo {
IBEP20 lpToken;          // Address of LP token contract.
uint256 lastRewardBlock;  // Last block number that Demos distribution occurs.
uint256 accMusaPerShare;   // Accumulated Demos per share, times 1e18. See below.
}
```

MusaPerBlock = Demo tokens created per block.

TotalFee: Deposit Fee address

Deposit: Total Musa in contract from deposits and rewards

poolInfo: Info of each pool.

mapping (uint256 => mapping (address => UserInfo)) public userInfo: Info of each user that stakes LP tokens.

StartBlock: The block number when Demo mining starts.

Claim: Claim any pending rewards.

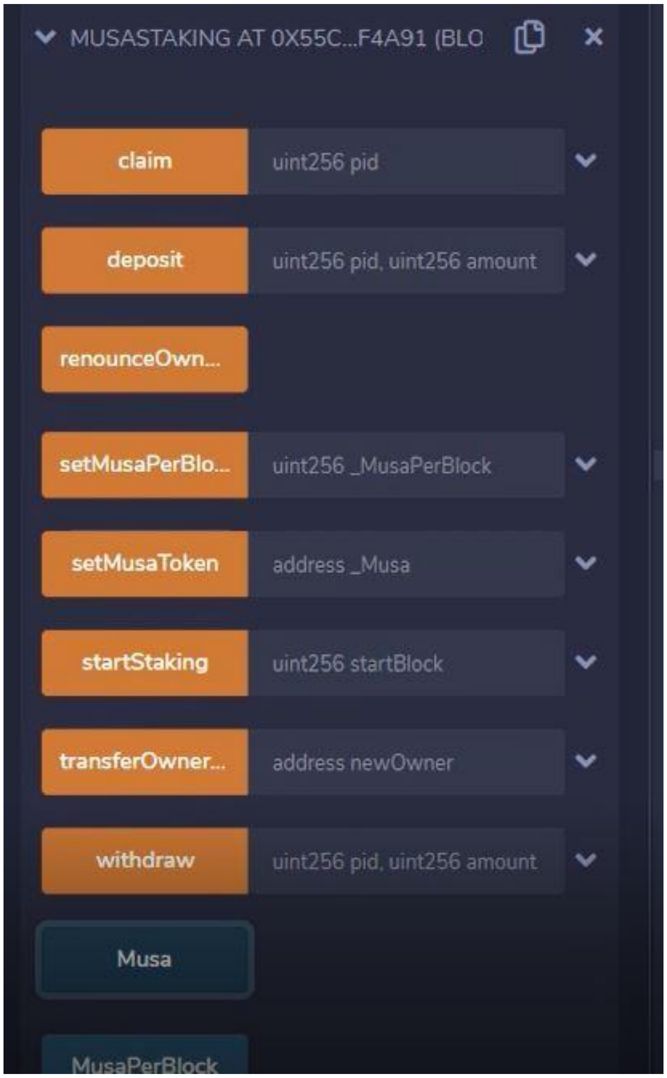
pendingReward: View function to see pending Musa on frontend. Require: Address

StartMining: For user who deposit and now want to start mining.

Renounces ownership: For dev to transfer ownership to the owner

SetMusaPerBlock: Number of amount of token per block.

Withdraw: Withdraw LP tokens from MusaLiquidity.



Info of each user:

struct UserInfo {

uint256 amount;   // How many LP tokens the user has provided.

uint256 rewardDebt;   // Reward debt. See explanation below.

uint256 pendingRewards; ;   // pending Reward if applicable.

uint256 lastClaim; //last reward claim token detail

```
// Info of each pool.
struct PoolInfo {
    uint256 allocPoint; // total number of points allocated
    uint256 lastRewardBlock; // Show the detail of last rewarded block
    uint256 accMusaPerShare; // Accumulated Musa per share, times 1e18. See below.
    uint256 depositedAmount; // Deposited amount in the stake poool
    uint256 rewardsAmount; // Accumulated reward generated in the required time
    uint256 lockupDuration; // Staking lock duration
}
```

Claim: Claim any pending rewards.

Startstaking: For user who deposit and now want to start staking.

Renounces ownership: For dev to transfer ownership to the owner.

SetMusaPerBlock: Number of amount of token per block.

Withdraw: Withdraw LP tokens from MusaLiquidity.

Deposit: Total Musa in contract from deposits and rewards.

poolInfo: Info of each pool.

TotalFees: The total fees generated using the transfer function.

Owner: Returns the address token owner.

UserInfo: Require: Address. Return user stake and reward info.