Globatalent

DECENTRALIZED SPORTS

Globatalent smart contract documentation

V0.03 - 04/24/2018

Globatalent smart contract documentation

CROWDSALE

Understanding the Ethereum contracts for the Globatalent crowdsale.

TECHNICAL INFORMATION

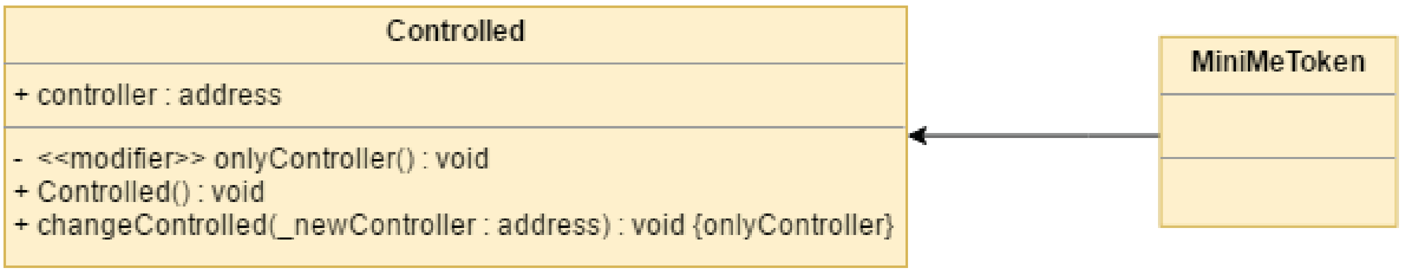
ERC20-compliant token, derived from the **MiniMe Token** that allows for token cloning (forking).

CONTRACTS

Utilities

Controlled.sol

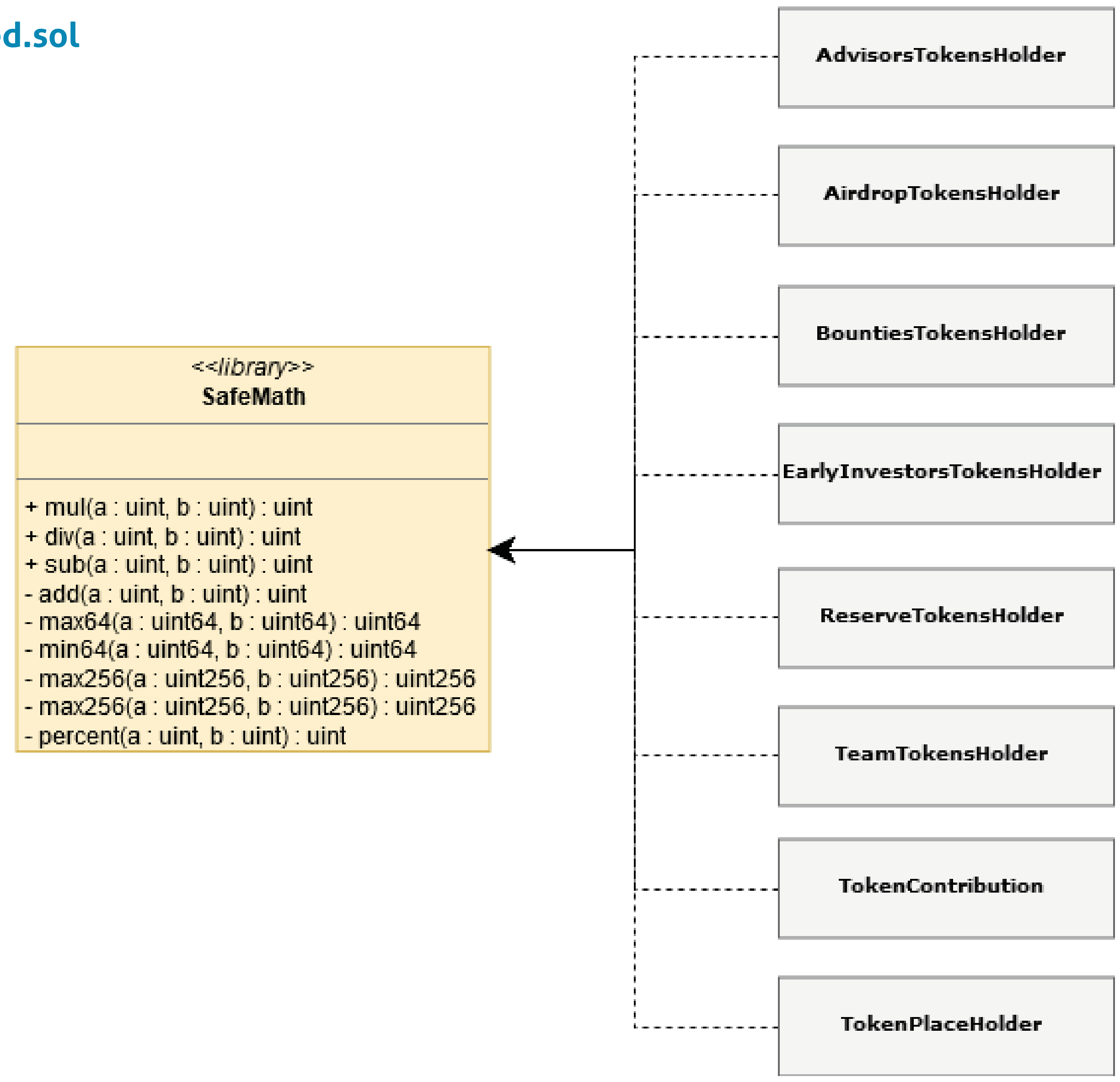
This contract provides useful methods to establish the address that can call functions of another contract.


Controlled
+ controller : address
- <<modifier>> onlyController() : void
+ Controlled() : void
+ changeControlled(_newController : address) : void {onlyController}
MiniMeToken

• onlyController() [L6]: Checks that the address calling the function equals the controller address.
• controller [L8]: Controller address.
• Controlled() [L10]: Contract constructor. Sets controller variable to the address that creates the contract.
• changeController(newController) [L14-L16]: The actual controller can call this function to change controller to newController.

