

LESSON 6: HARDHAT SIMPLE STORAGE

Hardhat is the development environment for Ethereum software. It can deploy, debug, and compile smart contract and dApps.

By running `npx hardhat` you can see the available task :

```
^C
mumun@Mumun-MacBook-Air hardhat-simple-storage % npx hardhat
Hardhat version 2.9.9

Usage: hardhat [GLOBAL OPTIONS] <TASK> [TASK OPTIONS]

GLOBAL OPTIONS:

  --config          A Hardhat config file.
  --emoji           Use emoji in messages.
  --help           Shows this message, or a task's help if its name is provided
  --max-memory      The maximum amount of memory that Hardhat can use.
  --network         The network to connect to.
  --show-stack-traces Show stack traces.
  --tsconfig        A TypeScript config file.
  --verbose        Enables Hardhat verbose logging
  --version        Shows hardhat's version.

AVAILABLE TASKS:

  accounts          Prints the list of accounts
  block-number      print current block number
  check            Check whatever you need
  clean            Clears the cache and deletes all artifacts
  compile          Compiles the entire project, building all artifacts
  console          Opens a hardhat console
  flatten          Flattens and prints contracts and their dependencies
  gas-reporter:merge
  help            Prints this message
  node            Starts a JSON-RPC server on top of Hardhat Network
  run             Runs a user-defined script after compiling the project
  test            Runs mocha tests
  verify          Verifies contract on Etherscan

To get help for a specific task run: npx hardhat help [task]

mumun@Mumun-MacBook-Air hardhat-simple-storage %
```

You run it default task or you can define your own task like `block-number`, but to run your own task after creating it in the task folder you have to import it in the `hardhat.config.js` file

Deploy:

`npx hardhat run scripts/deploy.js`

```
mumun@Mumun-MacBook-Air hardhat-simple-storage % npx hardhat run scripts/deploy.js
Deploying contract .....
deployed to: 0x5FbDB2315678afecb367f032d93F642f64180aa3
{
  hardfork: 'arrowGlacier',
  blockGasLimit: 30000000,
  gasPrice: 'auto',
  chainId: 31337,
  throwOnTransactionFailures: true,
  throwOnCallFailures: true,
  allowUnlimitedContractSize: false,
  mining: { auto: true, interval: 0, mempool: { order: 'priority' } },
  accounts: {
    initialIndex: 0,
    count: 20,
    path: "m/44'/60'/0'/0",
    passphrase: '',
    mnemonic: 'test test test test test test test test test test junk',
    accountsBalance: '1000000000000000000000000'
  },
  loggingEnabled: false,
  gasMultiplier: 1,
  minGasPrice: <BN: 0>,
  chains: Map {
    1 => { hardforkHistory: [Map] },
    3 => { hardforkHistory: [Map] },
    4 => { hardforkHistory: [Map] },
    42 => { hardforkHistory: [Map] }
  },
  gas: 30000000,
  initialDate: '2022-07-08T15:56:23.907Z'
}
current value is: 0
update value is 7
mumun@Mumun-MacBook-Air hardhat-simple-storage %
```

Testing / Estimate gas

`npx hardhat test`

You can estimate how much gas things cost by running:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

simpleStorage
✓ Should start with a favorite number of 0
✓ Should update when we call store
```

Solc version: 0.8.8		Optimizer enabled: false		Runs: 200	Block limit: 30000000 gas	
Methods						
Contract	Method	Min	Max	Avg	# calls	eur (avg)
SimpleStorage	store	-	-	43724	2	-
Deployments						
SimpleStorage		-	-	463682	1.5 %	-

```
2 passing (1s)
mumun@Mumun-MacBook-Air hardhat-simple-storage %
```

And you'll see an output file called `gas-report.txt`

Local Deployment

If you'd like to run your own local hardhat network, you can run:

npx hardhat node

```
mumun@Mumun-MacBook-Air hardhat-simple-storage % npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
=====

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.

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cFfB92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90F79bf6EB2c4f870365E785982E1f101E93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507D1a55bcC2695C58ba16FB37d819B0A4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976EA74026E726554dB657fA54763abd0C3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dC79964da2C08b23698B3D3cc7Ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcd9bf7cbf4356

Account #8: 0x23618e81E3f5cdF7f54C3d65f7FBc0aBf5B21E8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22f6ea921c0cf5d620ea67b97

Account #9: 0xa0Ee7A142d267C1f36714E4a8F75612F20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6
```

And then in a different terminal

npx hardhat run scripts/deploy.js --network localhost

```

mumun@mumun-MacBook-Air hardhat-simple-storage % npx hardhat console --network localhost
Welcome to Node.js v12.13.0.
Type ".help" for more information.
> const simpleStorageFactory = await ethers.getContractFactory("SimpleStorage")
undefined
> const simpleStorage = await SimpleStorageFactory.deploy()
Thrown:
ReferenceError: SimpleStorageFactory is not defined
    at repl:1:16
> const simpleStorage = await simpleStorageFactory.deploy()
undefined
> await simpleStorage.retrieve()
BigNumber { value: "0" }
> await simpleStorage.store("50")
{
  hash: '0xef717cc211c9252f9b50739c9dbefadccd1fe313ca99a4c2b79627f2ba2225b5',
  type: 2,
  accessList: [],
  blockHash: '0xdd9f8350e73884a096fc33a7436dd496bbf4d611663260225dca95089786488',
  blockNumber: 2,
  transactionIndex: 0,
  confirmations: 1,
  from: '0xf39Fd6e51aad88F6F4ce6aB8827279cFfFb92266',
  gasPrice: BigNumber { value: "769006015" },
  maxPriorityFeePerGas: BigNumber { value: "0" },
  maxFeePerGas: BigNumber { value: "973273237" },
  gasLimit: BigNumber { value: "43724" },
  to: '0x5FbDB2315678afecb367f032d93F642f64180aa3',
  value: BigNumber { value: "0" },
  nonce: 1,
  data: '0x6057361d0000000000000000000000000000000000000000000000000000000000000032',
  r: '0x45890345fb77683b826785a586d36c491a4d9d2be9345362aafb5e4a43af28e4',
  s: '0x2bf81015f77d01b2ac1ea3620f9ffd8bf8a56647f2442756c4fd44a1e25ead98',
  v: 0,
  creates: null,
  chainId: 31337,
  wait: [Function]
}
> await simpleStorage.retrieve()
BigNumber { value: "50" }
>

```

In the hardhat console we can do basic function like in simpleStorage file. Like the example above we can do simpleStorage.retrieve() to see the value of BigNumber and then we can do simpleStorage.store("50") to store it in the simpleStorage and then retrieve it again to see the value. Hardhat can run on your localhost.