



Gaurav Singhal

# How to Use React Bootstrap with Redux

Gaurav Singhal

Sep 15, 2020 • 15 Min read • 5,184 Views

Sep 15, 2020 • 15 Min read • 5,184 Views

Web Development

React

## Introduction

28

- [Introduction](#)
- [React-Bootstrap](#)
- [Example](#)
- [LoginForm](#)
- [Component](#)
- [Building the Store](#)

## Introduction

Bootstrap is the most popular CSS framework, used by over a million websites on the internet. So it's imperative to know how we can integrate this excellent CSS framework with our favorite JavaScript front-end stack—React and Redux.

One way to add Bootstrap into your app is by using CDN links. But, fortunately, there's also already an npm library out there that solves

- [Wrapping Root Component with Provider](#)
- [Connecting <LoginForm /> Component with Redux](#)
- [Complete Source Code](#)
- [Conclusion](#)
- [Top ^](#)

our problem.

## React-Bootstrap

React-Bootstrap is the most popular front-end framework. It's rebuilt for React.js—that is, it completely replaces Bootstrap Javascript. React-Bootstrap is an npm package that completely re-implements Bootstrap components for React and has no dependency on [JQuery](#). It uses the state as supposed to direct DOM manipulation, which makes this package more reliable in the React world.

### Installing React-Bootstrap

To install [react-bootstrap](#), run the following command in your terminal.

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

console

Note: Please make sure you have node installed in your system before running the command.

## Example

For this guide, we will use a `<LoginForm />` component from the [React Bootstrap documentation](#) and validate the fields using Redux.

# Login

---

Email address

✓

We'll never share your email with anyone else.

Password

✓

Check me out

Submit

# Login

Email address



Email is invalid

We'll never share your email with anyone else.

Password



Password must have 8 characters

Check me out

Submit

## LoginForm Component

We will import the `<Form />` and `<Button />` components from `react-bootstrap` and use them in our `<LoginForm />` component.

jsx

```
1  import Form from "react-bootstrap/Form";
2  import Button from "react-bootstrap/Button";
3
4  const LoginForm = () => (
5      <Form>
```

```
6      <h2>Login</h2>
7      <hr />
8      <Form.Group controlId="formBasicEmail">
9          <Form.Label>Email address</Form.Label>
10         <Form.Control type="email" placeholder="Enter email" />
11         <Form.Text className="text-muted">
12             We'll never share your email with anyone else.
13         </Form.Text>
14     </Form.Group>
15
16     <Form.Group controlId="formBasicPassword">
17         <Form.Label>Password</Form.Label>
18         <Form.Control type="password" placeholder="Password" />
19     </Form.Group>
20     <Form.Group controlId="formBasicCheckbox">
21         <Form.Check type="checkbox" label="Check me out" />
22     </Form.Group>
23     <Button variant="primary" type="button">
24         Submit
25     </Button>
26     </Form>
27 );
28
29 export default LoginForm;
```

## Building the Store

In the global store object, we will store values and errors of the form.

```
1  const initialState = {
2      loginForm: {
3          values: {
4              email: "",
5              password: ""
6          },
7          errors: {
8              email: "",
9              password: ""
10         }
11     };
12
13     export default (state = initialState, action) => {
14         if (action.type === "FORM_SUBMIT") {
15             const { email, password } = action.payload;
16             const values = {
17                 email,
18                 password
19             };
20             const errors = {} // validate fields
21             return {
22                 loginForm: {
23                     values,
24                     errors
25                 }
26             };
27         }
28         return state;
29     };
30 }
```

To validate the fields, we will write a `validateEmail()` function to check whether the email entered by the user is valid or not and whether the password has at least 8 characters.

js

```
1  function validateEmail(email) {
2      var re = /^[^<>()\\[\\]\\.,;:\\s@"]+(\.[^<>()\\[\\]\\.,;:\\s@"]+)*|(.+")
3      return re.test(String(email).toLowerCase());
4  }
5
6  const setErrors = (email, password) => {
7      let errors = { email: "", password: "" };
8      if (!email && email.length === 0) {
9          errors.email = "Email is required";
10     } else if (!validateEmail(email)) {
11         errors.email = "Email is invalid";
12     }
13     if (!password && password.length === 0) {
14         errors.password = "Password is required";
15     } else if (password.length < 8) {
16         errors.password = "Password must have 8 characters";
17     }
18     return errors;
19 };
20
21 // ...
22
23 export default (state = initialState, action) => {
24     if (action.type === "FORM_SUBMIT") {
25         const { email, password } = action.payload;
26         const values = {
27             email,
28             password
```

```
29      };
30      const errors = setErrors(email, password); // validate fields
31      return {
32        loginForm: {
33          values,
34          errors
35        }
36      };
37    }
38    return state;
39  };

```

## Wrapping Root Component with Provider

To make the store accessible in all the components, we need to wrap the root component with the `<Provider />` component from the `react-redux` package.

