

Swatee Chand

Sep 15,

# React Redux Tutorial – Efficient Management of States in React

2017

Add to Bookmark

Email this Post

1.8K

0

## React Redux Tutorial

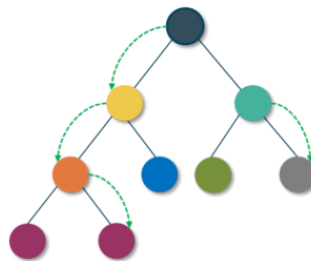
React is one of the most popular JavaScript libraries which is used for front-end development. It has made our application development easier and faster by providing the component based approach.

As you might know, it's not the complete framework but just the view part of the MVC (Model-View-Controller) framework. So, how do you keep track of the data and handle the events in the applications developed using React? Well, this is where Redux comes as a savior and handles the data flow of the application from the backend.

Through this blog on React Redux tutorial, I will explain everything you need to know on how to integrate Redux with React applications. Below are the topics I will be discussing under React Redux tutorial:

- Why Redux With React?
- What Is Redux?
- Advantages Of Redux
- Components Of Redux
- React with Redux

## Why Redux With React? – React Redux Tutorial



As I have already mentioned that React follows the component based approach, where the data flows through the components. In-fact, the data in React always flows from parent to child components which makes it *unidirectional*. This surely keeps our data organized and helps us in controlling the application better. Because of this, the application's state is contained in specific stores and as a result, the rest of the components remain loosely coupled. This makes our application more flexible leading to increased efficiency. That's why the communication from a parent component to a child component is convenient.

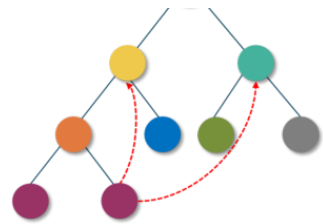
But what happens when we try to communicate from a non-parent component?

A child component can never pass data back up to the parent component. React does not provide any way for direct component-to-component communication. Even though React has features to support this approach, it is considered to be a poor practice. It is prone to errors and leads to spaghetti

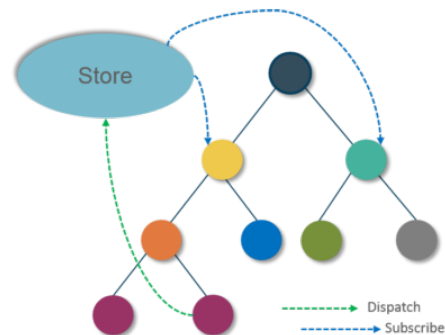


component?

A child component can never pass data back up to the parent component. React does not provide any way for direct component-to-component communication. Even though React has features to support this approach, it is considered to be a poor practice. It is prone to errors and leads to spaghetti code. So, how can two non-parent components pass data to each other?



This is where the React fails to provide a solution and Redux comes into the picture.



Redux provides a “**store**” as a solution to this problem. A store is a place where you can store all your application state together. Now the components can “*dispatch*” state changes to the store and not directly to the other components. Then the components that need the updates about the state changes can “*subscribe*” to the store.

Thus, with Redux, it becomes clear where the components get their state from as well as where should they send their states to. Now the component initiating the change does not have to worry about the list of components needing the state change and can simply dispatch the change to the store. This is how Redux makes the *data flow* easier.

## What Is Redux? – React Redux Tutorial

Just like React, Redux is also a library which is used widely for front-end development. It is basically a tool for managing both data-state and UI-state in JavaScript applications. Redux separates the application data and business logic into its own container in order to let React manage just the view. Rather than a traditional library or a framework, it’s an application data-flow architecture. It is most compatible with Single Page Applications (SPAs) where the management of the states over time can get complex.

Redux was created by *Dan Abramov* and *Andrew Clark* around June 2015. It was inspired by Facebook’s Flux and influenced by functional programming language *Elm*. Redux got popular very quickly because of its simplicity, small size (only 2 KB) and great documentation.

**GET REACT CERTIFIED  
TODAY!**

## Principles Of Redux

Redux follows three fundamental principles:

1. **Single source of truth:** The state of the entire application is stored in an object/ state tree within a single store. The single state tree makes it easier to keep track of the changes over time and debug or inspect the application. For a faster development cycle, it helps to persist the application’s state in development.
2. **State is read-only:** The only way to change the state is to trigger an action. An action is a plain JS object describing the change. Just like the state is the minimal representation of data, the action is the minimal representation of the change to that data. An action must have a type property (conventionally a String constant). All the changes are centralized and occur one by one in a strict order.
3. **Changes are made with pure functions:** In order to specify how the state tree is transformed by actions, you need pure functions. Pure functions are those whose return values depend solely on the values of their arguments. Reducers are just pure functions that take the previous state and an action and return the next state. You can have a single reducer in your application and as it grows, you can

(conventionally, a `String` constant). All the changes are centralized and occur one by one in a strict order.

