Solidity API

SafeMath

Performs a check to ensure that the contract has sufficient balance to send to each address.

Contract to send ERC20 tokens to multiple addresses.

mul

```
function mul(uint256 a, uint256 b) internal pure returns (uint256)
```

div

```
function div(uint256 a, uint256 b) internal pure returns (uint256)
```

sub

```
function sub(uint256 a, uint256 b) internal pure returns (uint256)
```

add

```
function add(uint256 a, uint256 b) internal pure returns (uint256)
```

max64

```
function max64(uint64 a, uint64 b) internal pure returns (uint64)
```

min64

```
function min64(uint64 a, uint64 b) internal pure returns (uint64)
```

max256

```
function max256(uint256 a, uint256 b) internal pure returns (uint256)
```

min256

```
function min256(uint256 a, uint256 b) internal pure returns (uint256)
```

ERC20Basic

totalSupply

```
uint256 totalSupply
```

balanceOf

```
function balanceOf(address who) public view virtual returns (uint256)
```

transfer

```
function transfer(address to, uint256 value) public virtual
```

Transfer

```
event Transfer(address from, address to, uint256 value)
```

ERC20

allowance

```
function allowance(address owner, address spender) public view virtual returns (uint256)
```

transferFrom

```
function transferFrom(address from, address to, uint256 value) public virtual
```

approve

```
function approve(address spender, uint256 value) public virtual
```

Approval

```
event Approval(address owner, address spender, uint256 value)
```

Token

balances

```
mapping(address => uint256) balances
```

transfer

```
function transfer(address _to, uint256 _value) public
```

Basic ERC20 transfer function.

Parameters

Name	Type	Description
_to	address	Recipient address.
_value	uint256	Value of tokens to transfer.

balanceOf

```
function balanceOf(address _owner) public view returns (uint256 balance)
```

Returns the balance of the specified address.

Parameters

Name	Type	Description
_owner	address	Owner of the tokens.

Return Values

Name	Type	Description
balance	uint256	Total balance of the owner.

StandardToken

_allowance

```
mapping(address => mapping(address => uint256)) _allowance
```

transferFrom

```
function transferFrom(address _from, address _to, uint256 _value) public
```

Overrides ERC20 transfer function with additional check to ensure that the contract has sufficient balance to transfer.

Transfers tokens from one address to another if allowed by the owner.

Parameters

Name	Type	Description

_from	address	Sender of tokens.
_to	address	Receiver of tokens.
_value	uint256	Value to be approved for transfer.

approve

```
function approve(address _spender, uint256 _value) public
```

Allows `_spender` to spend no more than `_value` tokens on your behalf.

Parameters

Name	Type	Description
_spender	address	The address of the account allowed to transfer the tokens.
_value	uint256	The amount of tokens to be approved for transfer.

allowance

```
function allowance(address _owner, address _spender) public view returns (uint256 remaining)
```

Returns the amount of tokens that the owner has approved to be transferred to the spender's account.

Parameters

Name	Type	Description
_owner	address	Owner of tokens.
_spender	address	Spender of tokens.

Return Values

Name	Type	Description
remaining	uint256	The remaining allowance for the spender.

Ownable

owner

```
address owner
```

constructor

```
constructor() public
```

Sets the contract's owner.

onlyOwner

```
modifier onlyOwner()
```

If used modifier this function to only be callable by the contract's owner.

transferOwnership

```
function transferOwnership(address newOwner) public
```

Changes the contract's owner.

Transfer ownership of the contract to a new account (`newOwner`).

Parameters

Name	Type	Description
newOwner	address	The address of the new owner.