Globatalent smart contract documentation

Owned.sol

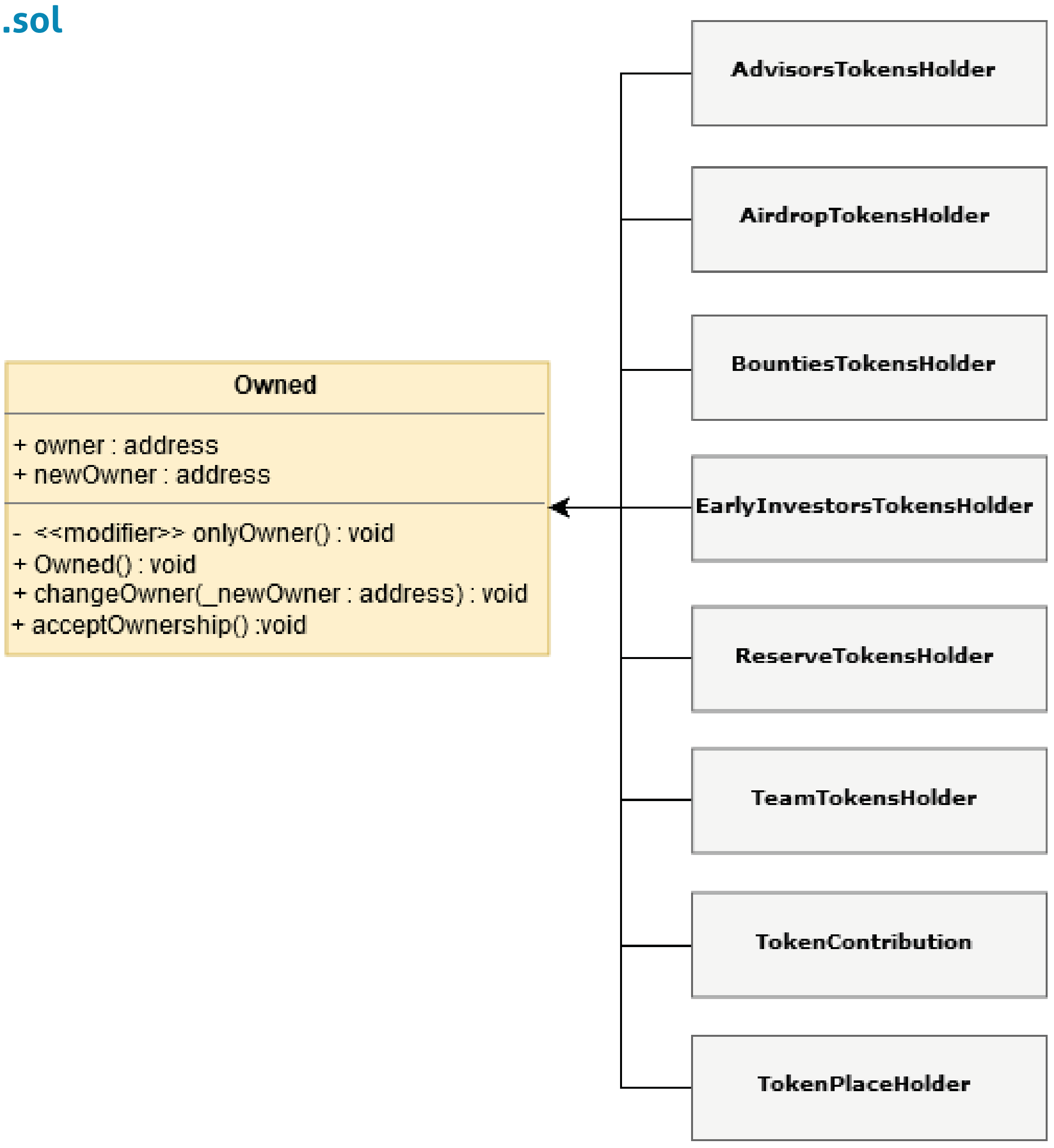


<<library>>
SafeMath
+ mul(a : uint, b : uint) : uint
+ div(a : uint, b : uint) : uint
+ sub(a : uint, b : uint) : uint
- add(a : uint, b : uint) : uint
- max64(a : uint64, b : uint64) : uint64
- min64(a : uint64, b : uint64) : uint64
- max256(a : uint256, b : uint256) : uint256
- max256(a : uint256, b : uint256) : uint256
- percent(a : uint, b : uint) : uint
AdvisorsTokensHolder
AirdropTokensHolder
BountiesTokensHolder
EarlyInvestorsTokensHolder
ReserveTokensHolder
TeamTokensHolder
TokenContribution
TokenPlaceHolder

• onlyOwner() [L10]: Checks that the address calling the function equals the owner address.
• owner [L15]: Owner address.
• Owned() [L19]: Contract constructor. Sets owner variable to to the address that creates the contract.
• newOwner [L22]: Address of the new owner.
• changeOwner(newOwner) [L27-29]: Old owner can call this function to set a newOwner.
• acceptOwnership() [L32-L36]: If there is a new owner and calls this function the new owner is accepted as the owner of the contract.