3. **Changes are made with pure functions:** In order to specify how the state tree is transformed by actions, you need pure functions. Pure functions are those whose return values depend solely on the values of their arguments. Reducers are just pure functions that take the previous state and an action and return the next state. You can have a single reducer in your application and as it grows, you can split it off into smaller reducers. These smaller reducers will then manage specific parts of the state tree.

## Advantages Of Redux – React Redux Tutorial

Following are some of the major advantages of Redux:

- **Predictability of outcome** – Since there is always one source of truth, i.e. the store, there is no confusion about how to sync the current state with actions and other parts of the application.
- **Maintainability** – The code becomes easier to maintain with a predictable outcome and strict structure.
- **Server side rendering** – You just need to pass the store that is created on the server, to the client side. This is very useful for initial render and provides a better user experience as it optimizes the application performance.
- **Developer tools** – From actions to state changes, developers can track everything going on in the application in real time.
- **Community and ecosystem** – Redux has a huge community behind it which makes it even more captivating to use. A large community of talented individuals contribute to the betterment of the library and develop various applications with it.
- **Ease of testing** – Redux code are mostly functions which are small, pure and isolated. This makes the code testable and independent.
- **Organization** – Redux is very precise about how the code should be organized, this makes the code more consistent and easier when a team works with it.

## Components Of Redux – React Redux Tutorial

Redux has four components.

1. Action
2. Reducer
3. Store
4. View

Let us discuss them in detail:

- **Action** – The only way to change state content is by emitting an action. Actions are the plain JavaScript objects which are the main source of information used to send data (user interactions, internal events such as API calls, and form submissions) from the application to the store. The store receives information only from the actions. You have to send the actions to the store using `store.dispatch()`.

Internal actions are simple JavaScript objects that have a `type` property (usually `String` constant), describing the type of action and the entire information being sent to the store.

```
1 | {  
2 |   type: ADD_TODO,  
3 |   text  
4 | }
```

Actions are created using action creators which are the normal functions that return actions.

```
1 | function addTodo(text) {  
2 |   return {  
3 |     type: ADD_TODO,  
4 |     text  
5 |   }  
6 | }
```

To call actions anywhere in the app, use `dispatch()` method:

```
1 | dispatch(addTodo(text));
```

```

1 function addTodo(text) {
2   return {
3     type: ADD_TODO,
4     text
5   }
6 }

```

To call actions anywhere in the app, use `dispatch()` method:

```

1 dispatch(addTodo(text));

```

- **Reducer** – Actions describe the fact that *something happened*, but don't specify how the application's state changes in response. This is the job of reducers. It is based on the array reduce method, where it accepts a callback (reducer) and lets you get a single value out of multiple values, sums of integers, or an accumulation of streams of values. In Redux, reducers are functions (pure) that take the current state of the application and an action and then return a new state. Understanding how reducers work is important because they perform most of the work.

```

1 function reducer(state = initialState, action) {
2   switch (action.type) {
3     case ADD_TODO:
4       return Object.assign({}, state,
5         { todos: [ ...state.todos,
6           {
7             text: action.text,
8             completed: false
9           }
10        ]
11      });
12   default:
13     return state
14   }
15 }

```

- **Store** – A store is a JavaScript object which can hold the application's state and provide a few helper methods to access the state, dispatch actions and register listeners. The entire state/ object tree of an application is saved in a single store. As a result of this, Redux is very simple and predictable. We can pass middleware to the store to handle processing of data as well as to keep a log of various actions that change the state of stores. All the actions return a new state via reducers.

