**KaseiCoinCrowdsale.sol**

Home | KaselCoinCrowdsale.sol | KaseiCoin.sol

```
pragma solidity ^0.5.0;

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


// Have the KaseiCoinCrowdsale contract inherit the following OpenZeppelin:
// * Crowdsale
// * MintedCrowdsale
contract KaseiCoinCrowdSale is Crowdsale, MintedCrowdsale { // UPDATE THE CONTRACT SIGNATURE TO ADD INHERITANCE

    // Provide parameters for all of the features of your crowdsale, such as the `rate`, `wallet` for fundraising, and `token`.
    constructor(uint rate, address payable wallet, KaseiCoin token) Crowdsale(rate, wallet, token) public {
    }
}


contract KaseiCoinCrowdSaleDeployer {
    // Create an `address public` variable called `kasei_token_address`.
    address public kasei_token_address;
    // Create an `address public` variable called `kasei_crowdsale_address`.
    address public kasei_crowdsale_address;


    // Add the constructor.
    constructor(string memory name, string memory symbol, address payable wallet) public {
        // Create a new instance of the KaseiCoin contract.
        KaseiCoin token = new KaseiCoin(name, symbol, 0);

        // Assign the token contract's address to the `kasei_token_address` variable.
        kasei_token_address = address(token);


        // Create a new instance of the `KaseiCoinCrowdsale` contract
        KaseiCoinCrowdSale crowdsale = new KaseiCoinCrowdSale(1, wallet, token);
```

```
37
38          // Aassign the `KaseiCoinCrowdsale` contract's address to the `kasei_crowdsale_address` variable.
39          kasei_crowdsale_address = address(crowdsale);
40
41          // Set the `KaseiCoinCrowdsale` contract as a minter
42          token.addMinter(kasei_crowdsale_address);
43
44          // Have the `KaseiCoinCrowdsaleDeployer` renounce its minter role.
45          token.renounceMinter();
46      }
47  }
48
49
```