|  |  |
| --- | --- |
| **ch28.ReactJS Bootstrap** | **Date: 22-02-2022** |

**Topics**

React Props Validation,

# React Props Validation

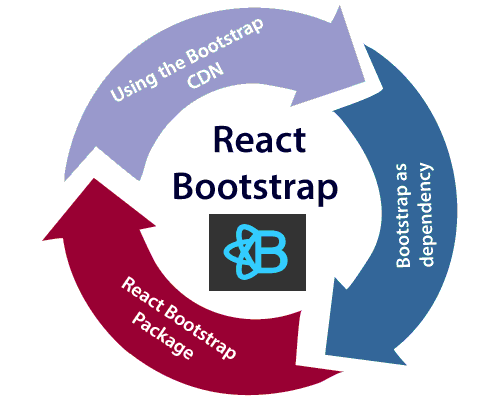
# React Bootstrap

Single-page applications gaining popularity over the last few years, so many front-end frameworks have introduced such as Angular, React, Vue.js, Ember, etc. As a result, jQuery is not a necessary requirement for building web apps. Today, React has the most used JavaScript framework for building web applications, and Bootstrap become the most popular CSS framework. So, it is necessary to learn various ways in which Bootstrap can be used in React apps, which is the main aim of this section.

## **Adding Bootstrap for React**

We can add Bootstrap to the React app in several ways. The **three** most common ways are given below:

1. Using the Bootstrap CDN
2. Bootstrap as Dependency
3. React Bootstrap Package



## **Using the Bootstrap CDN**

It is the easiest way of adding Bootstrap to the React app. There is no need to install or download Bootstrap. We can simply put an **<link>** into the **<head>** section of the **index.html** file of the React app as shown in the following snippet.

1. <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

If there is a need to use Bootstrap components which depend on JavaScript/jQuery in the React application, we need to include **jQuery**, **Popper.js**, and **Bootstrap.js** in the document. Add the following imports in the **<script>** tags near the end of the closing **</body>** tag of the **index.html** file.

Competitive questions on Structures in Hindi

Keep Watching

1. <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
3. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
5. <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

In the above snippet, we have used jQuery's slim version, although we can also use the full version as well. Now, Bootstrap is successfully added in the React application, and we can use all the CSS utilities and UI components available from Bootstrap in the React application.

## **Bootstrap as Dependency**

If we are using a build tool or a module bundler such as Webpack, then importing Bootstrap as dependency is the preferred option for adding Bootstrap to the React application. We can install Bootstrap as a dependency for the React app. To install the Bootstrap, run the following commands in the terminal window.

1. $ npm install bootstrap --save

Once Bootstrap is installed, we can import it in the React application entry file. If the React project created using the **create-react-app** tool, open the **src/index.js** file, and add the following code:

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

Now, we can use the CSS classes and utilities in the React application. Also, if we want to use the JavaScript components, we need to install the **jquery** and **popper.js** packages from **npm**. To install the following packages, run the following command in the terminal window.

1. $ npm install jquery popper.js

Next, go to the **src/index.js** file and add the following imports.

1. **import** $ from 'jquery';
2. **import** Popper from 'popper.js';
3. **import** 'bootstrap/dist/js/bootstrap.bundle.min';

Now, we can use Bootstrap JavaScript Components in the React application.

## **React Bootstrap Package**

The React Bootstrap package is the most popular way to add bootstrap in the React application. There are many Bootstrap packages built by the community, which aim to rebuild Bootstrap components as React components. The **two** most popular Bootstrap packages are:

1. **react-bootstrap:** It is a complete re-implementation of the Bootstrap components as React components. It does not need any dependencies like bootstrap.js or jQuery. If the React setup and React-Bootstrap installed, we have everything which we need.
2. **reactstrap:** It is a library which contains React Bootstrap 4 components that favor composition and control. It does not depend on jQuery or Bootstrap JavaScript. However, react-popper is needed for advanced positioning of content such as Tooltips, Popovers, and auto-flipping Dropdowns.

## **React Bootstrap Installation**

Let us create a new React app using the **create-react-app** command as follows.

