



To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

Published in [Coding the Smart Way](#)



Sebastian

[Follow](#)

Mar 28 · 5 min read · ✨ · 🎧 Listen



Save



Learn Full Stack Ethereum Development — Part 1

Setting Up The Project With React And Hardhat

This is the first part of the *Learn Full Stack Ethereum Development* series. This series will provide you with a beginner's tutorial to learn full stack Ethereum development from scratch.

In this series you'll learn how to setup Hardhat to run your local Ethereum blockchain and implement and deploy your first smart contract. You'll get to know how to use the





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

order to imp
framework.

of the React

This very first part of the tutorial series covers the first steps: setting up the general project structure, installing Hardhat and setting of the React front-end application. Let's get started.

Step 1: Creating A New React Application

First, we're creating a new React project by using the `create-react-app` script. To execute this script directly we're using the Node.js Package Runner `npx` like you can see in the following:

```
$ npx create-react-app my-react-dapp-01  
$ cd my-react-dapp-01
```

If the React app is created successfully you should be able to see the following output on the command line:





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

```
Success
pp-01
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-react-dapp-01
  npm start

Happy hacking!
(base) → Dapps
```

Creating a new React project

The default React project structure should then look like you can see in the following screenshot:





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

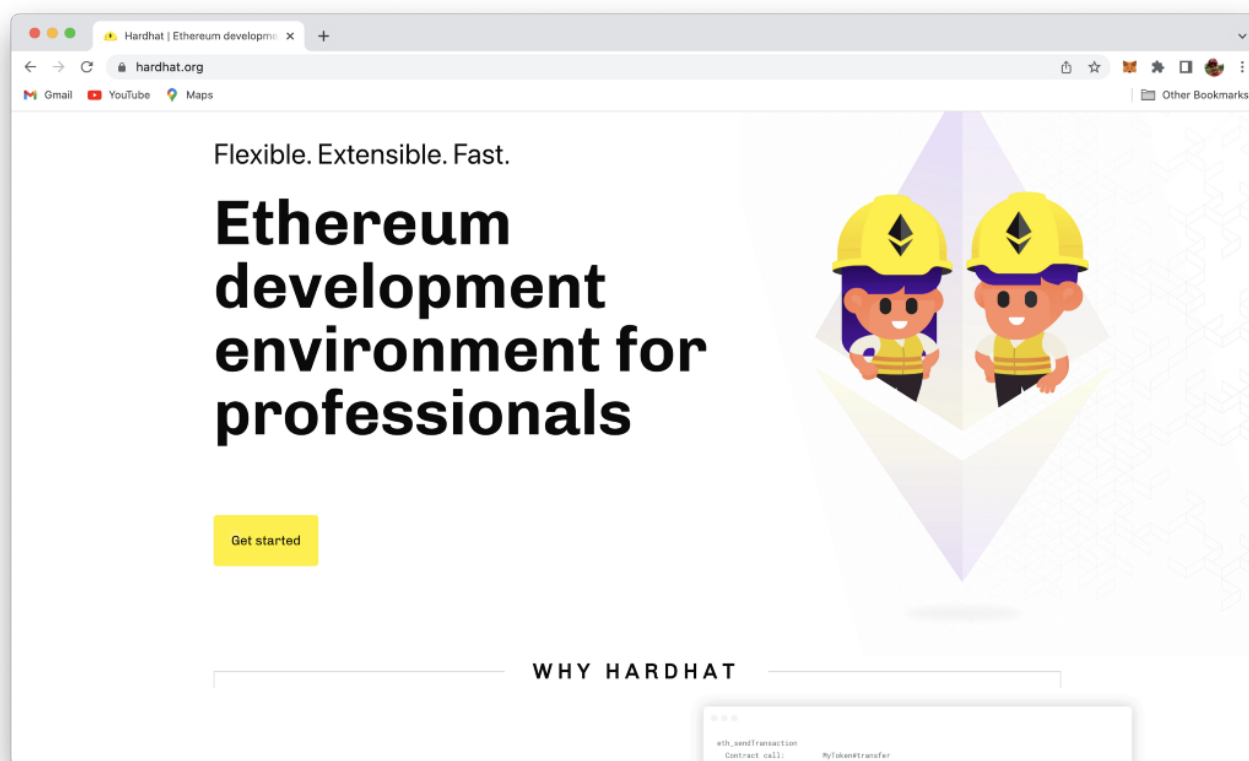
```
(base)
├── README.md
├── node_modules
├── package-lock.json
├── package.json
├── public
└── src

3 directories, 3 files
(base) → my-react-dapp-01 git:(master)
```

Default project structure of the React project

Step 2: Installing Hardhat

To setup our Ethereum development stack we need to add Hardhat next. Hardhat is a development environment for Ethereum which runs locally. The project's homepage can be found at <https://hardhat.org>:





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

```
$ npm install --save-dev hardhat
```

```
sebastian@Sebastians-MacBook-Pro:~/Projects/Dapps/my-react-dapp-01
(base) → my-react-dapp-01 git:(master) npm install --save-dev hardhat
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'hardhat@2.9.2',
npm WARN EBADENGINE   required: { node: '^12.0.0 || ^14.0.0 || ^16.0.0' },
npm WARN EBADENGINE   current: { node: 'v17.8.0', npm: '8.5.5' }
npm WARN EBADENGINE }

added 196 packages, and audited 1605 packages in 12s

195 packages are looking for funding
  run `npm fund` for details

6 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
(base) → my-react-dapp-01 git:(master) x
```

Hardhat is installed into the project

Once Hardhat is installed we need to make sure that a few more dependencies are added to the project.

