

This file is represented as MultiSignatureWithToken contract document.

FUNCTIONS

setOwnerStatus(newOwnerAddress)

@PARAMS : newOwnerStatus => string that represents the user address.

[illegible]

setTransactionStatus(amount, receiver)

@PARAMS : amount => number of coins in WEI wich will be transfer to the receiver.

receiver => string that represents the user address.

NOTICE : WEI is the smallest part of the whole ETH, which means 10 to the power of 18.

i.e if user wants to set 'amount' to 2 ETH, you should pass 2000000000000000000 to this function

[illegible]

transferCoinsToReceiver(transactionId)

@PARAMS : transactionId => id of the transaction status that user wants to vote.

NOTICE : transcation id must be successful and you have to check it, if it was successful, pop up a menu for all owners to call this function. only one owner can call this.

[illegible]

voteYesToAddOwner(ownerStatusId)

@PARAMS : ownerStatusId => id of the owner status that user wants to vote.

[illegible]

voteNoToAddOwner(ownerStatusId)

@PARAMS : ownerId => id of the owner status that user wants to vote.

[illegible]

`voteYesToTxStatus(transactionId)`

@PARAMS : transactionId => id of the transaction status that user wants to vote.

[illegible]

voteNoToTxStatus(transactionId)

@PARAMS : transactionId => id of the transaction status that user wants to vote.

[illegible]

getOwner(index)

@RETURN : it returns an address which is one of the owners.

getOwnerStatusInfo(ownerStatusId)

@RETURN : it returns an array which is like this => [

notVotes,

allVotesSoFar,

neededVotes,

status

]

getTxStatusInfo(transactionId)

@PARAMS : transactionId => id of the transaction status that user wants to vote.

@RETURN : it returns an array which is like this => [

receiver;

amountOfCoins;

```
noVotes;
```

```
yesVotes;
```

allVotesSoFar;

neededVotes => always equal $\frac{2}{3}$ quantity of owners i.e 4 owners will be 3 needed votes

isTransferred => a boolean which shows that already transferred for the receiver.

```
status;
```

]

NOTICE : status has only 3 values => 1 => means pending , 2 => means successful , 3 => means failed

[illegible]

depositToken(amount)

@PARAMS : amount => a number which represents the amount of token to be transferred to the contract;

NOTICE : Before users calling this function, they should approve the contract address using approve function in MML contract.

```
## i.e => MMLtoken.approve(contractAddress, amount)
```

withdraw()

NOTICE : if there were coins that were blocked in the contract, one of the owners can withdraw it.

[illegible][illegible]

getAllOwners()

NOTICE: this function will return all owners in an array.