

UniBlock Documentation

Release 0.0.1

UniBlock

May 13, 2019

1	UniBlock Main	1
2	UniBlock Comunnication	3
3	UniBlock Blockchain	3
	Python Module Index	7
	Index	9

UniBlock Main

`main.main ()`

Função principal do programa

`main.parseArguments ()`

Função que identifica os argumentos passados

Return parser objetos contendo os argumentos

UniBlock Comunnication

UniBlock Comunnication

class Blockchain.BlockChain

classe pai da blockchain implementa os metodos essenciais para a blockchain

chain

getter da chain

static hash (block)

metodo estatico que gera a hash do bloco

last_block

metodo para retornar o ultimo bloco da chain

last_proof

retorna a ultima prova de trabalho adicionada na chain

rule

metodo que retorna a regra da prova de trabalho

valid_chain (chain)

confere se a chain eh valida atraves das hashes da chain

static valid_proof (last_proof, proof, rule)

metodo estatico que valida a proof gerada

class Blockchain.MinerChain

classe que estende a classe Blockchain classe que implementa os metodos da chain utilizada pelos mineradores da blockchain

current_transactions

metodo getter para retornar as transacoes atuais

finish_transactions

metodo getter para retornar as transacoes fechadas

mine ()

minera a carteira se ja estiver pronto para minerar muda a flag para false e retorna o block minerado

new_block (*proof*, *previous_hash*=None)

Cria um novo bloco com as informacoes

new_transaction (*transaction*)

metodo que recebe uma nova transacao e adiciona nas transacoes atuais se o numero maximo de transacoes da carteira for atingido uma nova carteira eh adicionada na lista de transacoes

proof_of_work (*last_proof*)

metodo de prova de trabalho determina a dificuldade de minerar um block

start_miner

metodo getter para retornar o valor da flag `_start_miner` responsavel por dizer (return True) quando uma carteira esta pronta para ser minerada

transactions

metodo getter para retornar as transacoes

class Blockchain.TraderChain

classe que implementa a chain dos traders

new_transaction (*myIp*)

cria uma nova transacao que sera enviada para a carteira ativa

b

Blockchain, [5](#)

c

comunnication, [3](#)

m

main, [1](#)

B

BlockChain (class in BlockChain), 5
BlockChain (module), 5

C

chain (BlockChain.BlockChain attribute), 5
comunnication (module), 3
current_transactions (BlockChain.MinerChain attribute), 5

F

finish_transactions (BlockChain.MinerChain attribute), 6

H

hash() (BlockChain.BlockChain static method), 5

L

last_block (BlockChain.BlockChain attribute), 5
last_proof (BlockChain.BlockChain attribute), 5

M

main (module), 1
main() (in module main), 1
mine() (BlockChain.MinerChain method), 6
MinerChain (class in BlockChain), 5

N

new_block() (BlockChain.MinerChain method), 6
new_transaction() (BlockChain.MinerChain method), 6
new_transaction() (BlockChain.TraderChain method), 6

P

parseArguments() (in module main), 1
proof_of_work() (BlockChain.MinerChain method), 6

R

rule (BlockChain.BlockChain attribute), 5

S

start_miner (BlockChain.MinerChain attribute), 6

T

TraderChain (class in BlockChain), 6
transactions (BlockChain.MinerChain attribute), 6

V

valid_chain() (BlockChain.BlockChain method), 5
valid_proof() (BlockChain.BlockChain static method), 5