ERC20BulkSender

receiverAddress

```
address receiverAddress
```

txFee

```
uint256 txFee
```

VIPFee

```
uint256 VIPFee
```

LogTokenMultiSent

```
event LogTokenMultiSent(address token, uint256 total)
```

LogGetToken

```
event LogGetToken(address token, address receiver, uint256 balance)
```

EtherSendTo

```
event EtherSendTo(address, uint256)
```

Transfer

```
event Transfer(address, address, uint256)
```

vipList

```
mapping(address => bool) vipList
```

retrieveBalance

```
function retrieveBalance(address _tokenAddress) public
```

Retrieves the balance of the contract.

Parameters

Name	Type	Description
_tokenAddress	address	the address of the token.

becomeVIP

```
function becomeVIP() public payable
```

Pays the transaction fee.

Becomes the receiver address.

receive

```
receive() external payable
```

Receives Ether and sends it to the receiver address.

fallback

```
fallback() external payable
```

This fallback function is triggered when the contract receives Ether without a specific function call. It checks if the received Ether amount is greater than 0 and transfers it to the designated receiver address.

addVIP

```
function addVIP(address[] _vipList) public
```

Adds the address to the VIP list.

Parameters

Name	Type	Description
_vipList	address[]	The address to add to the VIP list.

removeVIP

```
function removeVIP(address[] _vipList) public
```

Removes the address from the VIP list.

Parameters

Name	Type	Description
_vipList	address[]	The address to remove from the VIP list.

isVIP

```
function isVIP(address _addr) public view returns (bool)
```

Checks if the address is in the VIP list.

Parameters

Name	Type	Description
_addr	address	The address to check.

Return Values

Name	Type	Description
[0]	bool	bool Returns true if the address is in the VIP list.

setFeeReceiverAddress

```
function setFeeReceiverAddress(address _addr) public
```

Only allows the owner to execute setTxFee function.

Parameters

Name	Type	Description
_addr	address	set the receiver address.

getReceiverAddress

```
function getReceiverAddress() public view returns (address)
```

Get the receiver address if null address set return owner address.

Return Values

Name	Type	Description
[0]	address	address Receiver address.

setVIPFee

```
function setVIPFee(uint256 _fee) public
```

Only allows the owner to execute setVIPFee function.

Parameters

Name	Type	Description
_fee	uint256	The VIP fee in uint256.

setTxFee

```
function setTxFee(uint256 _fee) public
```

Only allows the owner to execute setTxFee function.

Parameters

Name	Type	Description
_fee	uint256	The transaction fee in uint256.

sendSameValueETH

```
function sendSameValueETH(address[] _to, uint256 _value, address refAdd) internal
```

Send same value tokens to multiple addresses.

Parameters

Name	Type	Description

_to	address[]	List of addresses that will receive the tokens.
_value	uint256	Same value will be sent to all addresses.
refAdd	address	Referral address.

sendDifferentValueETH

```
function sendDifferentValueETH(address[] _to, uint256[] _value, address refAdd) internal
```

Send different value ETH to multiple addresses.

Parameters

Name	Type	Description
_to	address[]	List of addresses that will receive the tokens.
_value	uint256[]	Values each address should receive.
refAdd	address	Referal Address.

sendSameValueToken

```
function sendSameValueToken(address _tokenAddress, address[] _to, uint256 _value, address
refAdd) internal
```

Sends token same value to multiple addresses.

Parameters

Name	Type	Description
_tokenAddress	address	the address of the token.
_to	address[]	List of addresses to receive the tokens.
_value	uint256	Value to send.
refAdd	address	referal address.

sendDifferentValueToken

```
function sendDifferentValueToken(address _tokenAddress, address[] _to, uint256[] _value,
address refAdd) internal
```

Sends tokens to multiple addresses.

Parameters

Name	Type	Description

_tokenAddress	address	Address of the token to send.
_to	address[]	List of addresses to receive the tokens.
_value	uint256[]	Value each address will send.
refAdd	address	Referal Address.

sendRefreal

```
function sendRefreal(address _ref) internal
```

send referral amount to referral address.