Globatalent smart contract documentation

SafeMath.sol



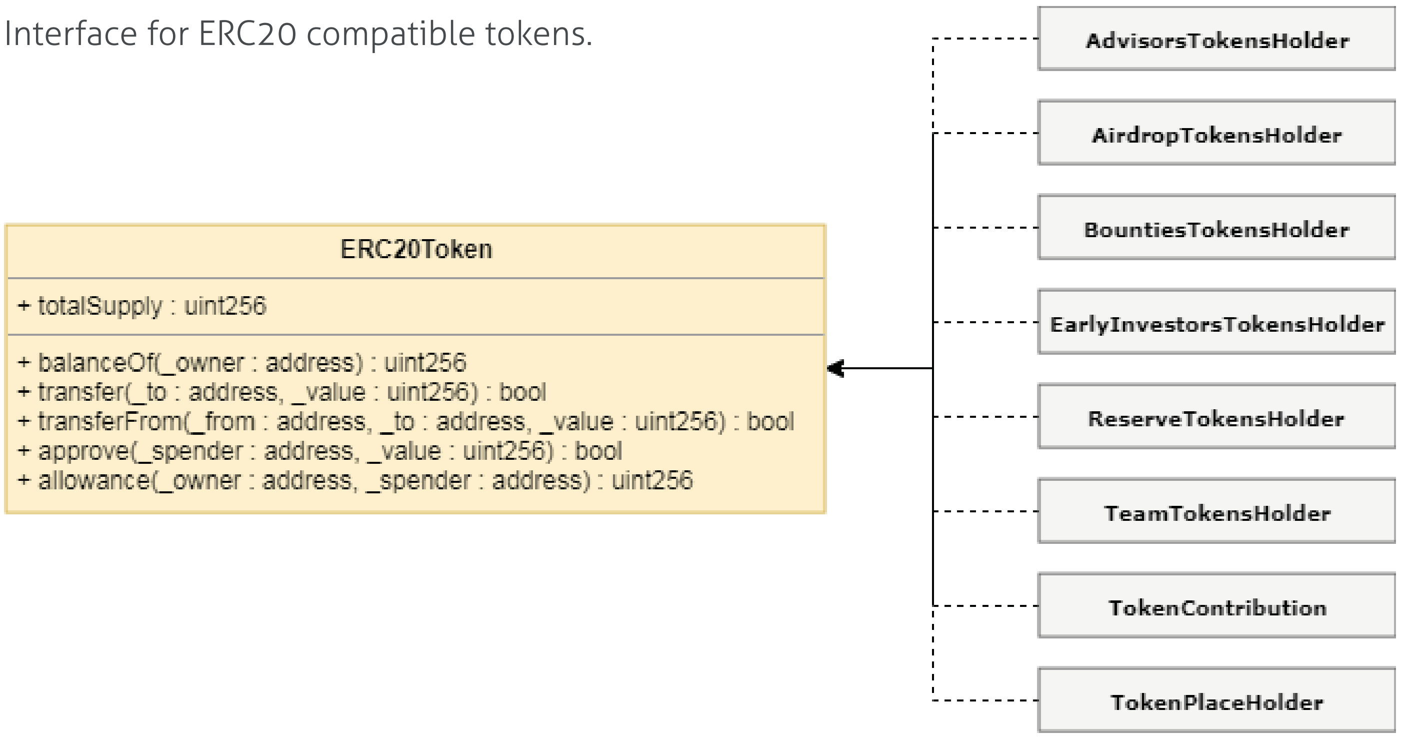
- mul(a, b) [L8-12]: Multiplies two numbers checking for overflows.
- div(a, b) [L14-19]: Divides two numbers checking for overflows and zero divisions.
- sub(a, b) [L21-24]: Does a subtraction checking a is greater or equal than b.
- max64(a, b) [L32-34]: Checks the maximum between two 64 space numbers.
- min64(a, b) [L32-34]: Checks the maximum between two 64 space numbers.
- max256(a, b) [L40-42]: Checks the maximum between two 256 space numbers.
- min256(a, b) [L4-46]: Checks the maximum between two 256 space numbers.
- percent(a, b) [L48-50]: Does the percentage between a and b.

Globatalent smart contract documentation

Token specific

ERC20Token.sol

Interface for ERC20 compatible tokens.

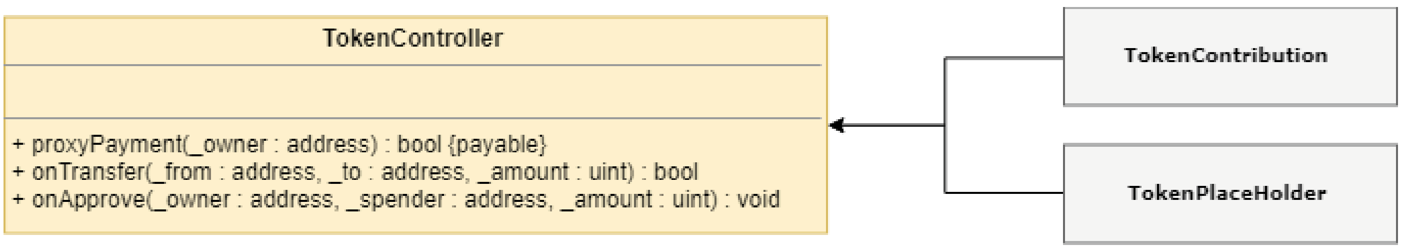

ERC20Token
+ totalSupply : uint256
+ balanceOf(_owner : address) : uint256
+ transfer(_to : address, _value : uint256) : bool
+ transferFrom(_from : address, _to : address, _value : uint256) : bool
+ approve(_spender : address, _value : uint256) : bool
+ allowance(_owner : address, _spender : address) : uint256
AdvisorsTokensHolder
AirdropTokensHolder
BountiesTokensHolder
EarlyInvestorsTokensHolder
ReserveTokensHolder
TeamTokensHolder
TokenContribution
TokenPlaceHolder

- totalSupply [L16]: Variable that contains the total supply of the token.
- balanceOf(owner) [L20]: Returns the token balance of the owner.
- transfer(to, value) [L26]: Transfers value amount of tokens from your account to to another account.
- transferFrom(from, to, value) [L33]: Transfers value amount of tokens from one account to another.
- approve(spender, value) [L39]: Approves someone to spend value amount of tokens from your account.
- allowance(owner, spender) [L44]: Checks the amount of tokens a person has authorized another to spend from his account.
- Contract events [L46-47]: Events to log event from smart contract

Globatalent smart contract documentation

TokenController.sol

Our token contract allows a token controller, this way we can upgrade functionality and modify it's behaviour.