1. $ npx create-react-app react-bootstrap-app

After creating the React app, the best way to install Bootstrap is via the npm package. To install Bootstrap, navigate to the React app folder, and run the following command.

1. $ npm install react-bootstrap bootstrap --save

### **Importing Bootstrap**

Now, open the **src/index.js** file and add the following code to import the Bootstrap file.

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

We can also import individual components **like import { SplitButton, Dropdown } from 'react-bootstrap';** instead of the entire library. It provides the specific components which we need to use, and can significantly reduce the amount of code.

In the React app, create a new file named **ThemeSwitcher.js** in the **src** directory and put the following code.

1. **import** React, { Component } from 'react';
2. **import** { SplitButton, Dropdown } from 'react-bootstrap';
4. **class** ThemeSwitcher **extends** Component {
6. state = { theme: **null** }
8. chooseTheme = (theme, evt) => {
9. evt.preventDefault();
10. **if** (theme.toLowerCase() === 'reset') { theme = **null** }
11. **this**.setState({ theme });
12. }
14. render() {
15. **const** { theme } = **this**.state;
16. **const** themeClass = theme ? theme.toLowerCase() : 'default';
18. **const** parentContainerStyles = {
19. position: 'absolute',
20. height: '100%',
21. width: '100%',
22. display: 'table'
23. };
25. **const** subContainerStyles = {
26. position: 'relative',
27. height: '100%',
28. width: '100%',
29. display: 'table-cell',
30. };
32. **return** (
33. <div style={parentContainerStyles}>
34. <div style={subContainerStyles}>
36. <span className={`h1 center-block text-center text-${theme ? themeClass : 'muted'}`} style={{ marginBottom: 25 }}>{theme || 'Default'}</span>
38. <div className="center-block text-center">
39. <SplitButton bsSize="large" bsStyle={themeClass} title={`${theme || 'Default Block'} Theme`}>
40. <Dropdown.Item eventKey="Primary Block" onSelect={**this**.chooseTheme}>Primary Theme</Dropdown.Item>
41. <Dropdown.Item eventKey="Danger Block" onSelect={**this**.chooseTheme}>Danger Theme</Dropdown.Item>
42. <Dropdown.Item eventKey="Success Block" onSelect={**this**.chooseTheme}>Success Theme</Dropdown.Item>
43. <Dropdown.Item divider />
44. <Dropdown.Item eventKey="Reset Block" onSelect={**this**.chooseTheme}>Default Theme</Dropdown.Item>
45. </SplitButton>
46. </div>
47. </div>
48. </div>
49. );
50. }
51. }
52. export **default** ThemeSwitcher;

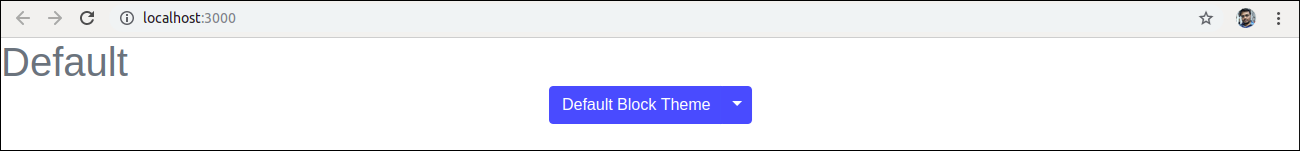
Now, update the **src/index.js** file with the following snippet.

