

Truffle and Next.js

Rapid Ethereum Dapp Development

A Minimal Smart Contract Development Boilerplate

[Truffle](#) is great for developing Solidity smart contracts, but building a React frontend for a smart contract is often a chore. [Next.js](#) is one of the easiest ways to build such a frontend and the integration between Truffle and Next.js is what this boilerplate is trying to demonstrate.

There are two major features:

- A plaintruffle init
- project is used as the base (along with a SimpleStorage example contract).
- A Next.js project resides in theclient
- directory with a symlink to the output folder of the contract ABI definitions. The Next.js app also provides a simple skeleton for connecting to and interacting with the smart contract on a network.

For more information on how the frontend works, go read the [README.md](#) located in theclient directory.

Installation

1. Install Truffle globally.
2. npm
3. install
4. -g
5. truffle
6. Download the box. This also takes care of installing the necessary dependencies.
7. truffle
8. unbox
9. adrianmcli/truffle-next
10. Run the development console.
11. truffle
12. develop
13. Compile and migrate the smart contracts. Note inside the development console we don't preface commands withtruffle
14. .
15. compile
16. migrate
17. Run the next.js server for the front-end. Smart contract changes must be manually recompiled and migrated.
18. //
19. Change
20. directory
21. to
22. the
23. front-end
24. foldercd
25. client
26. //
27. Serves
28. the
29. front-end
30. on
31. http://localhost:3000
32. npm
33. run
34. dev
35. Truffle can run tests written in Solidity or JavaScript against your smart contracts. Note the command varies slightly if you're in or outside of the development console.
36. //
37. If
38. inside
39. the
40. development
41. console.test
42. //
43. If

44. outside
45. the
46. development
47. console..
48. truffle
49. test

Running with MetaMask ¶

Since truffle develop exposes the blockchain onto port 9545, you'll need to add a Custom RPC network of `http://localhost:9545` in your MetaMask to make it work.

Running with TestRPC ¶

We highly recommend using `truffle develop` over `testrpc`, but if you want to use `testrpc`, there are a couple things you need to do:

- Change Line 6 of `client/lib/getWeb3.js`
- to use `localhost:8545`
- instead of `localhost:9545`
- so we refer to `testrpc`
- instead of `truffle develop`
- .
- Run your `testrpc`
- with the following command (because [reasons](#)
-):

```
testrpc --gasLimit 6721975 --gasPrice 100000000000
```