

## Basic Usage



17

Ashutosh Singh

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

17

# How to Use React Bootstrap with NPM

Ashutosh Singh

Aug 23, 2020 • 9 Min read • 2,712 Views

Aug 23, 2020 • 9 Min read • 2,712 Views

Web Development

Front End Web Dev...

Client-side Framew...

React

## Introduction

React is currently the most preferred JavaScript framework, according to the [StackOverflow 2020 Developer Survey](#). In this guide, we will discuss [React Bootstrap](#), a popular React UI library.

While there are so many React UI libraries to choose from, React Bootstrap remains a popular choice because of its similarity with traditional Bootstrap and the availability of several Bootstrap

themes. It is one of the fastest ways to build interfaces using React and Bootstrap.

## Basic Usage

17

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

## Installation

To install `react-bootstrap` as a dependency, run the following command in your React project root directory.

1        `npm install react-bootstrap bootstrap`

bash

Or if you prefer Yarn, use this command.

1        `yarn add react-bootstrap bootstrap`

bash

You might ask why `bootstrap` is installed in addition to `react-bootstrap`. The reason is that `react-bootstrap` doesn't depend on a specific version of Bootstrap and does not include any CSS on its own, but a stylesheet is required to use `react-bootstrap` components.

## Importing Bootstrap

To use `react-bootstrap` components, add the following code to `src/index.js` or `App.js`.

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

17 1

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

JSX

And that's it; you can now use React Bootstrap and its components in your app.

## Basic Usage

Below are the five most commonly used React Bootstrap components.

- Buttons
- Forms
- Navbar
- Toasts
- Modals

## Importing Components

There are two ways to import components from React Bootstrap.

1

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

JSX

Or by pulling specific components from the library.

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

17

```
import Button from 'react-bootstrap/Button';
```

JSX

This import can significantly reduce the amount of code you end up sending to the client. This may be best for individual components, but not if several components are imported.

### Using Components

Here is how you can use any React Bootstrap component in your React app.

1. Import the component that you want to use, for example, `Button`.

JSX

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

2. Use the imported component with the props that can be found in React Bootstrap docs. For example, `Button` has props like `variant`, `type`, `target`, `size`, etc.

JSX

```
1 // src/App.js
2 import React from "react";
3 import Button from "react-bootstrap/Button";
4
```

## Basic Usage

17

```
5     function App() {
6       return (
7         <div>
8           <Button variant="primary">Click Me!</Button>
9         </div>
10      );
11    }
12
13   export default App;
```

Here is how the button will look.



Click Me!

Even if `variant="primary"` is removed, the button will still look the same since `primary` is the default value of `variant` prop.

You can experiment with different props and create new combinations.

- Here is a button with `variant="danger"` and `disabled`.

JSX

```
1   <Button variant="danger" disabled>Click Me!</Button>
```

Click Me!

## Basic Usage

17

Here is a button with `variant="outline-success"` and `size="lg"`.

JSX

```
1 <Button variant="outline-success" size="lg">Click Me!</Button>
```

Click Me!

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

## Example

This example shows the step by step process of styling an app with React Bootstrap.

It will use the character endpoint of the [Rick and Morty API](#), a free REST/GraphQL based API that provides information about characters, episodes, and locations of the Rick and Morty TV show.

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

1. First, request data from the `/character` endpoint. One way to do this is to use `fetch()` inside `useEffect()` hook and store the response data inside the state variable. By providing an empty array as a second argument to `useEffect()`, you can ensure that the request is made only after the initial render.

JSX

```
1  const [data, setData] = useState([]);  
2  
3  useEffect(() => {  
4      fetch("https://rickandmortyapi.com/api/character/")  
5          .then((res) => res.json())  
6          .then((data) => setData(data.results));  
7    }, []);
```

Next, you need to decide which data we want to show in your app; this example will show `name`, `image`, `status`, and `url`. You can check out the Character Schema [here](#).

2. You can show the character data in React Bootstrap's `Card` component. First, make empty cards based on the number of characters in the response using `.map()` method on the `data` state.

