

SOLIDITY COMPILER

COMPILER +

0.5.5+commit.47a71e8f

☐ Include nightly builds

☐ Auto compile

☐ Hide warnings

Advanced Configurations

Compile

KaseiCoinCrowdsale.sol

Compile and Run script

CONTRACT

KaseiCoin (KaseiCoin.sol)

Publish on Ipfs

Publish on Swarm

Compilation Details

ABI

Bytecode

1pragma solidity ^0.5.0;

2

3import "../KaseiCoin.sol";

4import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/Crowdsale.sol";

5import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/emission/MintedCrowdsale.sol";

6

7

8// Have the KaseiCoinCrowdsale contract inherit the following OpenZeppelin:

9// * Crowdsale

10// * MintedCrowdsale

11contract KaseiCoinCrowdsale { // UPDATE THE CONTRACT SIGNATURE TO ADD INHERITANCE

12

13// Provide parameters for all of the features of your crowdsale, such as the `rate`, `wallet` for fundraising, and `token`.

14constructor{

15// YOUR CODE HERE!

16uint rate,

17address payable wallet,

18KaseiCoin token

19} //Crowdsale(rate, wallet, token)

20public

21{

22// constructor can stay empty

23}

24}

25

26/*

27contract KaseiCoinCrowdsaleDeployer {

28// Create an `address public` variable called `kasei_token_address`.

29address public kasei_token_address;

30// Create an `address public` variable called `kasei_crowdsale_address`.

31address public kasei_crowdsale_address;

32

33// Add the constructor.

34constructor{

35string memory name,

36string memory symbol,

37

38}

39}

40// Create a new instance of the KaseiCoin contract.

41KaseiCoin token = new KaseiCoin(name, symbol, 0);

42

Code deposit cost: 10600 gas, Execution cost: 10854 gas

FunctionDefinition

1 reference(s)

0

☐ listen on all transactions

Search with transaction hash or address

Welcome to Remix 0.25.3

Your files are stored in IndexedDB, 10.68 MB / 285.8 GB used

You can use this terminal to:

- Check transactions details and start debugging.
- Execute JavaScript scripts:
 - Input a script directly in the command line interface
 - Select a Javascript file in the file explorer and then run `\`remix.execute()\`` or `\`remix.executeCurrent()\`` in the command line interface
 - Right click on a Javascript file in the file explorer and then click `\`Run\``

The following libraries are accessible:

- web3 version 1.5.2
- ethers.js
- remix

Type the library name to see available commands.

SOLIDITY COMPILER

COMPILER +

0.5.5+commit.47a71e8f

☐ Include nightly builds

☐ Auto compile

☐ Hide warnings

Advanced Configurations

Compile

KaseiCoinCrowdsale.s
ol

Compile and
Run script

CONTRACT

KaseiCoin (KaseiCoin.sol)

Publish on Ipfs

Publish on Swarm

Compilation Details

ABI

Bytecode

1pragma solidity ^0.5.0;

2

3import "../KaseiCoin.sol";

4import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/Crowdsale.sol";

5import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/emission/MintedCrowdsale.sol";

6

7

8// Have the KaseiCoinCrowdsale contract inherit the following OpenZeppelin:

9// * Crowdsale

10// * MintedCrowdsale

11contract KaseiCoinCrowdsale is Crowdsale, MintedCrowdsale { // UPDATE THE CONTRACT SIGNATURE TO ADD INHERITANCE

12

13// Provide parameters for all of the features of your crowdsale, such as the `rate`, `wallet` for fundraising, and `token`.

14constructor{

15// YOUR CODE HERE!

16uint rate,

17address payable wallet,

18KaseiCoin token

19} Crowdsale(rate, wallet, token)

20public

21{

22// constructor can stay empty

23}

24}

25

26

27contract KaseiCoinCrowdsaleDeployer {

28// Create an `address public` variable called `kasei_token_address`.

29address public kasei_token_address;

30// Create an `address public` variable called `kasei_crowdsale_address`.

31address public kasei_crowdsale_address;

32

33// Add the constructor.

34constructor{

35string memory name,

36string memory symbol,

37address payable wallet

38}

39}

40// Create a new instance of the KaseiCoin contract.

41KaseiCoin token = new KaseiCoin(name, symbol, 0);

42

Code deposit cost: 10600 gas, Execution cost: 10854 gas

FunctionDefinition

1 reference(s)

0

☐ listen on all transactions

Search with transaction hash or address

Welcome to Remix 0.25.3

Your files are stored in IndexedDB, 10.68 MB / 285.8 GB used

You can use this terminal to:

- Check transactions details and start debugging.
- Execute JavaScript scripts:
 - Input a script directly in the command line interface
 - Select a Javascript file in the file explorer and then run `\`remix.execute()\`` or `\`remix.executeCurrent()\`` in the command line interface
 - Right click on a Javascript file in the file explorer and then click `\`Run\``

The following libraries are accessible:

- web3 version 1.5.2

Type the library name to see available commands.

