Creating a Library of React Components using Create React App

April 9th 2020 ★ 67,070 reads

11

Audio Presented by plivo

Speed: 1x

Read by: D

@aakashns
Aakash N S

UPDATE: You can use the npm package `create-component-lib` to automate all the steps described in this blog post.

create-component-lib

Create React App is easily the best tool for creating and developing React applications. With a little bit of work, it can also be used to create React components that can be published to `npm` and used in other React applications. Here's what you need to do:

1. Create a new project using `create-react-app`:

```
create-react-app simple-component-library
```

2. Delete all the files inside `src/` , and create a new `index.js` file with some starter code:

```
import React from "react";
import ReactDOM from "react-dom";

ReactDOM.render(<div>Hello world</div>, document.getElementById("root"));
```

3. Create a new folder `src/lib` and place your React components inside it. `src/lib` will serve as the root folder of the module published to
code for an example component:

```
import React from "react";
import "./TextInput.css";

const TextInput = ({ type = "text", label, value, onChange }) => (
  <div className="simple-form-group">
    {label && <label className="simple-text-label">{label}</label>}
    <input
      type={type}
      className="simple-text-input"
      value={value}
      onChange={e => onChange && onChange(e.target.value)}
    />
  </div>
);

export default TextInput;
```

Styling can be done inline, or in separate CSS files:

```
.simple-form-group {
  margin-bottom: 1rem;
}
.simple-text-label {
  display: block;
  color: red;
  color: red;
}
.simple-text-input {
  display: inline-block;
  margin-bottom: 0.5rem;
  font-size: 16px;
  font-weight: 400;
  color: rgb(33, 37, 41);
}
```

Finally, the component can be exported from `src/lib` for ease of importing:

```
import TextInput from "./TextInput";
export { TextInput };
```

Optionally, you can also write tests for the components inside `src/lib`:

```
import React from "react";
import TextInput from "./TextInput";
import renderer from "react-test-renderer";

describe("TextInput", () => {
  it("renders properly", () => {
    const tree = renderer
      .create(<TextInput label="Email" placeholder="name@example.com" />)
      .toJSON();
    expect(tree).toMatchSnapshot();
  });
});
```

4. *(Optional)* Use the components in `src/index.js` to create examples for testing and debugging during development. Any code placed outs not be published to `npm`. Here is an example usage of `TextInput`:

```
import React from 'react';
import { render } from "react-dom";
import { TextInput } from "./lib";

const App = () => (
  <div style={{ width: 640, margin: "15px auto" }}>
    <h1>Hello React</h1>

    <TextInput label="Email Address" placeholder="name@example.com" />
    </div>
  );

  render(<App />, document.getElementById("root"));
```

Run `npm start` and navigate to `http://localhost:3000` to view the result.

It works (duh!)

Tip: Use `react-live` to create a live-editable documentation site!

5. Install `babel-cli` using the command `npm i babel-cli --save-dev` and create a file `.babelrc` in the root of the project with the following con

```
    "presets": [["react-app", { "absoluteRuntime": false }]]
  }
```

6. Replace the `build` script inside `package.json` with the following:

```
  "build": "rm -rf dist && NODE_ENV=production babel src/lib --out-dir dist --copy-files --ignore __tests__,spec.js,test.js,__snapshots__
```

The command `npm run build` will now transpile the code inside `src/lib` (ignoring tests and snapshots) into the folder `dist`.

7. Remove the line `"private": true` from `package.json`. Also remove `react-scripts`, react and react-dom from dependencies, and move ther `devDependencies`. Additionally, you can also add `react` and `react-dom` to `peerDependencies`.

8. To prepare for publishing, add the following lines to `package.json`:

```
"main": "dist/index.js",
  "module": "dist/index.js",
  "files": [ "dist", "README.md" ],
  "repository": {
    "type": "git",
    "url": "URL_OF_YOUR_REPOSITORY"
  }
```

9. Remove the default `README.md` file, and create a new one with some information about the library.

```
# simple-component-library
A library of React components created using `create-react-app`.## Installation
Run the following command:
`npm install simple-component-library`
```

10. Publish to `npm`!

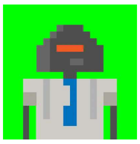
```
npm run publish
```

And that’s it! Now you can install the library with the command `npm install simple-component-library` and use it in any project created using (
App. Here’s the full code for this article:

aakashns/simple-component-library

You can clone the repository and use it as a starting point to skip some of the steps. Hope it helps!

11



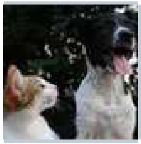
by Aakash N S @aakashns.

Read my stories

TAGS

#javascript #react #front-end-development #open-source #coding #nodejs #programming #beginners

RELATED STORIES



Training an Image Classifier From Scratch in 15 Minutes

Published at Apr 09, 2020 by aakashns #machine-learning

My Top 5 Best Practices for ReactJs

Published at Mar 14, 2022 by sachinchaurasiya #javascript

Cache API Calls in JavaScript by Memoizing Promises

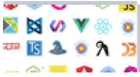
Published at Mar 14, 2022 by aldrinvincent #javascript

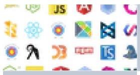
Javascript Cheatsheet: Spread Operators

Published at Mar 14, 2022 by imsabir #javascript

What is all the Fuss about Full Stack Developers?

Published at Mar 14, 2022 by max101 #developer





What it Takes to Create a Simple JavaScript Table in 2022

Published at Mar 14, 2022 by matkainer #technology

THIS ARTICLE WAS FEATURED IN...

Daveceddia Github Npmjs Dev Stackoverflow Cometchat Qiita Hatenablog

Join

HackerNoon

Curated Stories.|

THE HACKERNOON NEWSLETTER

Quality Weekly Reads About Technology Infiltrating Everything

name@company.com

Subscribe*free*

□ *Yes, I agree to receive emails about tech eating the world.*

ABOUT

Careers
Contact
Cookies
Emails
Help
Privacy
Terms

READ

Archive
Leaderboard
Noonification
Signup
Tech Brief
Tech Tags
Top Stories

WRITE

Distribution
Editor Tips
Guidelines
New Story
Perks
Prompts
Why Write

SPONSOR

Billboard
Brand Publishing
Case Studies
Contests
Niche Marketing
Newsletter
Writing Contests