Step 3: Installing Further Dependencies

Beside the *hardhat* package we need to install a few more dependencies by using the following *npm* command:

```
$ npm install --save-dev @nomiclabs/hardhat-waffle ethereum-waffle chai
@nomiclabs/hardhat-ethers ethers
```





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

ts.

- *ethereum*
- *hardhat-waffle*: This plugin adds a Hardhat-ready version of Waffle to the Hardhat Runtime Environment.
- *chai*: Chai is an *assertion library*, similar to Node's built-in `assert`. It makes testing much easier by giving you lots of assertions you can run against your code.
- *ethers*: The ethers.js library aims to be a complete and compact library for interacting with the Ethereum Blockchain and its ecosystem.
- *hardhat-ethers*: This plugin brings to Hardhat the Ethereum library *ethers*, which allows you to interact with the Ethereum blockchain in a simple way.

Step 4: Setting Up Hardhat

Now that all the needed packages have been added to the project we can start to setup Hardhat in our project by using the following command:

```
$ npx hardhat
```

Executing this command guides you through the process which is required to setup Hardhat. You need to answer a few questions on the command line to complete the process:





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

Get started

To learn
t.org/n

/hardha

```
888 888 888 888 888
888 888 888 888 888
888 888 888 888 888
8888888888 8888b. 888d888 .d88888 88888b. 8888b. 888888
888 888 "88b 888P" d88" 888 888 "88b "88b 888
888 888 .d888888 888 888 888 888 .d888888 888
888 888 888 888 888 Y88b 888 888 888 888 Y88b.
888 888 "Y888888 888 "Y88888 888 888 "Y888888 "Y888
```

👤 Welcome to Hardhat v2.9.2 👤

? What do you want to do? ...

> Create a basic sample project

Create an advanced sample project

Create an advanced sample project that uses TypeScript

Create an empty hardhat.config.js

Quit

Choose **Create a basic sample project** when setting up Hardhat for our project

Choose option *Create a basic sample project* when you're being asked the question: "What do you want to do?".

Next confirm the default project root location which is being proposed and choose to add a *.gitignore* file.





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

Get started

```
t.org/n
888
888 888      888 888      888
888 888      888 888      888
88888888888 8888b. 888d888 .d888888 88888b. 8888b. 8888888
888 888      "88b 888P" d88" 888 888 "88b      "88b 888
888 888 .d888888 888 888 888 888 .d888888 888
888 888 888 888 888 Y88b 888 888 888 888 Y88b.
888 888 "Y888888 888 "Y88888 888 888 "Y888888 "Y888

👤 Welcome to Hardhat v2.9.2 👤

✓ What do you want to do? · Create a basic sample project
✓ Hardhat project root: · /Users/sebastian/Projects/Dapps/my-react-dapp-01
✓ Do you want to add a .gitignore? (Y/n) · y

✨ Project created ✨
See the README.md file for some example tasks you can run.
(base) → my-react-dapp-01 git:(master) x
```

The Hardhat project configuration is finished successfully

When you take a look at the project structure again you'll notice that for Hardhat the *contracts* and *scripts* folder have been added and a Hardhat configuration file (*hardhat.config.js*) has been created in the project root folder as well:

```
sebastian@Sebastians-MacBook-Pro:~/Projects/Dapps/my-react-dapp-01
(base) → my-react-dapp-01 git:(master) x tree -L 1
.
├── README.md
├── contracts
├── hardhat.config.js
├── node_modules
├── package-lock.json
├── package.json
├── public
├── scripts
├── src
└── test

6 directories, 4 files
(base) → my-react-dapp-01 git:(master) x
```





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

Get started

Finally, we're

lockchain.

by using the command:

```
$ npx hardhat node
```

As you can see from the output the server starts and also creates a bunch of test accounts. In the list which is outputted you can see the account key together with its private key. Each of these test accounts already has a balance of 10000 ETH. Later in this tutorial we'll use some of these test accounts for interacting with the Hardhat local blockchain.



33



Sign up for CodingTheSmartWay Newsletter

By CodingTheSmartWay

Coding Tutorials Made Easy [Take a look.](#)

Your email



Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

Get started

```
(base)
Started

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: 0xf39fd6e51aad88f6f4ce6ab8827279cfff92266 (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: 0x4bbb85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcdbf7cbf4356

Account #8: 0x23618e81e3f5cdf7f54c3d65f7fbc0abf5b21e8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22feea921c0cf5d620ea67b97

Account #9: 0xa0ee7a142d267c1f36714e4a8f75612f20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6

Account #10: 0xbcd4042de499d14e55001ccbb24a551f3b954096 (10000 ETH)
Private Key: 0xf214f2b2cd398c806f84e317254e0f0b801d0643303237d97a22a48e01628897
```

Test accounts are being created when the Hardhat server is started

What's Next?

In the next part of this tutorial series we'll take a deeper look at the default Hardhat sample project to gain a better understanding of how a smart contract is implemented, how testing





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

[Get started](#)

Top Course

Ethereum and Solidity: The Complete Developer's Guide*

Use Ethereum, Solidity, and Smart Contracts to build production-ready apps based on the blockchain

[Go To Course ...*](#)

** Affiliate Link / Advertisement: This blog post contains affiliate links, which means that if you click on one of the product links and buy a product, we'll receive a small commission. There are no additional costs for you. Thank you for the support!*