**Index.js**

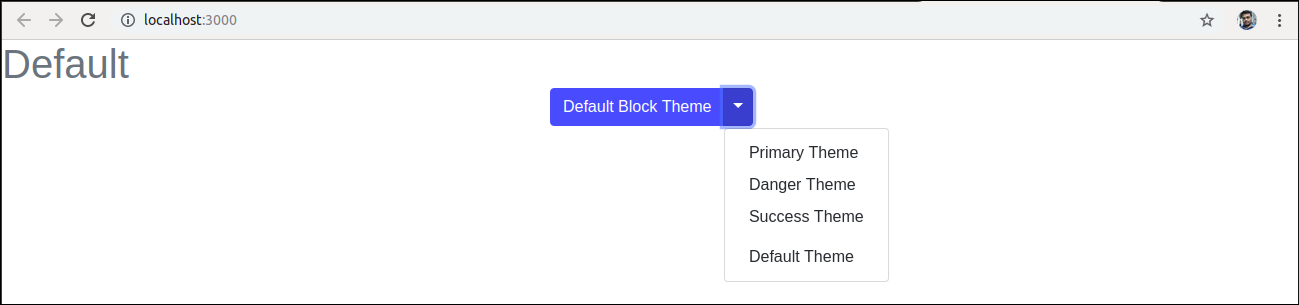
1. **import** 'bootstrap/dist/css/bootstrap.min.css';
2. **import** React from 'react';
3. **import** ReactDOM from 'react-dom';
4. **import** App from './App.js';
5. **import** './index.css';
6. **import** ThemeSwitcher from './ThemeSwitcher';
8. ReactDOM.render(<ThemeSwitcher />, document.getElementById('root'));

**Output**

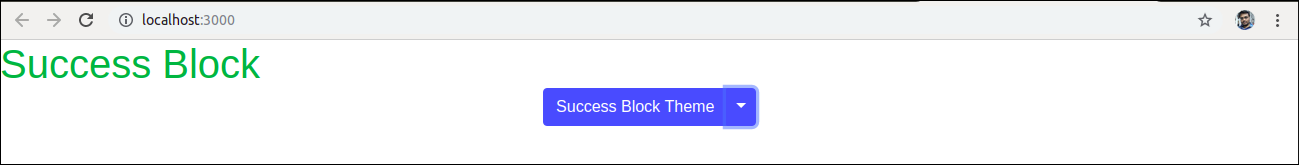
When we execute the React app, we should get the output as below.



Click on the dropdown menu. We will get the following screen.



Now, if we choose the **Success Theme**, we will get the below screen.



## **Using reactstrap**

Let us create a new React app using the create-react-app command as follows.

1. $ npx create-react-app reactstrap-app

Next, install the **reactstrap** via the npm package. To install reactstrap, navigate to the React app folder, and run the following command.

1. $ npm install bootstrap reactstrap --save

### **Importing Bootstrap**

Now, open the **src/index.js** file and add the following code to import the Bootstrap file.

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

We can also import individual components **like import { Button, Dropdown } from 'reactstrap';** instead of the entire library. It provides the specific components which we need to use, and can significantly reduce the amount of code.

In the React app, create a new file named **ThemeSwitcher.js** in the **src** directory and put the following code.

1. **import** React, { Component } from 'react';
2. **import** { Button, ButtonDropdown, DropdownToggle, DropdownMenu, DropdownItem } from 'reactstrap';
4. **class** ThemeSwitcher **extends** Component {
6. state = { theme: **null**, dropdownOpen: **false** }
8. toggleDropdown = () => {
9. **this**.setState({ dropdownOpen: !**this**.state.dropdownOpen });
10. }
12. resetTheme = evt => {
13. evt.preventDefault();
14. **this**.setState({ theme: **null** });
15. }
17. chooseTheme = (theme, evt) => {
18. evt.preventDefault();
19. **this**.setState({ theme });
20. }
21. render() {
22. **const** { theme, dropdownOpen } = **this**.state;
23. **const** themeClass = theme ? theme.toLowerCase() : 'secondary';
25. **return** (
26. <div className="d-flex flex-wrap justify-content-center align-items-center">
28. <span className={`h1 mb-4 w-100 text-center text-${themeClass}`}>{theme || 'Default'}</span>
30. <ButtonDropdown isOpen={dropdownOpen} toggle={**this**.toggleDropdown}>
31. <Button id="caret" color={themeClass}>{theme || 'Custom'} Theme</Button>
32. <DropdownToggle caret size="lg" color={themeClass} />
33. <DropdownMenu>
34. <DropdownItem onClick={e => **this**.chooseTheme('Primary', e)}>Primary Theme</DropdownItem>
35. <DropdownItem onClick={e => **this**.chooseTheme('Danger', e)}>Danger Theme</DropdownItem>
36. <DropdownItem onClick={e => **this**.chooseTheme('Success', e)}>Success Theme</DropdownItem>
37. <DropdownItem divider />
38. <DropdownItem onClick={**this**.resetTheme}>Default Theme</DropdownItem>
39. </DropdownMenu>
40. </ButtonDropdown>
42. </div>
43. );
44. }
45. }
46. export **default** ThemeSwitcher;