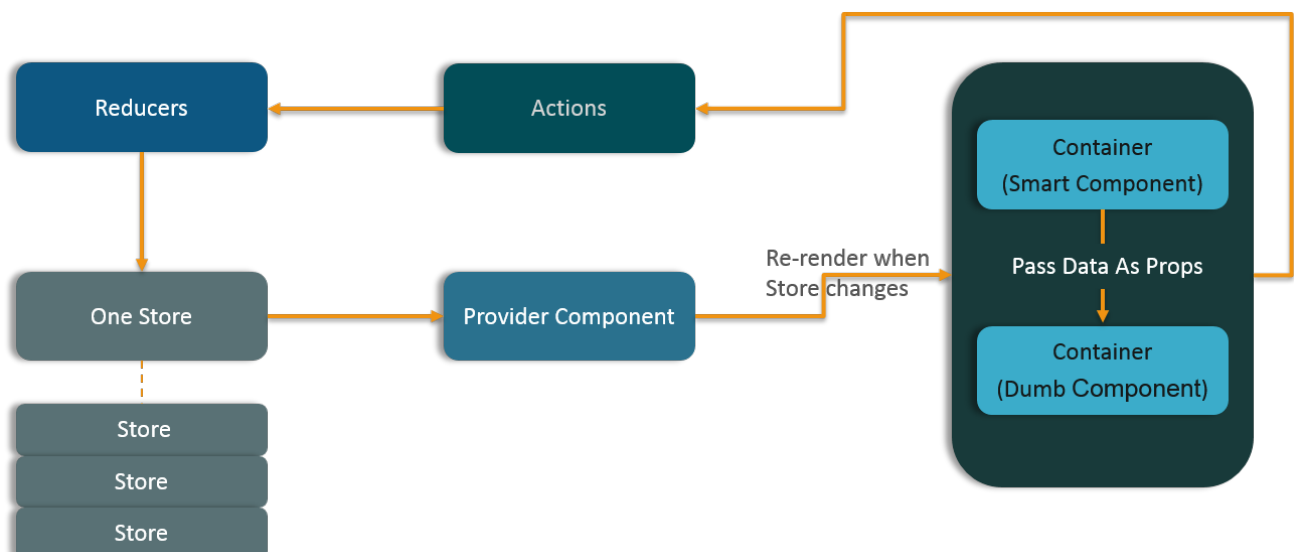
```

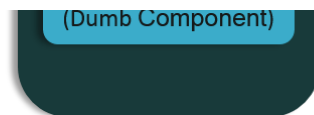
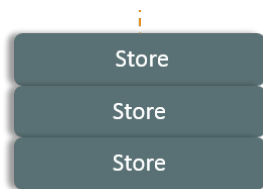
1 import { createStore } from 'redux'
2 import todoApp from './reducers'
3
4 let store = createStore(reducer);

```

- **View** – Smart and dumb components together build up the view. The only purpose of the view is to display the data passed down by the store. The smart components are in charge of the actions. The dumb components underneath the smart components notify them in case they need to trigger the action. The smart components, in turn, pass down the props which the dumb components treat as callback actions.

Following is a diagram which shows how the data actually flows through all the above-described components in Redux.





## React With Redux – React Redux Tutorial

Now that you are familiar with Redux and its components, let's now see how you can integrate it with a React application.

**STEP 1:** You need to setup the basic react, webpack, babel setup. Following are the dependencies we are using in this application.

```
1  "dependencies": {
2    "babel-core": "^6.10.4",
3    "babel-loader": "^6.2.4",
4    "babel-polyfill": "^6.9.1",
5    "babel-preset-es2015": "^6.9.0",
6    "babel-preset-react": "^6.11.1",
7    "babel-register": "^6.9.0",
8    "cross-env": "^1.0.8",
9    "css-loader": "^0.23.1",
10   "expect": "^1.20.1",
11   "node-libs-browser": "^1.0.0",
12   "node-sass": "^3.8.0",
13   "react": "^15.1.0",
14   "react-addons-test-utils": "^15.1.0",
15   "react-dom": "^15.1.0",
16   "react-redux": "^4.4.5",
17   "redux": "^3.5.2",
18   "redux-logger": "^2.6.1",
19   "redux-promise": "^0.5.3",
20   "redux-thunk": "^2.1.0",
21   "sass-loader": "^4.0.0",
22   "style-loader": "^0.13.1",
23   "webpack": "^1.13.1",
24   "webpack-dev-middleware": "^1.6.1",
25   "webpack-dev-server": "^1.14.1",
26   "webpack-hot-middleware": "^2.11.0"
27 },
```

**STEP 2:** Once you are done with installing the dependencies, then create **components** folder in **src** folder. Within that create **App.js** file.

```
1  import React from 'react';
2  import UserList from '../containers/user-list';
3  import UserDetails from '../containers/user-detail';
4  require('../../scss/style.scss');
5
6  const App = () => (
7    <div>
8      <h2>User List</h2>
9      <UserList />
10     <hr />
11     <h2>User Details</h2>
12     <UserDetails />
13   </div>
14 );
15
16 export default App;
```

**STEP 3:** Next create a new **actions** folder and create **index.js** in it.

```
1  export const selectUser = (user) => {
2    console.log("You clicked on user: ", user.first);
3    return {
4      type: 'USER_SELECTED',
5      payload: user
6    };
7  };
```

**STEP 3:** Now create **user-details.js** in a new folder called **containers**.

```
1  import React, {Component} from 'react';
2  import {connect} from 'react-redux';
3
4  class UserDetails extends Component {
5    render() {
6      if (!this.props.user) {
7        return (<div>Select a user...</div>);
```