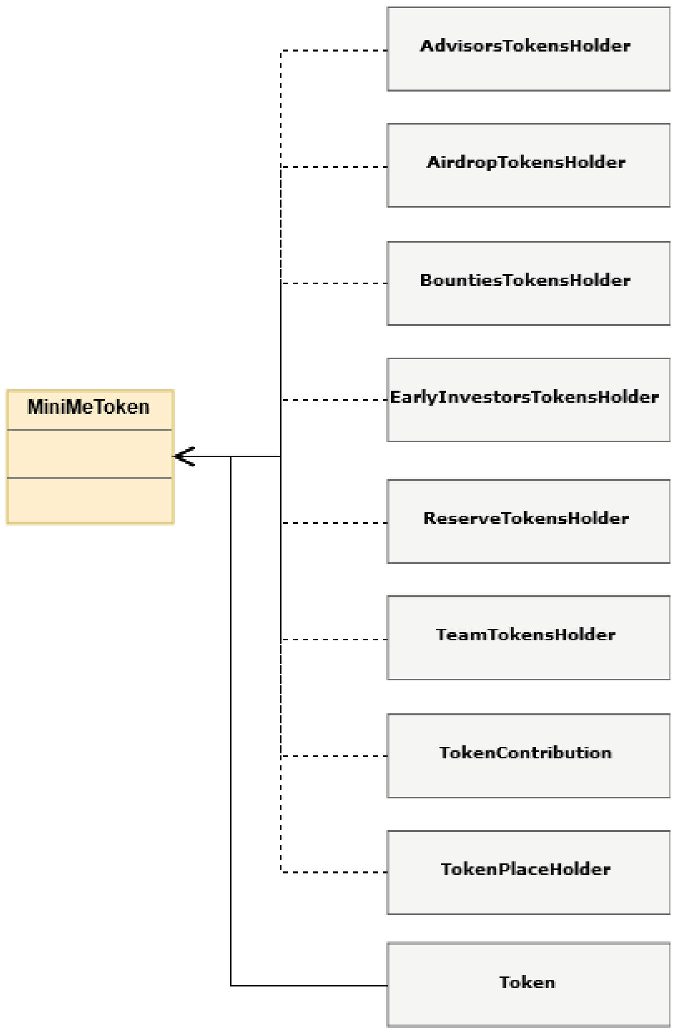

TokenController
+ proxyPayment(_owner : address) : bool {payable}
+ onTransfer(_from : address, _to : address, _amount : uint) : bool
+ onApprove(_owner : address, _spender : address, _amount : uint) : void
TokenContribution
TokenPlaceHolder

• proxyPayment(owner) [L8]: Method to call sending ether to purchase tokens.
• onTransfer(from, to, amount) [L16]: Method called when transferring tokens, let's us control when tokens can start to be tradable.
• onApprove(owner, spender, amount) [L24]: Method called when someone approves another person to spend tokens from his account.

MiniMeToken.sol

Inherits from: Controlled.

MiniMe token follows the ERC20 standard and implements all the functions described on ERC20-Token.sol but since more logic is required by this token it breaks ERC20 functions into more than one method.



Globatalent smart contract documentation

• name [L40]: Name of the token.
• decimals [L41]: Decimals of the token.
• symbol [L42]: Symbol of the token.
• version [L43]: Version of the token.
• Checkpoint [L49-56]: Struct with two variables which is used to store the amount of token supply at a certain block.
• parentToken [L60]: Address from which the token has been cloned, if it's not been cloned it will be 0x0.
• parentSnapShotBlock [L64]: Block number to determine the initial distribution of the parent token if actual token was cloned.
• creationBlock [L67]: Block number when the token was created.
• balances [L72]: Map that tracks the amount of token of/to an address
• allowed [L75]: Map that stores the amount an address can transfer from another account.
• totalSupplyHistory [L78]: Array that stores the history of the token at certain blocknumbers.
• transfersEnabled [L81]: Determines if the token is transferable.
• tokenFactory [L84]: Factory used to create new clone tokens.
• MiniMeToken(tokenFactory, parentToken, parentSnapShotBlock, tokenName, decimalUnits, tokenSymbol, transfersEnabled) [L103]: Contract constructor.

See MINIME_README.md for more information.

[ERC20 methods]

• doTransfer [L167]: Actual transfer function in this contract. Inside it, balances are checked. The TokenController is warned that a transfer has been done and it checks that transfers are enabled.
• totalSupply(blockNumber) [L270]: Returns the total supply of tokens at blockNumber.
• balanceOfAt(owner, blockNumber) [L283]: Returns the amount of tokens of owner at blockNumber.
• createCloneToken(cloneTokenName, cloneDecimalUnits, cloneTokenSymbol, snapshotBlock, transfersEnabled) [L344-366]: Creates a new clone token with the initial distribution being this token at snapshotBlock.

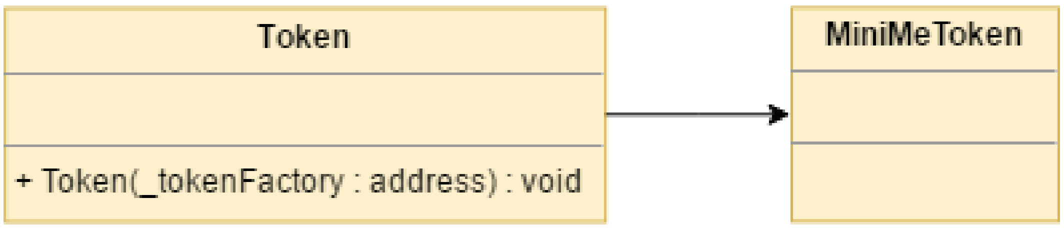
Globatalent smart contract documentation

[ERC20 methods]

• generateTokens(owner, amount) [L376-386]: amount number of tokens are asigned to owner. This method can only be called from the controller of the token.
• destroyTokens(owner, amount) [L393-403]: Burns amount tokens from owner.
• enableTransfers(transfersEnabled) [L412-414]: Sets transfers enabled.
• getValueAt(checkpoints, block) [L424-445]: Returns the number of tokens at a given number block.
• updateValueAtNow(checkpoints, value) [L451-461]: Updates the history of balances.
• isContract(addr) [L467-474]: Internal function to determine if addr is a contract. min(a, b) [L477-479]: Determine the minimum between two uints.
• Default function [L484-487]: The fallback function: If the contract's controller has not been set to 0, then the proxyPayment method is called which relays the ether and creates tokens as described in the token controller contract.
• claimToken(token) [L497-507]: This method can be used by the controller to extract mistakenly sent tokens to this contract.
• Events [L512-521]: Events used by the contract to emit values.
• MiniMeTokenFactory [L531-563]: This contract is used to generate clone contracts from a contract..

