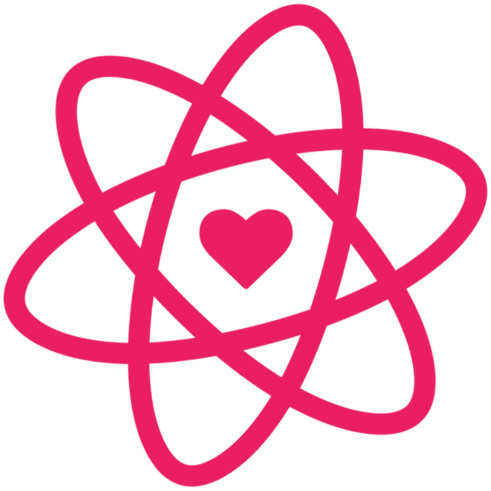
Noel Broda Follow
Solving problems, with people, words, and code. More about me: https://github.com/BrodaNoel/cv
Jun 7, 2017 · 3 min read

How to create a React component and publish it on NPM



Let's suppose this is an image about React and NPM

I was thinking about doing some kind of catharsis before starting this Story, but, let's go to the solutions.

If you want to create a React component and publish it in NPM, you have to do two things:

- Create the Component
- Create a project where you'll test your component

Creating the component

Let's suppose our component will be called `react-fancy-component`

```
cd ~
mkdir react-fancy-component
echo 'v6.10' > .nvmrc
echo '/node_modules' > .gitignore
nvm use
mkdir build
```

Then, manually, create these files:

`.babelrc`

```
{
  "presets": ["env"],
  "plugins": [
    "transform-object-rest-spread",
    "transform-react-jsx"
  ]
```

`webpack.config.js`

```
var path = require('path');
module.exports = {
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'build'),
    filename: 'index.js',
    libraryTarget: 'commonjs2' // THIS IS THE MOST IMPORTA
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        include: path.resolve(__dirname, 'src'),
        exclude: /(node_modules|bower_components|build)/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['env']
          }
```

`package.json` (do not forget to change the `name` in this file)

```
{
  "name": "react-fancy-component",
  "version": "0.0.1",
  "description": "Put a description here",
  "main": "build/index.js",
  "peerDependencies": {
    "react": "^15.5.4"
  },
  "dependencies": {
    "react": "^15.5.4",
    "webpack": "^2.6.1"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "webpack --watch",
    "build": "webpack"
  },
  "author": {
    "name": "Your name",
    "email": "your email"
```

Let’s continue running things:

```
npm i
npm run build
npm link
```

`npm link` will be used to dev-test our component in our testing-project while we are developing it.

Then, create your component in `/src/index.js`

Let’s try with this example. Create this file:

`src/index.js`

```
import React from 'react';
class Fancy extends React.Component {
  render() {
    return (
      <div>This is so Fancy!</div>
    );
  }
```

Done! Our fancy component is finished!

Creating our testing project

Now, let’s create a really basic project to test our fancy component.

```
cd ~
npm i -g create-react-app
mkdir react-fancy-project-test
cd react-fancy-project-test
nvm use 6.10
create-react-app .
```

Then, let’s modify our `src/App.js` file adding the import and the component usage. So, you should add these 2 lines:

`import Fancy from 'react–fancy–component';`

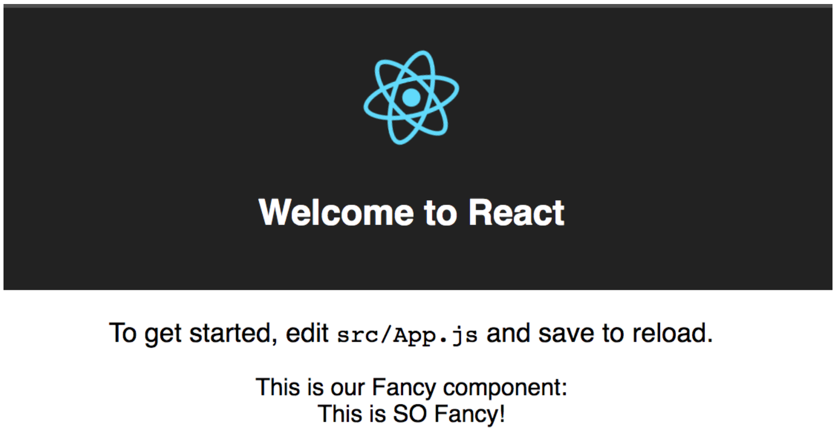
`<Fancy />`

Your `src/App.js` should look like this:

```
import React, { Component } from 'react';
import Fancy from 'react-fancy-component';
import logo from './logo.svg';
import './App.css';
class App extends Component {
  render() {
    return (
      <div className="App">
        <div className="App-header">
          <img src={logo} className="App-logo" alt="logo"
          <h2>Welcome to React</h2>
        </div>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and
```

And then, finally, run `npm start` and see what a beautiful fancy component you just created! You got it, man!

It should look like this



Welcome to React

To get started, edit `src/App.js` and save to reload.

This is our Fancy component:
This is SO Fancy!

Faaancy man

As last step, let’s publish our Component.

As first step, please, go to your component folder (`react-fancy-component`) and add all data that you should add in the `package.json` file, like name, version, description, and author.

Then, in the component folder, let's configure `npm` to publish your component (remember you need an account in http://npmjs.com/. Don't worry, it's free… for now…).

What you have to do, is run `npm login` and put your username, password and public email.

Then, run `npm publish` and… THAT'S ALL!

Check this out man: https://www.npmjs.com/package/react-fancy-component

Soo fancy!

Fancy maaaaan!

And the Magic? Where is the Magic?

The magic is in two lines in the whole project. Both lines are inside `webpack.config.json` file.

- `libraryTarget: 'commonjs2':` This line says "Hey, our Output should be an exportable module!". Check more info in the official documentation.
- `externals: { 'react': 'commonjs react' }` : This line says "Hey, let's use the React dependency that is using whoever is using me!".

Be careful with version-incompatibilities.

What about the development time?

You have to run `npm start` in two different Terminals. One terminal in the component folder, and another one in the test-project folder.

Webpack will recompile your component each time you do a change in any file (in your component), and then, automatically, the Webpack we have in our test-project will detect the change, and recompile our testing project. You will see your changes in your browser.