**STEP 3:** Now create **user-details.js** in a new folder called **containers**.

```
1 import React, {Component} from 'react';
2 import {connect} from 'react-redux';
3
4 class UserDetails extends Component {
5   render() {
6     if (!this.props.user) {
7       return (<div>Select a user...</div>);
8     }
9     return (
10       <div>
11         <img height="150" width="150" src={this.props.user.thumbnail} />
12         <h2>{this.props.user.first} {this.props.user.last}</h2>
13         <h3>Age: {this.props.user.age}</h3>
14         <h3>Description: {this.props.user.description}</h3>
15       </div>
16     );
17   }
18 }
19
20 function mapStateToProps(state) {
21   return {
22     user: state.activeUser
23   };
24 }
25
26 export default connect(mapStateToProps)(UserDetails);
```

**STEP 4:** Inside the same folder create **user-list.js** file.

```
1 import React, {Component} from 'react';
2 import {bindActionCreators} from 'redux';
3 import {connect} from 'react-redux';
4 import {selectUser} from '../actions/index'
5 class UserList extends Component {
6   renderList() {
7     return this.props.users.map((user) => {
8       return (
9         <li key={user.id}
10           onClick={() => this.props.selectUser(user)}
11           >
12           {user.first} {user.last}
13         </li>
14       );
15     });
16   }
17   render() {
18     return (
19       <ul>
20         {this.renderList()}
21       </ul>
22     );
23   }
24 }
25 function mapStateToProps(state) {
26   return {
27     users: state.users
28   };
29 }
30 function matchDispatchToProps(dispatch){
31   return bindActionCreators({selectUser: selectUser}, dispatch);
32 }
33 export default connect(mapStateToProps, matchDispatchToProps)(UserList);
```

**STEP 5:** Now create **reducers** folder and create **index.js** within it.

```
1 import {combineReducers} from 'redux';
2 import UserReducer from './reducer-users';
3 import ActiveUserReducer from './reducer-active-user';
4
5 const allReducers = combineReducers({
6   users: UserReducer,
7   activeUser: ActiveUserReducer
8 });
9 export default allReducers
```

