

Truffle and React (create-react-app)

Rapid Ethereum Dapp Development

A Minimal Smart Contract Development Boilerplate with create-react-app

-based frontend.

[Truffle](#) is great for developing Solidity smart contracts, and [create-react-app](#) is a great way to bootstrap a React project. Unfortunately, the official [truffle box for React](#) uses the eject mode of the create-react-app, which may be a disadvantage to many React developers. This box provides a basic integration between truffle and React app without using the eject mode of create-react-app.

There are two major features:

- A plain truffle init project is used as the base (along with a SimpleStorage example contract).
- A create-react-app based React project resides in the web-app directory with a symlink to the build/contracts
- folder containing ABI definitions (created after running truffle compile)
-). The provided React app is intentionally minimalistic to avoid imposing any specific requirements on the developer.

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

Installation

1. Install Truffle globally.
2. yarn
3. global
4. add
5. truffle
6. Download the box. This also takes care of installing the necessary dependencies.
7. truffle
8. unbox
9. Charterhouse/truffle-create-react-app
10. Run the development console.
11. truffle
12. develop
13. Compile and migrate the smart contracts. Note that inside the development console we don't preface commands with truffle
14. .
15. compile
16. migrate
17. 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.
18. **If inside the development console.**
19. test
20. **If outside the development console..**
21. truffle
22. test
23. Run the create-react-app server for the front-end. Smart contract changes must be manually recompiled and migrated.
24. **Change directory to the front-end folder**
25. cd
26. web-app# Serves the front-end on http://localhost:3000
27. yarn

- 28. start
- 29. We included some basic tests for our react components. You can run them from the web-app
- 30. folder:

31. **Change directory to the front-end folder**

- 32. cd
- 33. web-app
- 34. yarn
- 35. test

36. **for watch mode**

- 37. CI
- 38. =
- 39. TRUE
- 40. yarn
- 41. test

42. **for non-watch mode**

Visual Studio Code integration

The project is ready for Visual Studio Code. Out of the box it supports integration with [standardJS](#) and [vscode-jest](#) .

standardJS

The integration with standardJS is done on two levels: settings.json for the VSCode Workspace and the top-level package.json .

The workspace level options in settings.json are the following:

"javascript.validate.enable" :

false , "standard.usePackageJson" :

true , "standard.autoFixOnSave" :

true The top-level package.json includes the following standardJS configuration:

"standard" :

{

"parser" :

"babel-eslint" ,

"ignore" :

[

"build/**" ,

"node_modules/**" ,

"web-app/node_modules/**" ,

"web-app/src/contracts"

],

"envs" :

[

"es6" ,

```
"browser" ,
```

```
"jest"
```

```
],
```

```
"globals" :
```

```
[
```

```
"artifacts" ,
```

```
"contract" ,
```

```
"assert"
```

] } The only thing that still remains to be performed by the user is to install the JavaScript Standard Style extension (authored by Sam Chen).

vscode-jest

The vscode-jest extension (authored by orta and jest community) provides integration with jest test runner. Because the react project is in a subfolder, additional configuration has been added to the workspacesettings.json file:

```
"jest.pathToJest" :
```

```
"npm test --" , "jest.rootPath" :
```

```
"web-app" , "jest.restartJestOnSnapshotUpdate" :
```

true Note, that for the very same reason, Jest extension needs to be started manually via command palette (CMD+SHIFT+P and then Jest: Start Runner).

jest extension for VSCode only runs the tests for the web-app. You still need to run solidity tests using the truffle development console.