

Setting Up a React Project

This lesson will provide the guidelines to set up a react project.

In the Road to React, we'll use [create-react-app](#) to bootstrap your application. It's an opinionated yet zero-configuration starter kit for React introduced by Facebook in 2016, which is [recommended for beginners by 96% of React users](#). In *create-react-app*, the tools and configurations evolve in the background, while the focus remains on the application's implementation.

After installing Node and npm, use the command line to type the following command in a dedicated folder for your project. We'll refer to this project as *hacker-stories*, but you may choose any name you like:

```
npx create-react-app hacker-stories
```



Navigate into your new folder after the setup has finished:

```
cd hacker-stories
```



Now we can open the application in an editor or IDE. For Visual Studio Code, you can simply type `code .` on the command line. The following folder structure, or a variation of it depending on the *create-react-app* version, should be presented:

```
hacker-stories/  
--node_modules/  
--public/  
--src/  
----App.css  
----App.js  
----App.test.js  
----index.css  
----index.js  
logo.svg
```

```
---logo.svg
--.gitignore
--package-lock.json
--package.json
--README.md
```

This is a breakdown of the most important folders and files:

- **README.md:** The *.md* extension indicates the file is a markdown file. Markdown is a lightweight markup language with plain text formatting syntax. Many source code projects come with a *README.md* file that gives instructions and useful information about the project. When we push projects to platforms like GitHub, the *README.md* file usually displays information about the content contained in its repositories. Because you used create-react-app, your *README.md* should be the same as the official [create-react-app GitHub repository](#).
- **node_modules/:** This folder contains all node packages that have been installed via npm. Since we used create-react-app, a couple of node modules are already installed. We'll not touch this folder, since node packages are usually installed and uninstalled with npm via the command line.
- **package.json:** This file shows you a list of node package dependencies and other project configurations.
- **package-lock.json:** This file indicates npm to break down all node package versions. We'll not touch this file.
- **.gitignore:** This file displays all files and folders that shouldn't be added to your git repository when using git, as such files and folders should be located only in your local project. The *node_modules/* folder is one example. It is enough to share the *package.json* file with others, so they can install dependencies on their end with `npm install` without your entire dependency folder.
- **public/:** This folder holds development files, such as *public/index.html*. The index file is displayed on *localhost:3000* when the app is in development or on a domain that is hosted elsewhere. The default setup handles relating this *index.html* with all the JavaScript from *src/*.

In the beginning, everything you need is located in the *src/* folder. The main focus lies on the *src/App.js* file which is used to implement React components. It will be used to implement your application, but later you might want to split

up your components into multiple files, where each file maintains one or more components on its own.

Additionally, you will find a `src/App.test.js` file for your tests, and a `src/index.js` as an entry point to the React world. You will get to know both files intimately in later sections. There is also a `src/index.css` and a `src/App.css` file to style your general application and components, which comes with the default style when you open them. You will modify them later as well.

After you have learned about the folder and file structure of your React project, let's go through the available commands to get it started. All your project-specific commands can be found in your `package.json` under the `scripts` property. They may look similar to these:

```
{
  ...
},
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
  ...
}
```

These scripts are executed with the `npm run <script>` command in an IDE-integrated terminal or command line tool. The `run` can be omitted for the `start` and `test` scripts. The commands are as follows:

Another command from the previous npm scripts called `eject` shouldn't be used for this learning experience. It's a one way operation. Once you eject, you can't go back. Essentially this command is only there to make all the build tool and configuration from create-react-app accessible if you are not satisfied with the choices or if you want to change something. Here we will keep all the defaults though.

Runs the application in <http://localhost:3000>

```
npm start
```

Runs the tests

```
npm test
```

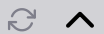
Builds the application for production

```
npm run build
```

You can test the above commands in the terminal below:

💡 A quick summary of the above commands.

Terminal



Note: This is just for you to test how to run your code locally using terminal. But with Educative, you can see the live execution of the code using our SPA widget.

Exercises:

- Read a bit more through React's [create-react-app documentation](#) and [getting started guide](#).
 - Read more about [the supported JavaScript features in create-react-app](#).
- Read more about [the folder structure in create-react-app](#).
 - Go through all of your React project's folders and files one by one.
- Read more about [the scripts in create-react-app](#).
 - Start your React application with `npm start` on the command line and check it out in the browser.
 - Exit the command on the command line by pressing `Control + C`.
 - Run the `npm test` script.
 - Run the `npm run build` script and verify that a `build/` folder was added to your project (you can remove it afterward). Note that the

added to your project (you can remove it afterward). Note that the build folder can be used later on to [deploy your application](#).

- Every time we change something in our code throughout the coming learning experience, make sure to check the output in your browser for getting visual feedback.

Let's learn about the react component in the next lesson.