**STEP 6:** Within the same **reducers** folder, create **reducer-users.js** file.

```
1 export default function () {
2   return [
3     {
4       id: 1,
5       first: "Maxx",
6       last: "Flinn",
7       age: 17,
8       description: "Loves basketball",
9       thumbnail: "https://goo.gl/1KNpiy"
10     },
11     {
12       id: 2,
13       first: "John",
14       last: "Doe",
15       age: 30,
16       description: "Software Engineer",
17       thumbnail: "https://goo.gl/1KNpiy"
18     },
19     {
20       id: 3,
21       first: "Jane",
22       last: "Smith",
23       age: 25,
24       description: "Marketing Specialist",
25       thumbnail: "https://goo.gl/1KNpiy"
26     },
27     {
28       id: 4,
29       first: "Mike",
30       last: "Johnson",
31       age: 40,
32       description: "Business Owner",
33       thumbnail: "https://goo.gl/1KNpiy"
34     },
35     {
36       id: 5,
37       first: "Emily",
38       last: "Brown",
39       age: 28,
40       description: "Data Analyst",
41       thumbnail: "https://goo.gl/1KNpiy"
42     },
43     {
44       id: 6,
45       first: "David",
46       last: "Wilson",
47       age: 35,
48       description: "Product Manager",
49       thumbnail: "https://goo.gl/1KNpiy"
50     },
51     {
52       id: 7,
53       first: "Sophia",
54       last: "Garcia",
55       age: 22,
56       description: "UX Designer",
57       thumbnail: "https://goo.gl/1KNpiy"
58     },
59     {
60       id: 8,
61       first: "Daniel",
62       last: "Martinez",
63       age: 38,
64       description: "Sales Representative",
65       thumbnail: "https://goo.gl/1KNpiy"
66     },
67     {
68       id: 9,
69       first: "Olivia",
70       last: "Hernandez",
71       age: 27,
72       description: "HR Manager",
73       thumbnail: "https://goo.gl/1KNpiy"
74     },
75     {
76       id: 10,
77       first: "Liam",
78       last: "Lopez",
79       age: 32,
80       description: "Operations Manager",
81       thumbnail: "https://goo.gl/1KNpiy"
82     },
83     {
84       id: 11,
85       first: "Ava",
86       last: "Gonzalez",
87       age: 24,
88       description: "Quality Assurance",
89       thumbnail: "https://goo.gl/1KNpiy"
90     },
91     {
92       id: 12,
93       first: "Noah",
94       last: "Perez",
95       age: 31,
96       description: "Systems Administrator",
97       thumbnail: "https://goo.gl/1KNpiy"
98     },
99     {
100      id: 13,
101      first: "Isabella",
102      last: "Robinson",
103      age: 29,
104      description: "Business Development",
105      thumbnail: "https://goo.gl/1KNpiy"
106    },
107    {
108      id: 14,
109      first: "Mason",
110      last: "Clark",
111      age: 33,
112      description: "Project Manager",
113      thumbnail: "https://goo.gl/1KNpiy"
114    },
115    {
116      id: 15,
117      first: "Mia",
118      last: "Rodriguez",
119      age: 26,
120      description: "Customer Support",
121      thumbnail: "https://goo.gl/1KNpiy"
122    },
123    {
124      id: 16,
125      first: "Ethan",
126      last: "Walker",
127      age: 36,
128      description: "Finance Analyst",
129      thumbnail: "https://goo.gl/1KNpiy"
130    },
131    {
132      id: 17,
133      first: "Charlotte",
134      last: "Young",
135      age: 23,
136      description: "Marketing Coordinator",
137      thumbnail: "https://goo.gl/1KNpiy"
138    },
139    {
140      id: 18,
141      first: "Alexander",
142      last: "Allen",
143      age: 39,
144      description: "Operations Director",
145      thumbnail: "https://goo.gl/1KNpiy"
146    },
147    {
148      id: 19,
149      first: "Amelia",
150      last: "King",
151      age: 21,
152      description: "UX Researcher",
153      thumbnail: "https://goo.gl/1KNpiy"
154    },
155    {
156      id: 20,
157      first: "Benjamin",
158      last: "Scott",
159      age: 34,
160      description: "Product Designer",
161      thumbnail: "https://goo.gl/1KNpiy"
162    },
163    {
164      id: 21,
165      first: "Harper",
166      last: "Torres",
167      age: 28,
168      description: "Sales Manager",
169      thumbnail: "https://goo.gl/1KNpiy"
170    },
171    {
172      id: 22,
173      first: "Lucas",
174      last: "Harris",
175      age: 37,
176      description: "Business Analyst",
177      thumbnail: "https://goo.gl/1KNpiy"
178    },
179    {
180      id: 23,
181      first: "Evelyn",
182      last: "Wright",
183      age: 25,
184      description: "Marketing Strategist",
185      thumbnail: "https://goo.gl/1KNpiy"
186    },
187    {
188      id: 24,
189      first: "Gabriel",
190      last: "Lopez",
191      age: 30,
192      description: "Systems Engineer",
193      thumbnail: "https://goo.gl/1KNpiy"
194    },
195    {
196      id: 25,
197      first: "Sofia",
198      last: "Gonzalez",
199      age: 27,
200      description: "Customer Success",