Also, notice that it's essential to import bootstrap CSS in the root component so that the styles are applied to the component.

```
1  import "bootstrap/dist/css/bootstrap.min.css";
2  import Card from "react-bootstrap/Card";
3  import { Provider } from "react-redux";
4  import { createStore } from "redux";
5
```

```
6      import reducer from "./reducer";
7
8      import LoginForm from "./LoginForm";
9
10     const store = createStore(reducer);
11
12     function App() {
13         return (
14             <Provider store={store}>
15                 <div className="App">
16                     <Card body>
17                         <LoginForm />
18                     </Card>
19                 </div>
20             </Provider>
21         );
22     }

```

## Connecting <LoginForm /> Component with Redux

First, we have to make this a controlled component—that is, the form data must be controlled by the component's internal state.

```
1      import React, { useState } from "react";
2
3      const LoginForm = props => {
4          const [email, setEmail] = useState("");

```

jsx

```
5  const [password, setPassword] = useState("");
6  return (
7    <Form>
8      <h2>Login</h2>
9      <hr />
10     <Form.Group controlId="formBasicEmail">
11       <Form.Label>Email address</Form.Label>
12       <Form.Control
13         type="email"
14         placeholder="Enter email"
15         onChange={e => setEmail(e.target.value)}
16       />
17       <Form.Control.Feedback type="invalid">
18         {props.loginForm.errors.email}
19       </Form.Control.Feedback>
20       <Form.Text className="text-muted">
21         We'll never share your email with anyone else.
22       </Form.Text>
23     </Form.Group>
24
25     <Form.Group controlId="formBasicPassword">
26       <Form.Label>Password</Form.Label>
27       <Form.Control
28         type="password"
29         placeholder="Password"
30         onChange={e => setPassword(e.target.value)}
31       />
32       <Form.Control.Feedback type="invalid">
33         {props.loginForm.errors.password}
34       </Form.Control.Feedback>
35     </Form.Group>
36     <Form.Group controlId="formBasicCheckbox">
37       <Form.Check type="checkbox" label="Check me out" />
38     </Form.Group>
39     <Button variant="primary" type="button">
```

```
40           Submit
41         </Button>
42       </Form>
43     );
44   };
```

We have discussed React hooks in an earlier ([guide](#))[/guides/change-page-background-color-each-route](#). Hooks are modern APIs provided by Redux to manage state in a functional component.

Next, we will connect the form to Redux using the `connect()` method.

jsx

```
1  // ...
2  import { connect } from "react-redux";
3
4  const LoginForm = props => {
5    const [email, setEmail] = useState("");
6    const [password, setPassword] = useState("");
7    return (
8      <Form>
9        <h2>Login</h2>
10       <hr />
11       <Form.Group controlId="formBasicEmail">
12         <Form.Label>Email address</Form.Label>
13         <Form.Control
14           type="email"
15           placeholder="Enter email"
16           isValid={props.loginForm.errors.email.length > 0}
```

```
17         isValid={  
18             props.loginForm.values.email &&  
19             props.loginForm.errors.email.length === 0  
20         }  
21         onChange={e => setEmail(e.target.value)}  
22     />  
23     <Form.Control.Feedback type="invalid">  
24         {props.loginForm.errors.email}  
25     </Form.Control.Feedback>  
26     <Form.Text className="text-muted">  
27         We'll never share your email with anyone else.  
28     </Form.Text>  
29 </Form.Group>  
30  
31 <Form.Group controlId="formBasicPassword">  
32     <Form.Label>Password</Form.Label>  
33     <Form.Control  
34         type="password"  
35         placeholder="Password"  
36         isValid={props.loginForm.errors.password.length > 0}  
37         onChange={e => setPassword(e.target.value)}  
38     />  
39         {props.loginForm.errors.password}  
40     </Form.Control.Feedback type="invalid">  
41         <Form.Control.Feedback type="invalid">  
42             {props.loginForm.errors.password}  
43         </Form.Control.Feedback>  
44     </Form.Group>  
45     <Form.Group controlId="formBasicCheckbox">  
46         <Form.Check type="checkbox" label="Check me out" />  
47     </Form.Group>  
48     <Button  
49         variant="primary"  
50     >
```

```
52         type="button"
53         onClick={() =>
54             props.dispatch({ type: "FORM_SUBMIT", payload: { email, password } })
55         }
56     >
57         Submit
58     </Button>
59     </Form>
60 );
61 };
62
63 const mapStateToProps = state => ({
64     loginForm: state.loginForm
65 });
66
67 export default connect(mapStateToProps)(LoginForm);
```