Token.sol

Inherits from: MiniMeToken. This creates an instance of MiniMeToken with the desired parameters to customize the token.



Globatalent smart contract documentation

Contribution

TokenController.sol

Inherits from: **Owned.sol**
TokenController.
Uses **SafeMath** for uint256 operations.

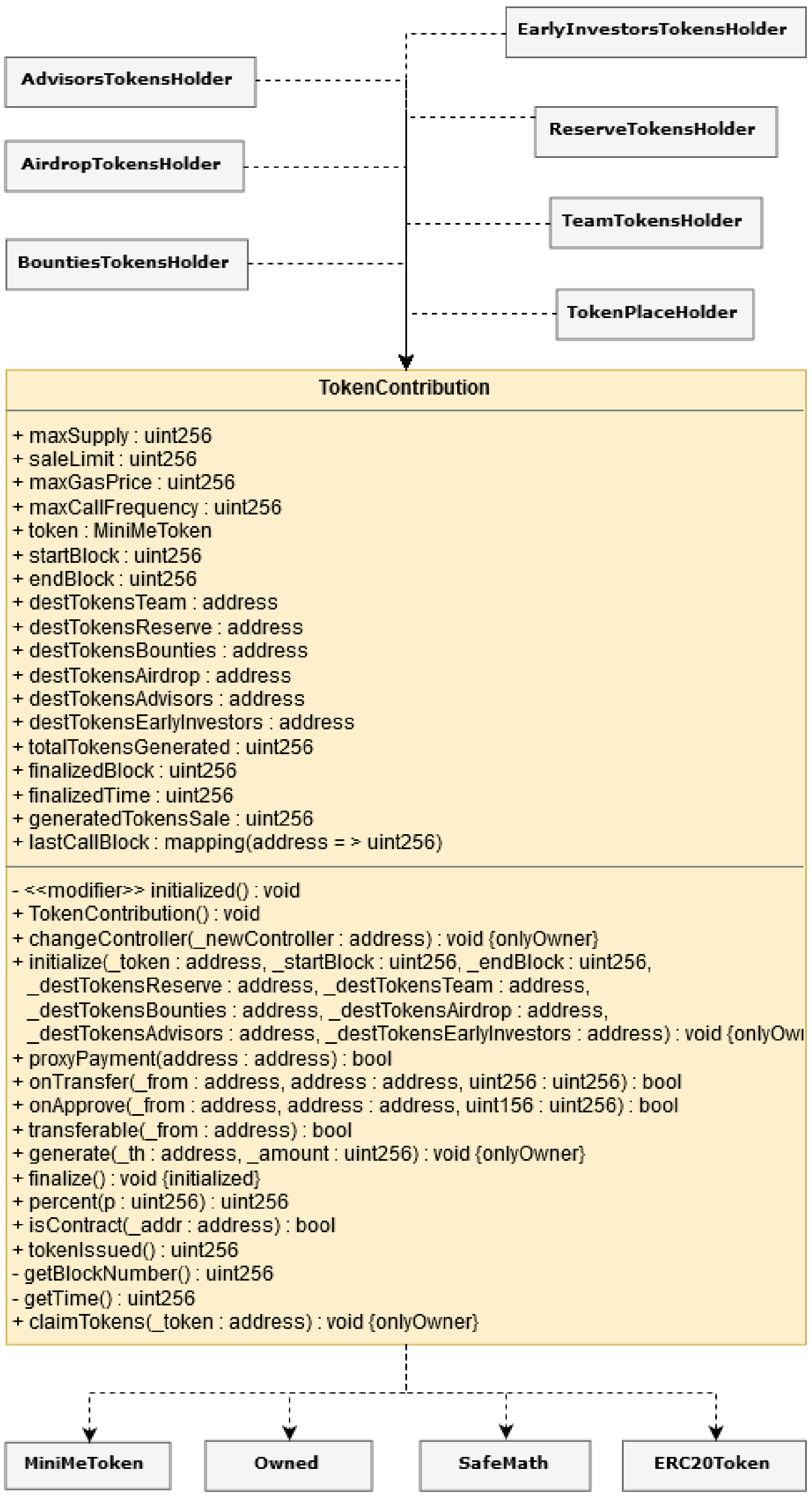
This is the contract where the contribution takes place. The first part of the contract declares the variables where the address of the holders will be stored, token, sets the sale limit and the maximum supply of the token.

• [L11] SafeMath is specified for the type uint256, this guarantees that all the operations done with uint256 are not in danger of an overflow.
• initialized() [L40-43]: Modifier to check if the token contribution is initialized.
• TokenContribution() [L45-46]: Contract constructor.
• changeController(newController) [L51-54]: The owner can call this method to set the new controller of the contract.
• initialize(token, destTokensReserve, destTokensTeam, desTokensBounties, destTokensAirdrop, destTokensAdvisors, destTokensEarlyInvestors) [L59-93]: The owner of the contract can call this function to initialize the token contribution.

[TokenController methods]

• generate(th, amount) [L118-126]: The owner of the contract can call this method to generate amount tokens to th address.
• finalize() [L150-226]: The owner of the contract can call this method to finalize the contribution. After this method is called the correspondent tokens are sent to their holders.
• percent(p) [L228-230]: Does the percentage of a number.
• isContract(addr) [L235-242]: Checks if addr is a contract.
• tokensIssued() [L250-252]: Returns total tokens issued.
• getBlockNumber() [L260-262]: Returns the block number.
• getTime() [L265-267]: Returns current time.
• claimTokens(token) [L278-291]: This method can be used by the controller to extract mistakenly sent tokens to this contract.
• Events [L293-299]: Events used by the contract to emit values.