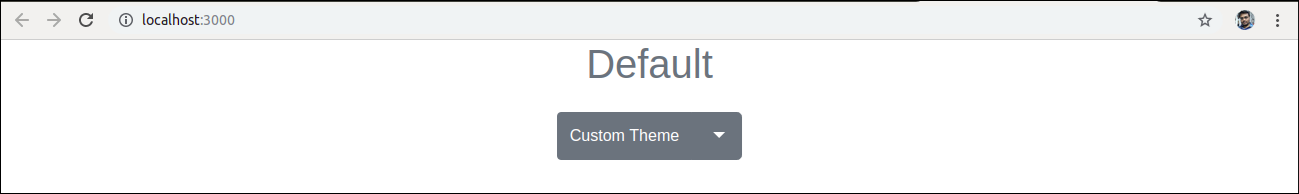
Now, update the **src/index.js** file with the following snippet.

**Index.js**

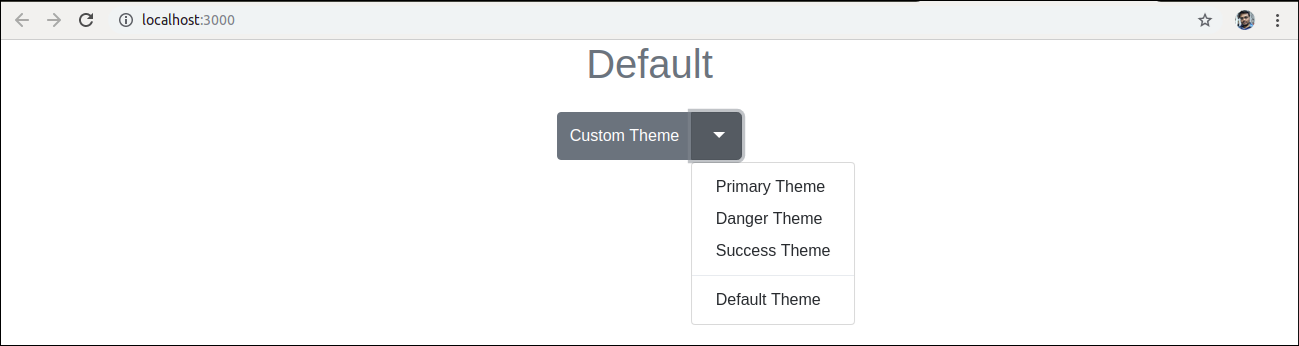
1. **import** 'bootstrap/dist/css/bootstrap.min.css';
2. **import** React from 'react';
3. **import** ReactDOM from 'react-dom';
4. **import** App from './App.js';
5. **import** './index.css';
6. **import** ThemeSwitcher from './ThemeSwitcher';
8. ReactDOM.render(<ThemeSwitcher />, document.getElementById('root'));

**Output**

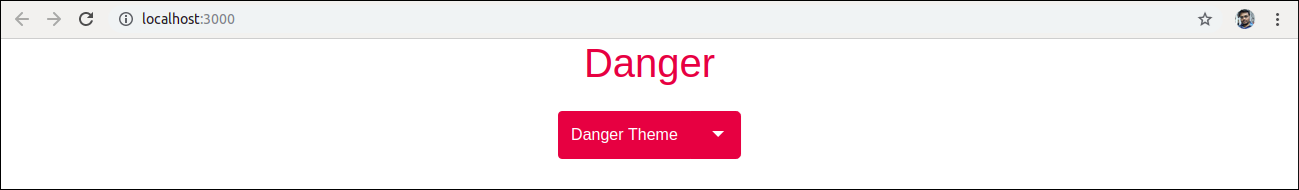
When we execute the React app, we should get the output as below.



Click on the dropdown menu. We will get the following screen.



Now, if we choose the **Danger Theme**, we will get the below screen.



Next Topic[React Map](https://www.javatpoint.com/react-map)