To display whether a field is valid or not, we have to pass a Boolean value to the `isValid` and `isValid` props of the `<Form.Control />` component. To display the error message, we will use the `<Form.Control.Feedback />` component with `type` prop as invalid.

```
1 <Form.Control.Feedback type="invalid">
2     {props.loginForm.errors.email}
3 </Form.Control.Feedback>
```

jsx

# Complete Source Code

## LoginForm.js

```
jsx
1  import React, { useState } from "react";
2  import Form from "react-bootstrap/Form";
3  import Button from "react-bootstrap/Button";
4  import { connect } from "react-redux";
5
6  const LoginForm = props => {
7      const [email, setEmail] = useState("");
8      const [password, setPassword] = useState("");
9      return (
10         <Form>
11             <h2>Login</h2>
12             <hr />
13             <Form.Group controlId="formBasicEmail">
14                 <Form.Label>Email address</Form.Label>
15                 <Form.Control
16                     type="email"
17                     placeholder="Enter email"
18                     isValid={props.loginForm.errors.email.length > 0}
19                     isValid={
20                         props.loginForm.values.email &&
21                         props.loginForm.errors.email.length === 0
22                     }
23                     onChange={e => setEmail(e.target.value)}
24                 />
25                 <Form.Control.Feedback type="invalid">
26                     {props.loginForm.errors.email}
27                 </Form.Control.Feedback>
28                 <Form.Text className="text-muted">
```

```
29           We'll never share your email with anyone else.
30       </Form.Text>
31   </Form.Group>
32
33   <Form.Group controlId="formBasicPassword">
34     <Form.Label>Password</Form.Label>
35     <Form.Control
36       type="password"
37       placeholder="Password"
38       isValid={props.loginForm.errors.password.length > 0}
39       isInvalid={
40         props.loginForm.values.password &&
41         props.loginForm.errors.password.length === 0
42       }
43       onChange={e => setPassword(e.target.value)}
44     />
45     <Form.Control.Feedback type="invalid">
46       {props.loginForm.errors.password}
47     </Form.Control.Feedback>
48   </Form.Group>
49   <Form.Group controlId="formBasicCheckbox">
50     <Form.Check type="checkbox" label="Check me out" />
51   </Form.Group>
52   <Button
53     variant="primary"
54     type="button"
55     onClick={() =>
56       props.dispatch({ type: "FORM_SUBMIT", payload: { email, password } })
57     }
58     >
59       Submit
60     </Button>
61   </Form>
62 );
63 };
```

```
64
65  const mapStateToProps = state => ({
66    loginForm: state.loginForm
67  });
68
69  export default connect(mapStateToProps)(LoginForm);
```

On form submission, we will dispatch an `FORM_SUBMIT` action with the form values in the `payload`.

## reducer.js

```
js
1  function validateEmail(email) {
2    var re = /^[^<>()\\[\\]\\\\.\\,;:\\s@"]+(\.[^<>()\\[\\]\\\\.\\,;:\\s@"]+)*|(.+)$/
3    return re.test(String(email).toLowerCase());
4  }
5
6  const initialState = {
7    loginForm: {
8      values: {
9        email: "",
10       password: ""
11      },
12      errors: {
13        email: "",
14        password: ""
15      }
16    }
17  };
18
19  const setErrors = (email, password) => {
```

```
20     let errors = { email: "", password: "" };
21     if (!email && email.length === 0) {
22         errors.email = "Email is required";
23     } else if (!validateEmail(email)) {
24         errors.email = "Email is invalid";
25     }
26     if (!password && password.length === 0) {
27         errors.password = "Password is required";
28     } else if (password.length < 8) {
29         errors.password = "Password must have 8 characters";
30     }
31     return errors;
32 };
33
34 export default (state = initialState, action) => {
35     if (action.type === "FORM_SUBMIT") {
36         const { email, password } = action.payload;
37         const values = {
38             email,
39             password
40         };
41         const errors = setErrors(email, password);
42         return {
43             loginForm: {
44                 values,
45                 errors
46             }
47         };
48     }
49     return state;
50 }
```

## index.js

```
import React from "react";
import ReactDOM from "react-dom";
1   import "bootstrap/dist/css/bootstrap.min.css";
2   import Card from "react-bootstrap/Card";
3   import { Provider } from "react-redux";
4   import { createStore } from "redux";

5
6   import reducer from "./reducer";
7
8   import LoginForm from "./LoginForm";
9
10  const store = createStore(reducer);

11  function App() {
12    return (
13      <Provider store={store}>
14        <div className="App">
15          <Card body>
16            <LoginForm />
17          </Card>
18        </div>
19      </Provider>
20    );
21  }

22
23  const rootElement = document.getElementById("root");
24  ReactDOM.render(<App />, rootElement);
25
26
27
```

## Conclusion

React Bootstrap allows us to quickly set up our application with a decent design and focus more on the business logic of the app. It also can be used as a starting point for building applications with complex UI as it's very flexible and can be highly customized.

I hope you like this guide. If you have any queries regarding this topic, feel free to contact me at [CodeAlphabet](#).

To learn more, check out [React Bootstrap Forms](#).