

## Introduction



38

[Get started](#)[Log in](#)

38

- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Changes to package.json](#)
- [Publishing the Component to npm](#)
- [Conclusion](#)
- [Top](#) ^



# Export React Components as Node Modules to NPM

Gaurav Singhal

Sep 12, 2020 • 5 Min read • 20,759 Views

Sep 12, 2020 • 5 Min read • 20,759 Views

Web Development

Front End Web Develo...

Client-side Frameworks

React

## Introduction

If you have been developing reusable components in React for your projects, an efficient approach for development is publishing them to npm as **node packages**. This is beneficial to you in two ways: First, you can use them in your future projects as dependencies without manually copying the code. And second, you can have other open-source developers look at the



We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).

[Disable cookies](#)[Accept cookies and close this message](#)

## Introduction



- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Ch](#)
- [Publishing the Component](#)
- [Conclusion](#)
- [Top](#) ^



In this guide, you'll learn how to create a simple React component using Create-React-App and export it as a **node module** to npm.

## Setting up

Let's build a simple horizontal rule component in React. For purposes of brevity, it will be a simple React component rendering an `hr` tag. However, you can do something more with it, such as add `props` giving more customizability to developers in terms of its usage and styles.

Create an empty React project using Create-React-App.

```
1 npx create-react-app react-hr-component
```

shell

Install Babel to transpile ES6 JavaScript.

```
1 npm install --save-dev @babel/cli @babel/pre
```

shell

Next, create a folder named `component` inside the root directory. Your custom component goes inside this folder. This is important because at some point you list out the path to your

component

We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).



Disable cookies

Accept cookies and close this message

## Introduction



38

- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Changes](#)
- [Publishing the Component](#)
- [Conclusion](#)
- [Top ^](#)



38

# Creating the Component

Inside the `component` folder, create a file named `HR.js` that contains the code for your component. Render a simple `hr` tag inside it.

```
1 import React from 'react';  
2  
3 export const HR=()=><hr/>
```

jsx

You can add styles directly to it as inline JSX styles or have a separate file for styling your JSX elements using classes. If you're using the latter method to style your component, make sure you import it correctly. You can also pass props for width, height, color, weight, etc. to this component and accordingly style the `hr` tag.

## Making Configurational Changes to package.json

Your component is good to go and ready to be published on npm. The next step is making some configurational changes to the `package.json` file that allows you to export this component as a node module on npm. In your `package.json` file under `scripts`, add the following line:

We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).



Disable cookies

Accept cookies and close this message

## Introduction



38

- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Ch](#)
- [Publishing the Component](#)
- [Conclusion](#)
- [Top](#) ^



38

The above line allows you to compile your JavaScript using Babel. Make a directory called `dist` and copy the compiled JavaScript to this file using a single command.

The next change you need to make is to set `private` to `false` to let anyone use your component as a normal dependency or node module. Also, add Babel `preset-react` inside the presets configuration for Babel to add React specific supports to your component, such as support for JSX.

json

```
1 {  
2   "name": "react-hr-component",  
3   "version": "0.1.0",  
4   "private": false,  
5   "babel": {  
6     "presets": [  
7       "@babel/preset-react"  
8     ]  
9   }  
10  ...  
11 }
```

The name appearing inside the `package.json` file is the actual name of the component that appears on npm. You can have a look at the entire `package.json` file here:

json



We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).

[Disable cookies](#)[Accept cookies and close this message](#)

## Introduction



38

- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Ch](#)
- [Publishing the Component](#)
- [Conclusion](#)
- [Top ^](#)



38

```
4  "private": false,
5  "babel": {
6    "presets": [
7      "@babel/preset-react"
8    ]
9  },
10 "dependencies": {
11   "@testing-library/jest-dom": "^4.2.4",
12   "@testing-library/react": "^9.5.0",
13   "@testing-library/user-event": "^7.2.1"
14   "react": "^16.13.1",
15   "react-dom": "^16.13.1",
16   "react-scripts": "3.4.3"
17 },
18 "scripts": {
19   "start": "react-scripts start",
20   "build": "react-scripts build",
21   "test": "react-scripts test",
22   "eject": "react-scripts eject",
23   "publish:npm": "rm -rf dist && mkdir di
24 },
25 "eslintConfig": {
26   "extends": "react-app"
27 },
28 "browserslist": {
29   "production": [
30     ">0.2%",
31     "not dead",
32     "not op_mini all"
33   ],
34   "development": [
35     "last 1 chrome version",
36     "last 1 firefox version",
37     "last 1 safari version"
38   ]
39 },
40 "devDependencies": {
41   "@babel/cli": "^7.11.5",
42   "@babel/preset-react": "^7.10.4"
43 }
44 }
```

We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).

[Disable cookies](#)[Accept cookies and close this message](#)

## Introduction



38

In case of any discrepancies, use the exact version of React being used in this example.



38

- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Changes](#)
- [Publishing the Component](#)
- [Conclusion](#)
- [Top](#) ^

## Publishing the Component to npm

You're just a few commands away from publishing your react component to npm. First, make sure you have an account on [npmjs](#).

Inside your terminal, run:

```
1 npm login
```

shell

The command line will prompt you to enter your npm credentials. After you have successfully logged in, run the following command:

```
1 npm run publish:npm
```

shell

The above line of code acts as the alias of the script you created for transpiling ES6 code using Babel and creating a build for your component. Finally, run the following command:

```
1 npm publish
```

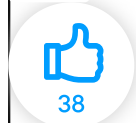
shell



We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).

[Disable cookies](#)[Accept cookies and close this message](#)

Introduction



- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Changes](#)
- [Publishing the Component](#)
- [Conclusion](#)
- [Top](#) ^

You can use it like any other React component available on npm.

# Conclusion

Publishing your React components to npm allows you to develop faster and also helps other developers use your components as node modules in their projects. After publishing your component on npm, you should also provide documentation for its installation and use along with a GitHub repository for it.

LEARN MORE



SOLUTIONS

[Pluralsight Skills \(/product/skills\)](#)

[Pluralsight Flow \(/product/flow\)](#)

PLATFORM

[Browse library \(/browse\)](#)

[Role IQ \(/product/role-iq\)](#)

We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).



[Disable cookies](#)

[Accept cookies and close this message](#)

services easy and meaningful. For more information about  
Introduction [View Pricing \(/pricing\)](/pricing) [Authors \(/authors\)](/authors)  
the cookies we use or to find out how you can disable

[cookies, click here](/product/contact-sales) [Contact Sales \(/product/contact-sales\)](/product/contact-sales)  
(<https://www.pluralsight.com/privacy.html>).  
[Skill up for free \(/product/skills/free\)](/product/skills/free)

[Professional Services \(/product/professional-services\)](/product/professional-services)

[Technology Index \(/tech-index\)](/tech-index)



38

ALLOW

DECLINE

- [Introduction](#)
- [Setting up](#)
- [Creating the Component](#)
- [Making Configurational Changes to package.json](#)
- [Publishing the Component to npm](#)
- [Conclusion](#)
- [Top ^](#)



We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, [click here](#).

[Disable cookies](#)[Accept cookies and close this message](#)