Globatalent smart contract documentation


EarlyInvestorsTokensHolder
AdvisorsTokensHolder
ReserveTokensHolder
AirdropTokensHolder
TeamTokensHolder
BountiesTokensHolder
TokenPlaceHolder
TokenContribution
+ maxSupply : uint256
+ saleLimit : uint256
+ maxGasPrice : uint256
+ maxCallFrequency : uint256
+ token : MiniMeToken
+ startBlock : uint256
+ endBlock : uint256
+ destTokensTeam : address
+ destTokensReserve : address
+ destTokensBounties : address
+ destTokensAirdrop : address
+ destTokensAdvisors : address
+ destTokensEarlyInvestors : address
+ totalTokensGenerated : uint256
+ finalizedBlock : uint256
+ finalizedTime : uint256
+ generatedTokensSale : uint256
+ lastCallBlock : mapping(address = > uint256)
- <<modifier>> initialized() : void
+ TokenContribution() : void
+ changeController(_newController : address) : void {onlyOwner}
+ initialize(_token : address, _startBlock : uint256, _endBlock : uint256,
_destTokensReserve : address, _destTokensTeam : address,
_destTokensBounties : address, _destTokensAirdrop : address,
_destTokensAdvisors : address, _destTokensEarlyInvestors : address) : void {onlyOw
+ proxyPayment(address : address) : bool
+ onTransfer(_from : address, address : address, uint256 : uint256) : bool
+ onApprove(_from : address, address : address, uint156 : uint256) : bool
+ transferable(_from : address) : bool
+ generate(_th : address, _amount : uint256) : void {onlyOwner}
+ finalize() : void {initialized}
+ percent(p : uint256) : uint256
+ isContract(_addr : address) : bool
+ tokenIssued() : uint256
- getBlockNumber() : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
Owned
SafeMath
ERC20Token

Globatalent smart contract documentation

TokenHolders

All token holders follow the same structure, the only difference is the timing of the collect-Tokens function:

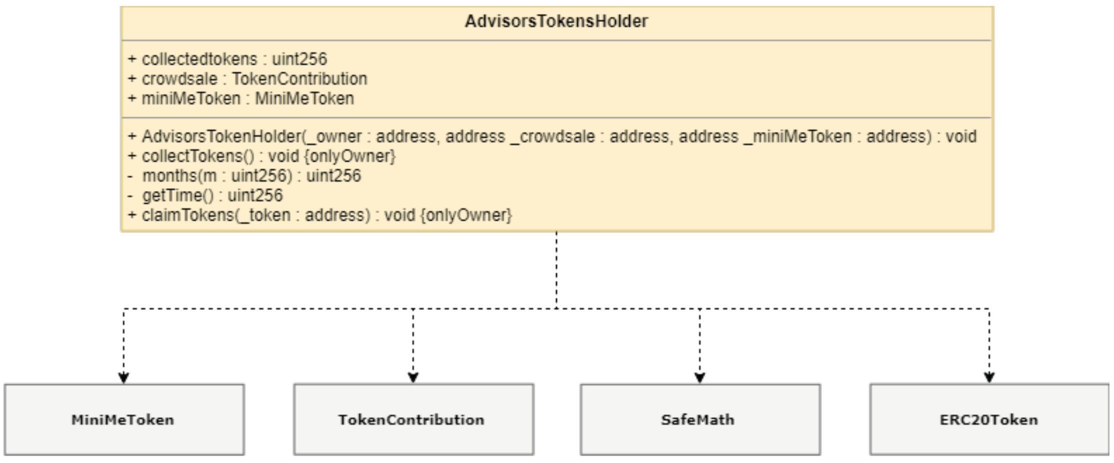
Inherits from: **Owned.sol** Uses SafeMath for uint256 operations.

- collectedTokens: Amount of tokens collected from the holder.
- crowdsale: Address of the token contribution used to check the finalized time.
- Minimetoken: Address of the token to do the transfers.
- Holder constructor (owner, crowdsale, minimetoken): Constructor of the contract.
- collectTokens(): The owner of the contract can call this function to collect tokens. If no tokens are available to collect at the time of call an exception is thrown. The locking period of the tokens differs from the holder. If there are tokens to collect, the amount is transferred to the owner account.
- months(m): Returns m amount of months. This is used inside collectTokens to check if the locking period is over.
- getTime(): Returns current time.
- claimTokens(token): This method can be used by the controller to extract mistakenly sent tokens to this contract.
- Events: Events used by the contract to emit values.

AdvisorsTokensHolder.sol

[TokenHolder methods]

- collectTokens(): [TokenHolder functionality] - Unlock 20% after 2 months; Then unlock 20% every month.

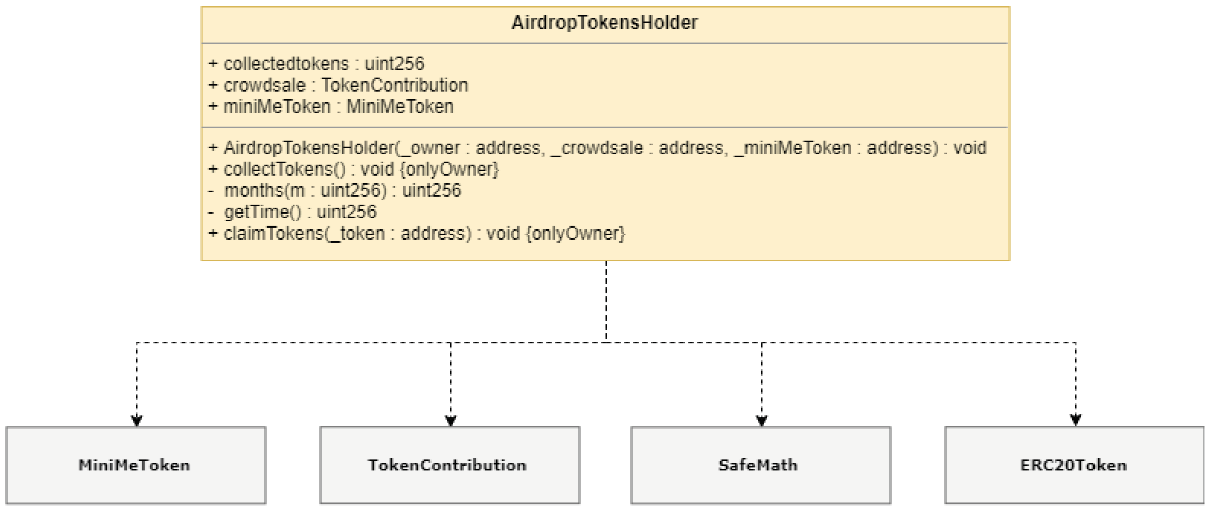

AdvisorsTokensHolder
+ collectedtokens : uint256
+ crowdsale : TokenContribution
+ miniMeToken : MiniMeToken
+ AdvisorsTokenHolder(_owner : address, address _crowdsale : address, address _miniMeToken : address) : void
+ collectTokens() : void {onlyOwner}
- months(m : uint256) : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
TokenContribution
SafeMath
ERC20Token