JSX

```
1  // src/App.js  
2  import React, { useEffect, useState } from "react";  
3  
4  function App() {  
5      const [data, setData] = useState([]);
```

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

```
6
7     useEffect(() => {
8         fetch("https://rickandmortyapi.com/api/character/")
9             .then((res) => res.json())
10            .then((data) => setData(data.results));
11        }, []);
12
13        return (
14            <div>
15                {data.map((character) => (
16                    <Card key={character.id} style={{ width: "18rem" }}>
17                        </Card>
18                    )));
19                </div>
20            );
21        }
22
23    export default App;
```

Here, `id` of the character is used as the value of the `key` prop. You will not see anything in the app, since the `<Card>` component is itself empty.

3. You can use `Card.Img` to display the image inside the `<Card>` component.

JSX

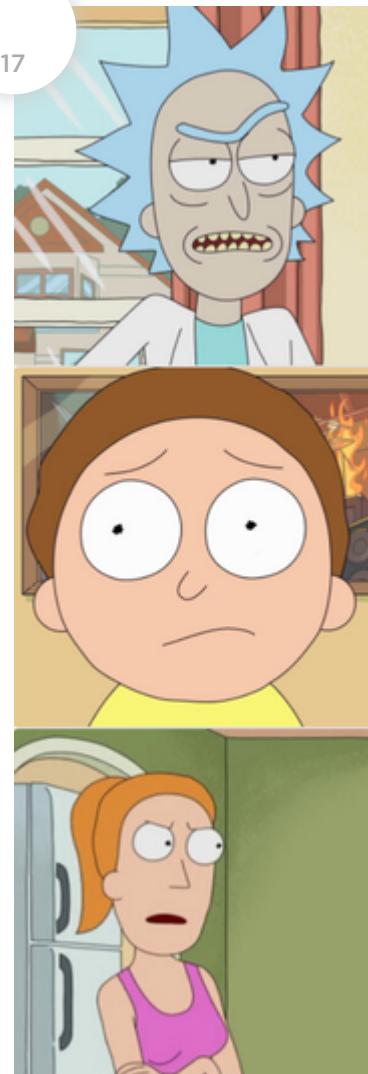
```
1     <Card key={character.id} style={{ width: "18rem" }}>
2         <Card.Img variant="top" src={character.image} />
3     </Card>
```

Now your application will show different characters' images in a single line without a margin.

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)



You can add `className="m-4"` to add some margin between cards.

JSX

```
<Card className="m-4" key={character.id} style={{ width: "18rem" }}>  
  <Card.Img variant="top" src={character.image} />  
</Card>
```

## Basic Usage

17  
2  
3

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

4. To show `name` and `species`, use `Card.Title` and `Card.Text` inside `Card.Body`.

JSX

```
1  <Card className="m-4" key={character.id} style={{ width: "18rem" }}>  
2    <Card.Img variant="top" src={character.image} />  
3    <Card.Body>  
4      <Card.Title>{character.name}</Card.Title>  
5      <Card.Text>{character.species}</Card.Text>  
6    </Card.Body>  
7  </Card>;
```

Here is how your card will look.

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)



17

Rick Sanchez

Human

5. Finally, add a `Button` to show more information about a particular character. You will also need to import the `Button` component from the `react-bootstrap` library.

JSX

```
<Card.Body>
  <Card.Title>{character.name}</Card.Title>
  1  <Card.Text>{character.species}</Card.Text>
  2  <Button variant="primary" href={character.url} target="_blank">
  3    More Info
  4  </Button>
  5  </Card.Body>
  6
  7
```

Here is how your app will look.

## Basic Usage

17

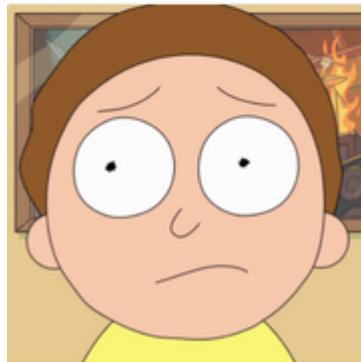
- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)



Rick Sanchez

Human

[More Info](#)



Morty Smith

Human

[More Info](#)

6. All the cards in a single line don't look right. To change the layout of the `<Card>` component, render everything inside `<CardColumns>` instead of `<div>`, which will divide the cards into columns.