201      thumbnail: "https://goo.gl/1KNpiy"
202    },
203    {
204      id: 26,
205      first: "Nathan",
206      last: "Perez",
207      age: 32,
208      description: "Finance Manager",
209      thumbnail: "https://goo.gl/1KNpiy"
210    },
211    {
212      id: 27,
213      first: "Avery",
214      last: "Robinson",
215      age: 24,
216      description: "Product Marketing",
217      thumbnail: "https://goo.gl/1KNpiy"
218    },
219    {
220      id: 28,
221      first: "Caleb",
222      last: "Clark",
223      age: 35,
224      description: "Operations Manager",
225      thumbnail: "https://goo.gl/1KNpiy"
226    },
227    {
228      id: 29,
229      first: "Madison",
230      last: "Rodriguez",
231      age: 26,
232      description: "Quality Assurance",
233      thumbnail: "https://goo.gl/1KNpiy"
234    },
235    {
236      id: 30,
237      first: "Elijah",
238      last: "Walker",
239      age: 38,
240      description: "Business Development",
241      thumbnail: "https://goo.gl/1KNpiy"
242    },
243    {
244      id: 31,
245      first: "Abigail",
246      last: "Harris",
247      age: 22,
248      description: "UX Researcher",
249      thumbnail: "https://goo.gl/1KNpiy"
250    },
251    {
252      id: 32,
253      first: "Carter",
254      last: "Lopez",
255      age: 31,
256      description: "Product Designer",
257      thumbnail: "https://goo.gl/1KNpiy"
258    },
259    {
260      id: 33,
261      first: "Elena",
262      last: "Torres",
263      age: 29,
264      description: "Sales Manager",
265      thumbnail: "https://goo.gl/1KNpiy"
266    },
267    {
268      id: 34,
269      first: "Grayson",
270      last: "Harris",
271      age: 36,
272      description: "Business Analyst",
273      thumbnail: "https://goo.gl/1KNpiy"
274    },
275    {
276      id: 35,
277      first: "Scarlett",
278      last: "Wright",
279      age: 23,
280      description: "Marketing Coordinator",
281      thumbnail: "https://goo.gl/1KNpiy"
282    },
283    {
284      id: 36,
285      first: "Jaxon",
286      last: "Lopez",
287      age: 39,
288      description: "Operations Director",
289      thumbnail: "https://goo.gl/1KNpiy"
290    },
291    {
292      id: 37,
293      first: "Lillian",
294      last: "Gonzalez",
295      age: 21,
296      description: "UX Researcher",
297      thumbnail: "https://goo.gl/1KNpiy"
298    },
299    {
300      id: 38,
301      first: "Oscar",
302      last: "Perez",
303      age: 34,
304      description: "Product Designer",
305      thumbnail: "https://goo.gl/1KNpiy"
306    },
307    {
308      id: 39,
309      first: "Paisley",
310      last: "Robinson",
311      age: 28,
312      description: "Sales Manager",
313      thumbnail: "https://goo.gl/1KNpiy"
314    },
315    {
316      id: 40,
317      first: "Ryder",
318      last: "Clark",
319      age: 37,
320      description: "Business Analyst",
321      thumbnail: "https://goo.gl/1KNpiy"
322    },
323    {
324      id: 41,
325      first: "Violet",
326      last: "Rodriguez",
327      age: 25,
328      description: "Marketing Strategist",
329      thumbnail: "https://goo.gl/1KNpiy"
330    },
331    {
332      id: 42,
333      first: "Wyatt",
334      last: "Walker",
335      age: 30,
336      description: "Systems Engineer",
337      thumbnail: "https://goo.gl/1KNpiy"
338    },
339    {
340      id: 43,
341      first: "Zoe",
342      last: "Gonzalez",
343      age: 27,
344      description: "Customer Success",
345      thumbnail: "https://goo.gl/1KNpiy"
346    },
347    {
348      id: 44,
349      first: "Adrian",
350      last: "Perez",
351      age: 32,
352      description: "Finance Manager",
353      thumbnail: "https://goo.gl/1KNpiy"
354    },
355    {
356      id: 45,
357      first: "Bella",
358      last: "Robinson",
359      age: 24,
360      description: "Product Marketing",
361      thumbnail: "https://goo.gl/1KNpiy"
362    },
363    {
364      id: 46,
365      first: "Cameron",
366      last: "Clark",
367      age: 35,
368      description: "Operations Manager",
369      thumbnail: "https://goo.gl/1KNpiy"
370    },
371    {
372      id: 47,
373      first: "Diana",
374      last: "Rodriguez",
375      age: 26,
376      description: "Quality Assurance",
377      thumbnail: "https://goo.gl/1KNpiy"
378    },
379    {
380      id: 48,
381      first: "Evan",
382      last: "Walker",
383      age: 38,
384      description: "Business Development",
385      thumbnail: "https://goo.gl/1KNpiy"
386    },
387    {
388      id: 49,
389      first: "Gabriella",
390      last: "Harris",
391      age: 22,
392      description: "UX Researcher",
393      thumbnail: "https://goo.gl/1KNpiy"
394    },
395    {
396      id: 50,
397      first: "Hudson",
398      last: "Lopez",
399      age: 31,
400      description: "Product Designer",
401      thumbnail: "https://goo.gl/1KNpiy"
402    },
