

Setup on a Local Machine

Learn to run a react app on a local machine.

We'll cover the following ^

- Requirements
 - Editor and Terminal
 - Node and NPM
- Setup a react project

Requirements

The following are requirements for local setup:

Editor and Terminal

I have provided [a setup guide](#) to get you started with general web development. For this learning experience, you will need a text editor (e.g., Sublime Text) and a command-line tool (e.g., iTerm) or an IDE (e.g. Visual Studio Code). I recommend Visual Studio Code (also called VS Code) for beginners, as it's an all-in-one solution that provides an advanced editor with an integrated command-line tool, and because it's a popular choice among web developers.

Throughout this learning experience I will use the term *command-line*, which will be used synonymously for the terms *command-line tool*, *terminal*, and *integrated terminal*. The same applies to the terms *editor*, *text editor*, and *IDE*, depending on what you decided to use for your setup.

Optionally, I recommend managing projects in GitHub while we conduct the exercises in this course, and I've provided a [short guide](#) on how to use these tools. Github has excellent version control, so you can see what changes were made if you make a mistake or just want a more direct way to follow along. It's also a great way to share your code later with other people.

Node and NPM

Before we can begin, we'll need to have [node](#) and [npm](#) installed. Both are used to manage libraries (node packages) you will need along the way. These node packages can be libraries or whole frameworks. We'll install external node packages via npm (node package manager).

If you have already installed Node and npm, make sure that your installation is the most recent version. If you're new to npm or need a refresher, this [npm crash course](#) I created will get you up to speed.

 You can verify node and npm versions in the command line using the `node --version` command.

Setup a react project

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  
--.gitignore  
--package-lock.json
```

```
--package.json
--README.md
```

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:

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


```
npm start
```

Runs the tests

```
npm test
```

Builds the application for production

```
npm run build
```

 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.