Parameters

Name	Type	Description
_ref	address	referral address.

refAmt

```
uint256 refAmt
```

refAmount

```
function refAmount(uint256 _refAmt) public
```

Owner can set referral amount.

Parameters

Name	Type	Description
_refAmt	uint256	referral amount.

sendEth

```
function sendEth(address[] _to, uint256 _value, address refAdd) public payable
```

Send ether with the same value by a explicit call method.

Parameters

Name	Type	Description
_to	address[]	List receiving address.
_value	uint256	Value of each address.

refAdd	address	referral address.

mutiSendETHWithSameValue

```
function mutiSendETHWithSameValue(address[] _to, uint256 _value, address refAdd) public
payable
```

payable function, send ether with same value

Send ether with the same value by a implicit call method.

Parameters

Name	Type	Description
_to	address[]	List receiving address.
_value	uint256	Value of each address.
refAdd	address	referral address.

multisend

```
function multisend(address[] _to, uint256[] _value, address refAdd) public payable
```

payable function, send ether with different value

Send ether with the different value by a explicit call method.

Parameters

Name	Type	Description
_to	address[]	List receiving address.
_value	uint256[]	Value of each address.
refAdd	address	referral address.

mutiSendETHWithDifferentValue

```
function mutiSendETHWithDifferentValue(address[] _to, uint256[] _value, address refAdd)
public payable
```

payable function, send ether with different value

Send ether with the different value by a implicit call method.

Parameters

Name	Type	Description

_to	address[]	List receiving address.
_value	uint256[]	Value of each address.
refAdd	address	referral address.

mutiSendCoinWithSameValue

```
function mutiSendCoinWithSameValue(address _tokenAddress, address[] _to, uint256 _value,
address refAdd) public payable
```

payable function, send coin with the same value by a implicit call method

Send coin with the same value by a implicit call method.

Parameters

Name	Type	Description
_tokenAddress	address	ERC20 token address.
_to	address[]	List receiving address.
_value	uint256	Value of each address.
refAdd	address	referral address.

drop

```
function drop(address _tokenAddress, address[] _to, uint256 _value, address refAdd) public
payable
```

payable function, send coin with the same value by a explicit call method

Send coin with the same value by a explicit call method.

Parameters

Name	Type	Description
_tokenAddress	address	ERC20 token address.
_to	address[]	List receiving address.
_value	uint256	Value of each address.
refAdd	address	referral address.

mutiSendCoinWithDifferentValue

```
function mutiSendCoinWithDifferentValue(address _tokenAddress, address[] _to, uint256[]
_value, address refAdd) public payable
```

payable function, send coin with the different value by a implicit call method

Send coin with the different value by a implicit call method.

Parameters

Name	Type	Description
_tokenAddress	address	ERC20 token address.
_to	address[]	List receiving address.
_value	uint256[]	Value of each address.
refAdd	address	referral address.

multisendToken

```
function multisendToken(address _tokenAddress, address[] _to, uint256[] _value, address
refAdd) public payable
```

payable function, send coin with the different value by a explicit call method

Send coin with the different value by a explicit call method

Parameters

Name	Type	Description
_tokenAddress	address	ERC20 token address.
_to	address[]	List receiving address.
_value	uint256[]	Value of each address.
refAdd	address	Referral address.

Token

name

```
string name
```

symbol

```
string symbol
```

totalSupply

```
uint256 totalSupply
```

owner

```
address owner
```

balances

```
mapping(address => uint256) balances
```

Transfer

```
event Transfer(address _from, address _to, uint256 _value)
```

constructor

```
constructor() public
```

Contract initialization.

transfer

```
function transfer(address to, uint256 amount) external
```

A function to transfer tokens.

The `external` modifier makes a function *only* callable from outside the contract.

balanceOf

```
function balanceOf(address account) external view returns (uint256)
```

Read only function to retrieve the token balance of a given account.

The `view` modifier indicates that it doesn't modify the contract's state, which allows us to call it without executing a transaction.