Globatalent smart contract documentation

AirdropTokensHolder.sol

[TokenHolder methods]

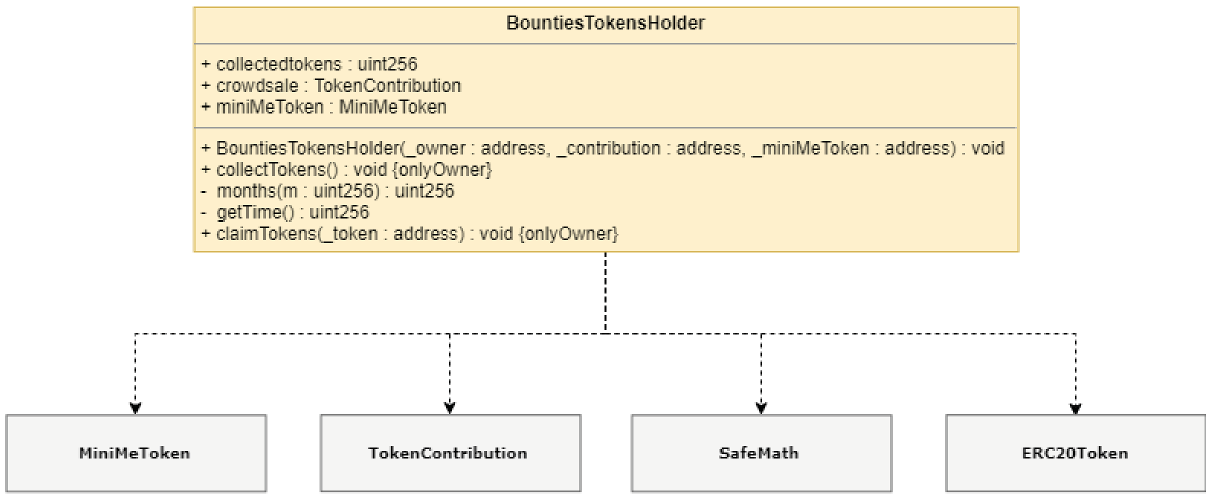
• collectTokens(): [TokenHolder functionality] - Unlock 25% tokens every 3 months from ICO date.


AirdropTokensHolder
+ collectedtokens : uint256
+ crowdsale : TokenContribution
+ miniMeToken : MiniMeToken
+ AirdropTokensHolder(_owner : address, _crowdsale : address, _miniMeToken : address) : void
+ collectTokens() : void {onlyOwner}
- months(m : uint256) : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
TokenContribution
SafeMath
ERC20Token

BountiesTokensHolder.sol

[TokenHolder methods]

• collectTokens(): [TokenHolder functionality] - No lock-up.

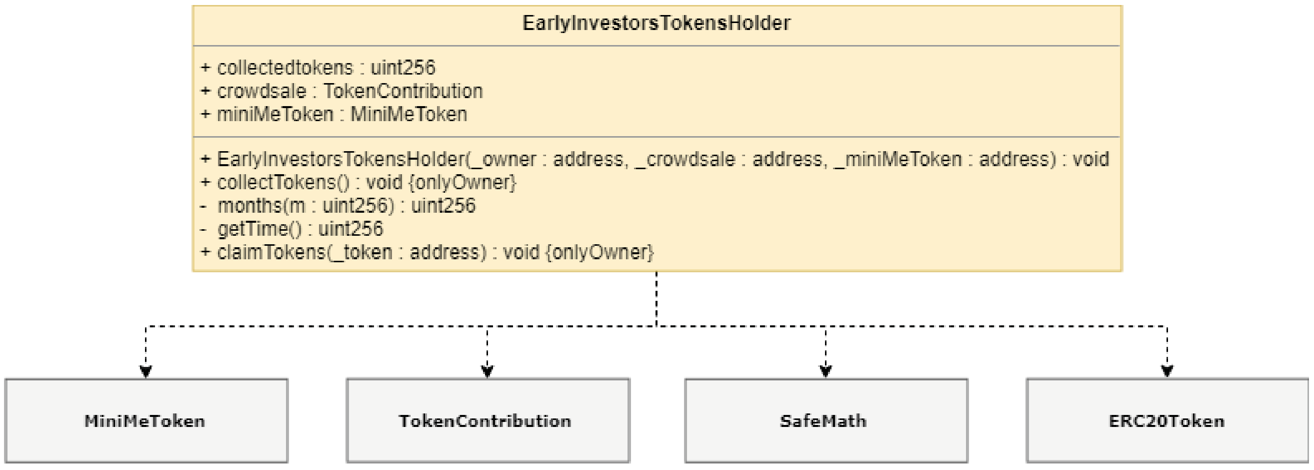

BountiesTokensHolder
+ collectedtokens : uint256
+ crowdsale : TokenContribution
+ miniMeToken : MiniMeToken
+ BountiesTokensHolder(_owner : address, _contribution : address, _miniMeToken : address) : void
+ collectTokens() : void {onlyOwner}
- months(m : uint256) : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
TokenContribution
SafeMath
ERC20Token

Globatalent smart contract documentation

EarlyInvestorsTokensHolder.sol

[TokenHolder methods]

• collectTokens(): [TokenHolder functionality] - Unlock 40% 1 day after Token Launch; Then unlock 20% every 3 months.

