This file is represented as MultiSignature contract document.

FUNCTIONS

setOwnerStatus(newOwnerAddress)

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

>>

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

>>

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.

>>

voteYesToAddOwner(ownerStatusId)

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

>>

voteNoToAddOwner(ownerStatusId)

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

>>

voteYesToTxStatus(transactionId)

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

>>

voteNoToTxStatus(transactionId)

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

>>

getOwner(index)

@PARAMS : index => a number which be the index of an array.

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

>>>

getOwnerStatusInfo(ownerStatusId)

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

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

addressToAdd,

notVotes,

yesVotes

allVotesSoFar,

neededVotes,

status

]

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

>>>

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 2/3 quantity of owners i.e 4 owners will be 3 needed votes

isTransfered => a boolean which shows that already transfered for the receiver.

status;

]

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

>>

deposit()

NOTICE : this function will transfer native coins to contract. User calls this function and you have to transfer their 'WEI' to the contract;

>>>

getAllOwners()

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