DEPLOY & RUN TRANSACTIONS ✓

ENVIRONMENT
Remix VM (London) ⓘ

ACCOUNT
0x5B93...eddC4 ⓘ

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
KaseiCoinCrowdsaleDeploy

Deploy string name

Publish to IPFS

OR

At Address Load contract

Transactions recorded ⓘ

Deployed Contracts

▼ KASEICOINCROWDSALEDEPLOYER ⓘ

kasei_crowdsale...
0: address: 0xb843c36718eCb3390750727736ba14f174277

kasei_token_a...
0: address: 0xc9b5d06c2c183efc52285c0a356f1169660

Low level interactions ⓘ

CALLDATA

Transact

0 listen on all transactions Search with transaction hash or address

```
1 pragma solidity ^0.5.0;
2
3 import "../KaseiCoin.sol";
4 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/Crowdsale.sol";
5 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/emission/MintedCrowdsale.sol";
6
7 // Have the KaseiCoinCrowdsale contract inherit the following OpenZeppelin:
8 // * Crowdsale
9 // * MintedCrowdsale
10 contract KaseiCoinCrowdsale is Crowdsale, MintedCrowdsale { // UPDATE THE CONTRACT SIGNATURE TO ADD INHERITANCE
11
12 // Provide parameters for all of the features of your crowdsale, such as the 'rate', 'wallet' for fundraising, and 'token'.
13 constructor(
14 // YOUR CODE HERE!
15 uint rate,
16 address payable wallet,
17 KaseiCoin token
18 ) Crowdsale(rate, wallet, token)
19 public
20 {
21 // constructor can stay empty
22 }
23 }
24
25
26 contract KaseiCoinCrowdsaleDeployer {
27 // Create an 'address public' variable called 'kasei_token_address'.
28 address public kasei_token_address;
29 // Create an 'address public' variable called 'kasei_crowdsale_address'.
30 address public kasei_crowdsale_address;
31
32 // Add the constructor.
33 constructor(
34 string memory name,
35 string memory symbol,
36 address payable wallet
37 ) public {
38 // Create a new instance of the KaseiCoin contract.
39 KaseiCoin token = new KaseiCoin(name, symbol, 0);
40
41 // Assign the token contract's address to the 'kasei_token_address' variable.
42 kasei_token_address = address(token);
43 }
```

call to KaseiCoinCrowdsaleDeployer.kasei_crowdsale_address

call [call] from: 0x58380a6a701c568545dcfc803fc8875f56beddC4 to: KaseiCoinCrowdsaleDeployer.kasei_crowdsale_address() data: 0x0f0...b4619

call to KaseiCoinCrowdsaleDeployer.kasei_token_address

call [call] from: 0x58380a6a701c568545dcfc803fc8875f56beddC4 to: KaseiCoinCrowdsaleDeployer.kasei_token_address() data: 0x2ab...6a2a1