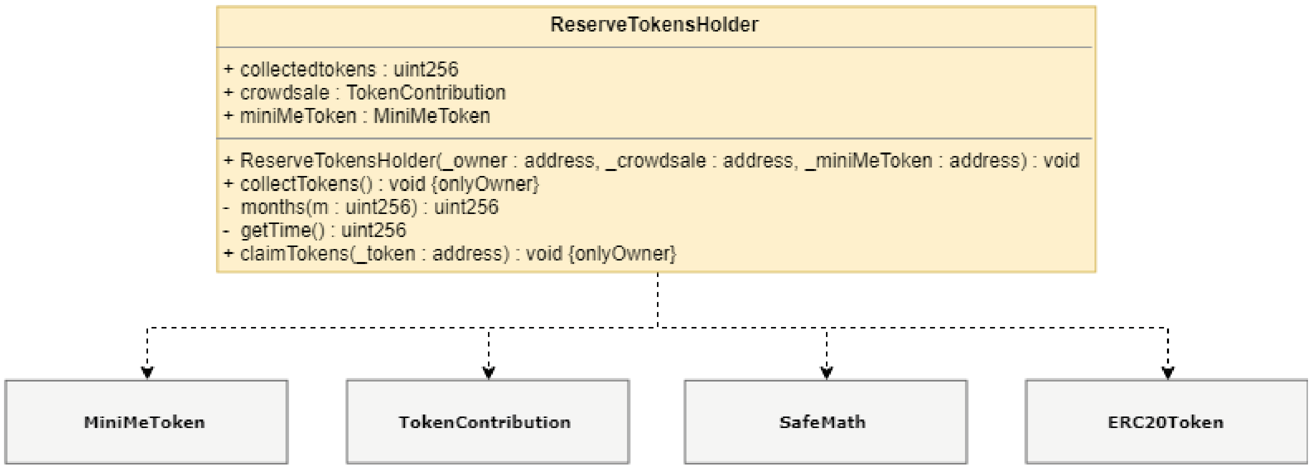


EarlyInvestorsTokensHolder
+ collectedtokens : uint256
+ crowdsale : TokenContribution
+ miniMeToken : MiniMeToken
+ EarlyInvestorsTokensHolder(_owner : address, _crowdsale : address, _miniMeToken : address) : void
+ collectTokens() : void {onlyOwner}
- months(m : uint256) : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
TokenContribution
SafeMath
ERC20Token

ReserveTokensHolder.sol

[TokenHolder methods]

• collectTokens(): [TokenHolder functionality] - Unlock 50% after 18 months; 100% after 36 months.



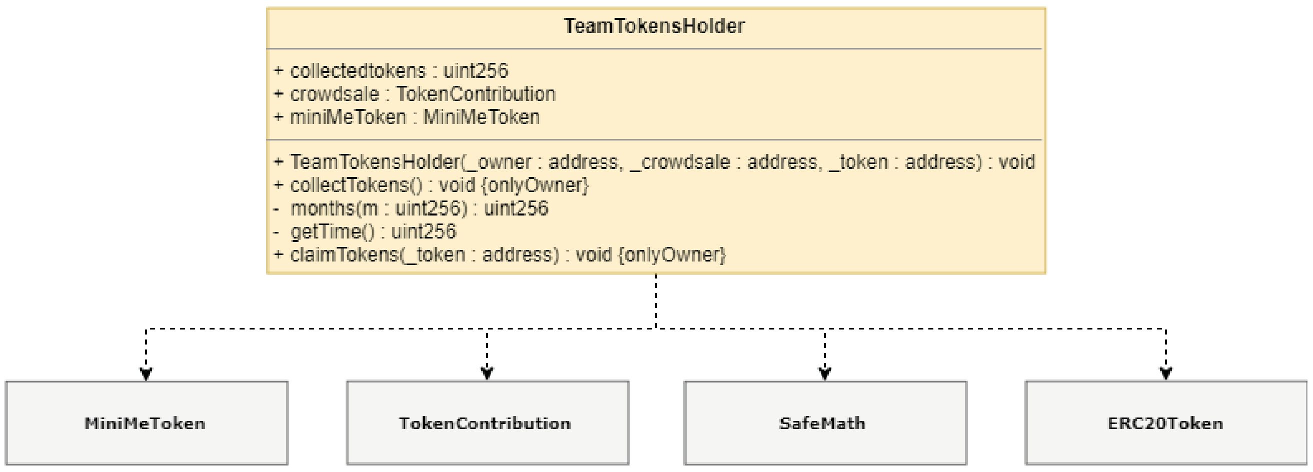
ReserveTokensHolder
+ collectedtokens : uint256
+ crowdsale : TokenContribution
+ miniMeToken : MiniMeToken
+ ReserveTokensHolder(_owner : address, _crowdsale : address, _miniMeToken : address) : void
+ collectTokens() : void {onlyOwner}
- months(m : uint256) : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
TokenContribution
SafeMath
ERC20Token

Globatalent smart contract documentation

TeamTokensHolder.sol

[TokenHolder methods]

• collectTokens(): [TokenHolder functionality] - Unlock 40% after 12 months; Unlock 80% after 24 months; Unlock 100% after 36 months.


TeamTokensHolder
+ collectedtokens : uint256
+ crowdsale : TokenContribution
+ miniMeToken : MiniMeToken
+ TeamTokensHolder(_owner : address, _crowdsale : address, _token : address) : void
+ collectTokens() : void {onlyOwner}
- months(m : uint256) : uint256
- getTime() : uint256
+ claimTokens(_token : address) : void {onlyOwner}
MiniMeToken
TokenContribution
SafeMath
ERC20Token

TEST AND DEPLOY

See **INSTRUCTIONS.md** for instructions on how to test and deploy the contracts.

Credits

Authors: Iván Martín, Javier Domínguez, Óscar López, Alejandro Martínez.
Revision: Robert Spitz, Ali Hararwala

https://github.com/Globatalent/crowdsale

v0.03 - Dated April 24, 2018

Globatalent
DECENTRALIZED SPORTS

More info:

WEBSITE

WHITE PAPER

JOIN US