



Gaurav Singhal

How to Set up a React Bootstrap App

Gaurav Singhal

Jun 22, 2020 • 5 Min read • 2,337 Views

Jun 22, 2020 • 5 Min read • 2,337 Views

Web Development

Front End Web Dev...

Client-side Framew...

React

Introduction

5

- [Introduction](#)
- [Creating an Empty React Project](#)
- [Installing react-bootstrap](#)
- [Cleaning up the Starter Template](#)

Introduction

React-Bootstrap is a component-based library that provides native Bootstrap components as *pure React components*. Instead of utilizing JavaScript source and plugins from the CDN, it converts all JavaScript to React and bundles all components together. It's compatible with all kinds of Bootstrap themes to give a consistent appearance and maintain homogeneity in your app. In this guide,

- [Setting up a Development Server](#)
- [Importing a Global Stylesheet](#)
- [Using Bootstrap SASS](#)
- [Importing Components](#)
- [Customizing Components](#)
- [Conclusion](#)
- [Top ^](#)

you'll learn how to easily integrate React Bootstrap in any React app by setting up a React Bootstrap app in just a few minutes.

Creating an Empty React Project

Navigate to the directory where you want to store your project locally and create a new project using create-react-app by running the following command:

shell

```
1 npx create-react-app react-bootstrap-app
```

Installing react-bootstrap

To consume react-bootstrap modules, install two packages: react-bootstrap and Bootstrap, by running the following command:

shell

```
1 npm install react-bootstrap bootstrap
```

The first package will install all the react-bootstrap core components and place them inside your React's node modules

folder. From there, you can easily import any component you want to use.

The Bootstrap package contains all the Bootstrap styles to style your components and other things, such as *layouts*, *utilities*, *typography*, etc.

Cleaning up the Starter Template

Remove the logo, [App.css](#), and all their imports from [App.js](#).

Clean out the starter template inside the App component. Your [App.js](#) should look like this:

```
jsx
1  import React from 'react';
2  import './App.css';
3
4  function App() {
5    return(
6      <div className="App">
7        </div>
8      )
9    }
10
11  export default App;
```

Note: Sometimes removing App.css might land you an error. In such cases, instead of deleting the App.css, keep it empty, or if you have already deleted it, create a blank CSS file under the same name in the root directory.

Setting up a Development Server

Inside the root directory, run :

```
1     npm start
```

This will spin up a **local development server** (usually on port 3000) and you can see all your changes live in the browser.

Importing a Global Stylesheet

To use Bootstrap styles in your component, you must import them on the top.

```
1     import 'bootstrap/dist/css/bootstrap.min.css';
```

This lets you use all your regular Bootstrap classes for layout, utilities, typography, colors, etc. and is the CDN equivalent of adding the following line inside your index.html file :

html

```
1   <link
2     rel="stylesheet"
3     href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
4     integrity="sha384-9gS9baD1411NQApFmC26EwAOH8WgZ15MYxFfc+Nc
5     crossorigin="anonymous"
6   />
```

Reasons react-bootstrap doesn't include default bootstrap styles with their components:

- It makes the entire library lightweight without any pre-rendering issues
- It offers more customizability in terms of designing and styling react-bootstrap components

The best and most efficient way to include Bootstrap styles is by always using a CDN. This is because a CDN loads much faster than your app due to *caching* in browsers, so it avoids a lagging user experience. Use the first method only when you want to use Bootstrap styles in a few specific components only.

Using Bootstrap SASS

You can also include Bootstrap's source SASS files in your main SASS file and import it inside your component. Just include the following line inside a `.scss` file:

scss

```
1 @import "~bootstrap/scss/bootstrap";
```

and the following inside your component (ex App.js file) :

jsx

```
1 import './App.scss';
```

Importing Components

Let's look at a quick and simple example that shows how to import the desired components and render them on the DOM.

To use the **Button** component, simply import it from react-bootstrap in either of these ways :

jsx

```
1 import Button from 'react-bootstrap/Button';
2 // OR
```

```
3 import { Button } from 'react-bootstrap';
```

Then render this component inside [App.js](#):

jsx

```
1 ...
2 <Button>Click Me!</Button>
3 ...
```

Customizing Components

You can easily customize any Bootstrap theme that you want to use in your app or rewrite the Bootstrap classes and variables accordingly.

Create a custom SASS file and include the following block of code :

scss

```
1 //custom-style.scss
2 $theme-colors: (
3   "info": blueviolet,
4   "danger": orangered
5 );
```

Then import Bootstrap to set these changes :

scss

```
1 @import "~bootstrap/scss/bootstrap";
```

jsx

```
1 @import "custom-style";
```

Conclusion

Setting up a React Bootstrap app is fairly simple and doesn't require any complex installations. React Bootstrap's official docs are precise, leaving no room for doubt or confusion. Using a component-based UI library like React Bootstrap reduces your development time so you can focus more on implementing complex event-driven features in your app.