

# FundMe Solidity Contract - Key Components

---

The FundMe contract is a basic funding and withdrawal contract where users can send funds to the contract and only the owner (deployer) can withdraw them. Below are the most critical parts of the contract:

## 1. State Variables:

- The owner is set as immutable, ensuring it is permanently set to the deployer.
- Funders are tracked via a mapping, and an array holds addresses of contributors.
- totalWithdrawn tracks the total ETH withdrawn from the contract.

## 2. Functions:

- **fund()**: This function allows users to send funds to the contract. A minimum of 10 gwei is required to prevent spam transactions. The funders' contributions are tracked and logged using events.
- **withdraw()**: Only callable by the owner, it withdraws the entire contract balance to the owner's address, resets all funders' balances, and logs the transaction.
- **withdrawToDistinctAddress()**: Allows the owner to withdraw a specified amount to another address, adjusting funders' balances proportionally.

## 3. Security:

- The onlyOwner modifier restricts sensitive operations (like withdrawal) to the owner.
- Fallback and **receive()** functions ensure the contract can receive ETH sent directly.

```
root@LAPTOP-3QURC2N7:~/BC# npx hardhat ignition deploy ./ignition/modules/FundMe.js
You are running Hardhat Ignition against an in-process instance of Hardhat Network.
This will execute the deployment, but the results will be lost.
You can use --network <network-name> to deploy to a different network.

Hardhat Ignition 🚀

Deploying [ FundMeDeployer ]

Batch #1
  Executed FundMeDeployer#FundMe

[ FundMeDeployer ] successfully deployed 🚀

Deployed Addresses

FundMeDeployer#FundMe - 0x5FbDB2315678afecb367f032d93F642f64180aa3
```

```

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

root@LAPTOP-3QURC2N7:~/BC# eth_chainId
hardhat_metadata
eth_accounts
hardhat_getAutomine
eth_chainId
eth_getBlockByNumber
eth_getTransactionCount (3)
eth_getBlockByNumber
eth_getTransactionCount
eth_getBlockByNumber
eth_chainId
eth_maxPriorityFeePerGas
eth_estimateGas
eth_call
Contract deployment: FundMe
Contract address: 0x5fbdb2315678afecb367f032d93f642f64180aa3
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b2266

eth_sendTransaction
Contract deployment: FundMe
Contract address: 0x5fbdb2315678afecb367f032d93f642f64180aa3
Transaction: 0x0afa8466a179ccec229d68db972684e20c5de85e07a710c4fd4
43aba557b0fc1
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfff9b2266
Value: 0 ETH
Gas used: 1117129 of 1117129
Block #1: 0x8f7a94d361efaa9db1768b9f5f59eed24727adaab63687bf4ea
7f616e45a57e4

eth_getTransactionByHash
eth_getBlockByNumber
eth_getTransactionReceipt
root@LAPTOP-3QURC2N7:~/BC#

```

```

root@LAPTOP-3QURC2N7:~/BC# npx hardhat ignition deploy ./ignition/modules/FundMe.js --network localhost
Hardhat Ignition

Deploying [ FundMeDeployer ]

Batch #1
Executed FundMeDeployer#FundMe

[ FundMeDeployer ] successfully deployed

Deployed Addresses

FundMeDeployer#FundMe - 0x5fbdb2315678afecb367f032d93f642f64180aa3
root@LAPTOP-3QURC2N7:~/BC#

```

```

root@LAPTOP-3QURC2N7:~/BC# npx hardhat test

FundMe
  Deployment
    ✓ Should set the right owner (627ms)
  Funding
    ✓ Should transfer correct amount of funds to contract
    ✓ Should fail if sent amount is less than 10 Gwei
  Withdrawing
    ✓ Should allow owner to withdraw full funds
    ✓ Should allow owner to withdraw funds to a distinct address
    ✓ Should fail if non-owner tries to withdraw

Lock
  Deployment
    ✓ Should set the right unlockTime
    ✓ Should set the right owner
    ✓ Should receive and store the funds to lock
    ✓ Should fail if the unlockTime is not in the future
  Withdrawals
    Validations
      ✓ Should revert with the right error if called too soon
      ✓ Should revert with the right error if called from another account
      ✓ Shouldn't fail if the unlockTime has arrived and the owner calls it
    Events
      ✓ Should emit an event on withdrawals
    Transfers
      ✓ Should transfer the funds to the owner

15 passing (766ms)

```

# FundMe Test File - Key Components

---

The test suite covers important functionalities of the FundMe contract. Key tests include:

## 1. Deployment Tests:

- The test ensures that the contract's owner is correctly set as the deployer of the contract.

## 2. Funding Functionality:

- Tests ensure users can successfully fund the contract if they send more than the required minimum.
- Also checks that contributions are accurately tracked in the contract's state.
- Another test ensures transactions fail if less than 10 gwei is sent, as required by the contract.

## 3. Withdrawal Tests:

- Tests that the owner can successfully withdraw the full balance of the contract, resetting funders' balances.
- Also ensures the owner can withdraw specific amounts to distinct addresses, with contributions deducted proportionally.
- Non-owner withdrawals are tested to ensure that only the owner can initiate a withdrawal, maintaining security.