DEPLOY & RUN TRANSACTIONS ✓

Transactions recorded ⓘ ⓘ

Deployed Contracts

▼ KASEICOINCROWDSALEDEPLOYER ⓘ

kasei_crowdsale...
0: address: 0xb843c36718eCb3390750727736ba14f174277

kasei_token_a...
0: address: 0xc9b5d06c2c183efc52285c0a356f1169660

Low level interactions ⓘ

CALLDATA

Transact

▼ KASEICOINCROWDSALE AT 0x8D1 ⓘ

buyTokens

beneficiary: 0x9555bC3c5650CE94aAdA61B7f8A38e0836589E4 ⓘ

Transact

rate

0: uint256: 1

token

0: address: 0x740B5a956346BD5FC15474A8880a9C9b13e8a9

wallet

0: address: 0x9655bC3c5650CE94aAdA61B7f8A38e0836589E4

weiRaised

0: uint256: 10

Low level interactions ⓘ

CALLDATA

Transact

▼ KASEICOIN AT 0xF74_3EBA9 (BLD) ⓘ

Home KaseiCoin.sol KaseiCoinCrowdsale.sol

```
1 pragma solidity ^0.5.0;
2
3 import "../KaseiCoin.sol";
4 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/Crowdsale.sol";
5 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/emission/MintedCrowdsale.sol";
6
7 // Have the KaseiCoinCrowdsale contract inherit the following OpenZeppelin:
8 // * Crowdsale
9 // * MintedCrowdsale
10 contract KaseiCoinCrowdsale is Crowdsale, MintedCrowdsale { // UPDATE THE CONTRACT SIGNATURE TO ADD INHERITANCE
11
12 // Provide parameters for all of the features of your crowdsale, such as the 'rate', 'wallet' for fundraising, and 'token'.
13 constructor(
14 // YOUR CODE HERE!
15 uint rate,
16 address payable wallet,
17 KaseiCoin token
18 ) Crowdsale(rate, wallet, token)
19 public
20 {
21 // constructor can stay empty
22 }
23 }
24
25
26 contract KaseiCoinCrowdsaleDeployer {
27 // Create an 'address public' variable called 'kasei_token_address'.
28 address public kasei_token_address;
29 // Create an 'address public' variable called 'kasei_crowdsale_address'.
30 address public kasei_crowdsale_address;
31
32 // Add the constructor.
33 constructor(
34 string memory name,
35 string memory symbol,
36 address payable wallet
37 ) public {
38 // Create a new instance of the KaseiCoin contract.
39 KaseiCoin token = new KaseiCoin(name, symbol, 0);
40
41 // Assign the token contract's address to the 'kasei_token_address' variable.
42 kasei_token_address = address(token);
43 }
```

0 listen on all transactions Search with transaction hash or address

call [call] from: 0x9555bC3c5650CE94aAdA61B7f8A38e0836589E4 to: KaseiCoinCrowdsale.rate() data: 0x2c4...e722e

call to KaseiCoinCrowdsale.weiRaised

call [call] from: 0x9555bC3c5650CE94aAdA61B7f8A38e0836589E4 to: KaseiCoinCrowdsale.weiRaised() data: 0x404...2b66f

DEPLOY & RUN TRANSACTIONS

0: address: 0x740bba9b634618a13b4744a8f80bA9C9b13eb9f

wallet

0: address: 0x95655bC3c5650CF94aAdA61B7fB438e0836589E4

weiRaised

0: uint256: 10

Low level interactions

CALLDATA

Transact

▼ KASEICOIN AT 0XF74_3EBA9 (BLC)

addMinter address account ▼

approve address spender, uint256 i ▼

decreaseAllow... address spender, uint256 i ▼

increaseAllow... address spender, uint256 i ▼

mint address account, uint256 a ▼

renounceMinter

transfer address recipient, uint256 ▼

transferFrom address sender, address re ▼

allowance address owner, address sp ▼

balanceOf address account ▼

decimals

isMinter address account ▼

name

symbol

totalSupply

0: uint256: 10

Low level interactions

```

1  pragma solidity ^0.5.0;
2
3  import "../KaseiCoin.sol";
4  import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/Crowdsale.sol";
5  import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/emission/MintedCrowdsale.sol";
6
7
8  // Have the KaseiCoinCrowdsale contract inherit the following OpenZeppelin:
9  // * Crowdsale
10 // * MintedCrowdsale
11 contract KaseiCoinCrowdsale is Crowdsale, MintedCrowdsale { // UPDATE THE CONTRACT SIGNATURE TO ADD INHERITANCE
12
13     // Provide parameters for all of the features of your crowdsale, such as the "rate", "wallet" for fundraising, and "token".
14     constructor(
15         // YOUR CODE HERE!
16         uint rate,
17         address payable wallet,
18         KaseiCoin token
19     ) Crowdsale(rate, wallet, token)
20     public
21     {
22         // constructor can stay empty
23     }
24 }
25
26
27 contract KaseiCoinCrowdsaleDeployer {
28     // Create an "address public" variable called "kasei_token_address".
29     address public kasei_token_address;
30     // Create an "address public" variable called "kasei_crowdsale_address".
31     address public kasei_crowdsale_address;
32
33     // Add the constructor.
34     constructor(
35         string memory name,
36         string memory symbol,
37         address payable wallet
38     ) public {
39         // Create a new instance of the KaseiCoin contract.
40         KaseiCoin token = new KaseiCoin(name, symbol, 0);
41
42         // Assign the token contract's address to the "kasei_token_address" variable.
43         kasei_token_address = address(token);
44     }
45 }

```

0 ☐ listen on all transactions

Search with transaction hash or address

call [call] from: 0x95655bC3c5650CF94aAdA61B7fB438e0836589E4 to: KaseiCoin.totalSupply() data: 0x181...60ddd

call to KaseiCoinCrowdsale.weiRaised

call [call] from: 0x95655bC3c5650CF94aAdA61B7fB438e0836589E4 to: KaseiCoinCrowdsale.weiRaised() data: 0x404...2b66f