403    {
404      id: 51,
405      first: "Isabella",
406      last: "Torres",
407      age: 29,
408      description: "Sales Manager",
409      thumbnail: "https://goo.gl/1KNpiy"
410    },
411    {
412      id: 52,
413      first: "Javier",
414      last: "Harris",
415      age: 36,
416      description: "Business Analyst",
417      thumbnail: "https://goo.gl/1KNpiy"
418    },
419    {
420      id: 53,
421      first: "Kenneth",
422      last: "Wright",
423      age: 23,
424      description: "Marketing Coordinator",
425      thumbnail: "https://goo.gl/1KNpiy"
426    },
427    {
428      id: 54,
429      first: "Liam",
430      last: "Lopez",
431      age: 39,
432      description: "Operations Director",
433      thumbnail: "https://goo.gl/1KNpiy"
434    },
435    {
436      id: 55,
437      first: "Mia",
438      last: "Gonzalez",
439      age: 21,
440      description: "UX Researcher",
441      thumbnail: "https://goo.gl/1KNpiy"
442    },
443    {
444      id: 56,
445      first: "Noah",
446      last: "Perez",
447      age: 34,
448      description: "Product Designer",
449      thumbnail: "https://goo.gl/1KNpiy"
450    },
451    {
452      id: 57,
453      first: "Olivia",
454      last: "Robinson",
455      age: 28,
456      description: "Sales Manager",
457      thumbnail: "https://goo.gl/1KNpiy"
458    },
459    {
460      id: 58,
461      first: "Parker",
462      last: "Clark",
463      age: 37,
464      description: "Business Analyst",
465      thumbnail: "https://goo.gl/1KNpiy"
466    },
467    {
468      id: 59,
469      first: "Quinn",
470      last: "Rodriguez",
471      age: 25,
472      description: "Marketing Strategist",
473      thumbnail: "https://goo.gl/1KNpiy"
474    },
475    {
476      id: 60,
477      first: "Ryder",
478      last: "Walker",
479      age: 30,
480      description: "Systems Engineer",
481      thumbnail: "https://goo.gl/1KNpiy"
482    },
483    {
484      id: 61,
485      first: "Sophia",
486      last: "Gonzalez",
487      age: 27,
488      description: "Customer Success",
489      thumbnail: "https://goo.gl/1KNpiy"
490    },
491    {
492      id: 62,
493      first: "Theodore",
494      last: "Perez",
495      age: 32,
496      description: "Finance Manager",
497      thumbnail: "https://goo.gl/1KNpiy"
498    },
499    {
500      id: 63,
501      first: "Uma",
502      last: "Robinson",
503      age: 24,
504      description: "Product Marketing",
505      thumbnail: "https://goo.gl/1KNpiy"
506    },
507    {
508      id: 64,
509      first: "Victor",
510      last: "Clark",
511      age: 35,
512      description: "Operations Manager",
513      thumbnail: "https://goo.gl/1KNpiy"
514    },
515    {
516      id: 65,
517      first: "Wendy",
518      last: "Rodriguez",
519      age: 26,
520      description: "Quality Assurance",
521      thumbnail: "https://goo.gl/1KNpiy"
522    },
523    {
524      id: 66,
525      first: "Xavier",
526      last: "Walker",
527      age: 38,
528      description: "Business Development",
529      thumbnail: "https://goo.gl/1KNpiy"
530    },
531    {
532      id: 67,
533      first: "Yara",
534      last: "Harris",
535      age: 22,
536      description: "UX Researcher",
537      thumbnail: "https://goo.gl/1KNpiy"
538    },
539    {
540      id: 68,
541      first: "Zachary",
542      last: "Lopez",
543      age: 31,
544      description: "Product Designer",
545      thumbnail: "https://goo.gl/1KNpiy"
546    },
547    {
548      id: 69,
549      first: "Adrian",
550      last: "Torres",
551      age: 29,
552      description: "Sales Manager",
553      thumbnail: "https://goo.gl/1KNpiy"
554    },
555    {
556      id: 70,
557      first: "Bella",
558      last: "Harris",
559      age: 36,
560      description: "Business Analyst",
561      thumbnail: "https://goo.gl/1KNpiy"
562    },
563    {
564      id: 71,
565      first: "Cameron",
566      last: "Wright",
567      age: 23,
568      description: "Marketing Coordinator",
569      thumbnail: "https://goo.gl/1KNpiy"
570    },
571    {
572      id: 72,
573      first: "Diana",
574      last: "Lopez",
575      age: 39,
576      description: "Operations Director",
577      thumbnail: "https://goo.gl/1KNpiy"
578    },
579    {
580      id: 73,
581      first: "Evan",
582      last: "Gonzalez",
583      age: 21,
584      description: "UX Researcher",
585      thumbnail: "https://goo.gl/1KNpiy"
586    },
587    {
588      id: 74,
589      first: "Gabriella",
590      last: "Perez",
591      age: 34,
592      description: "Product Designer",
593      thumbnail: "https://goo.gl/1KNpiy"
594    },
595    {
596      id: 75,
597      first: "Hudson",
598      last: "Robinson",
599      age: 28,
600      description: "Sales Manager",
601      thumbnail: "https://goo.gl/1KNpiy"
602    },
603    {
604      id: 76,
605      first: "Isabella",