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

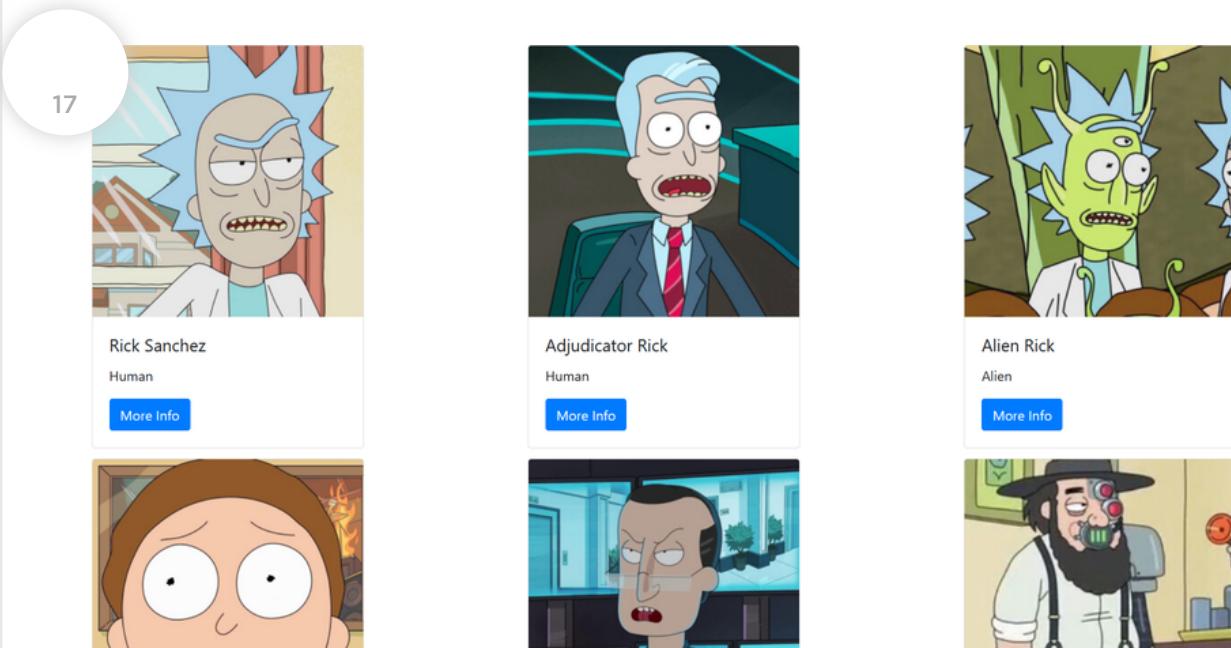
```
1 import React, { useEffect, useState } from "react";
2   import { Card, CardColumns, Button } from "react-bootstrap";
3
4
5 function App() {
6   const [data, setData] = useState([]);
7
8   useEffect(() => {
9     fetch("https://rickandmortyapi.com/api/character/")
10       .then((res) => res.json())
11       .then((data) => setData(data.results));
12   }, []);
13
14   return (
15     <CardColumns>
16       {data.map((character) => (
17         <Card className="m-4" key={character.id} style={{ width: "20rem" }}
18           <Card.Img variant="top" src={character.image} />
19
20           <Card.Body>
21             <Card.Title>{character.name}</Card.Title>
22             <Card.Text>{character.species}</Card.Text>
23             <Button variant="primary" href={character.url} target="_blank"
24               More Info
25             </Button>
26           </Card.Body>
27         </Card>
28       ))}
29     </CardColumns>
30   );
31 }
32
33 export default App;
```

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

Here is how your app will look.



You can use the example used here to experiment with different React Bootstrap components.

Here are some ideas to experiment with:

- Use [Modals](#) to show more information about a character when **More Info** button is clicked.
- Add a [Navbar](#) to the app.

## Conclusion

## Basic Usage

17

- [Introduction](#)
- [Installation](#)
- [Basic Usage](#)
- [Example](#)
- [Conclusion](#)
- [Top ^](#)

In this guide, we discussed how to install and use React Bootstrap in any React app, and we saw an example where styling became more straightforward and faster with React Bootstrap. Not every component was discussed here, so you should try to explore different components available in React Bootstrap.

17 Here are some additional resources that can be helpful.

- [React Bootstrap Docs](#)
- [Bootstrap Docs](#)

Happy coding!