606      last: "Clark",
607      age: 37,
608      description: "Business Analyst",
609      thumbnail: "https://goo.gl/1KNpiy"
610    },
611    {
612      id: 77,
613      first: "Javier",
614      last: "Rodriguez",
615      age: 25,
616      description: "Marketing Strategist",
617      thumbnail: "https://goo.gl/1KNpiy"
618    },
619    {
620      id: 78,
621      first: "Kenneth",
622      last: "Walker",
623      age: 30,
624      description: "Systems Engineer",
625      thumbnail: "https://goo.gl/1KNpiy"
626    },
627    {
628      id: 79,
629      first: "Liam",
630      last: "Gonzalez",
631      age: 27,
632      description: "Customer Success",
633      thumbnail: "https://goo.gl/1KNpiy"
634    },
635    {
636      id: 80,
637      first: "Madison",
638      last: "Perez",
639      age: 32,
640      description: "Finance Manager",
641      thumbnail: "https://goo.gl/1KNpiy"
642    },
643    {
644      id: 81,
645      first: "Nathan",
646      last: "Robinson",
647      age: 24,
648      description: "Product Marketing",
649      thumbnail: "https://goo.gl/1KNpiy"
650    },
651    {
652      id: 82,
653      first: "Olivia",
654      last: "Clark",
655      age: 35,
656      description: "Operations Manager",
657      thumbnail: "https://goo.gl/1KNpiy"
658    },
659    {
660      id: 83,
661      first: "Parker",
662      last: "Rodriguez",
663      age: 26,
664      description: "Quality Assurance",
665      thumbnail: "https://goo.gl/1KNpiy"
666    },
667    {
668      id: 84,
669      first: "Quinn",
670      last: "Walker",
671      age: 38,
672      description: "Business Development",
673      thumbnail: "https://goo.gl/1KNpiy"
674    },
675    {
676      id: 85,
677      first: "Ryder",
678      last: "Harris",
679      age: 22,
680      description: "UX Researcher",
681      thumbnail: "https://goo.gl/1KNpiy"
682    },
683    {
684      id: 86,
685      first: "Sophia",
686      last: "Lopez",
687      age: 31,
688      description: "Product Designer",
689      thumbnail: "https://goo.gl/1KNpiy"
690    },
691    {
692      id: 87,
693      first: "Theodore",
694      last: "Torres",
695      age: 29,
696      description: "Sales Manager",
697      thumbnail: "https://goo.gl/1KNpiy"
698    },
699    {
700      id: 88,
701      first: "Uma",
702      last: "Harris",
703      age: 36,
704      description: "Business Analyst",
705      thumbnail: "https://goo.gl/1KNpiy"
706    },
707    {
708      id: 89,
709      first: "Victor",
710      last: "Wright",
711      age: 23,
712      description: "Marketing Coordinator",
713      thumbnail: "https://goo.gl/1KNpiy"
714    },
715    {
716      id: 90,
717      first: "Wendy",
718      last: "Lopez",
719      age: 39,
720      description: "Operations Director",
721      thumbnail: "https://goo.gl/1KNpiy"
722    },
723    {
724      id: 91,
725      first: "Xavier",
726      last: "Gonzalez",
727      age: 21,
728      description: "UX Researcher",
729      thumbnail: "https://goo.gl/1KNpiy"
730    },
731    {
732      id: 92,
733      first: "Yara",
734      last: "Perez",
735      age: 34,
736      description: "Product Designer",
737      thumbnail: "https://goo.gl/1KNpiy"
738    },
739    {
740      id: 93,
741      first: "Zachary",
742      last: "Robinson",
743      age: 28,
744      description: "Sales Manager",
745      thumbnail: "https://goo.gl/1KNpiy"
746    },
747    {
748      id: 94,
749      first: "Adrian",
750      last: "Clark",
751      age: 37,
752      description: "Business Analyst",
753      thumbnail: "https://goo.gl/1KNpiy"
754    },
755    {
756      id: 95,
757      first: "Bella",
758      last: "Rodriguez",
759      age: 25,
760      description: "Marketing Strategist",
761      thumbnail: "https://goo.gl/1KNpiy"
762    },
763    {
764      id: 96,
765      first: "Cameron",
766      last: "Walker",
767      age: 30,
768      description: "Systems Engineer",
769      thumbnail: "https://goo.gl/1KNpiy"
770    },
771    {
772      id: 97,
773      first: "Diana",
774      last: "Gonzalez",
775      age: 27,
776      description: "Customer Success",
777      thumbnail: "https://goo.gl/1KNpiy"
778    },
779    {
780      id: 98,
781      first: "Evan",
782      last: "Perez",
783      age: 32,
784      description: "Finance Manager",
785      thumbnail: "https://goo.gl/1KNpiy"
786    },
787    {
788      id: 99,
789      first: "Gabriella",
790      last: "Robinson",
791      age: 24,
792      description: "Product Marketing",
793      thumbnail: "https://goo.gl/1KNpiy"
794    },
795    {
796      id: 100,
797      first: "Hudson",
798      last: "Clark",
799      age: 35,
800      description: "Operations Manager",
801      thumbnail: "https://goo.gl/1KNpiy"
802    },
803    {
804      id: 101,
805      first: "Isabella",
806      last: "Rodriguez",
807      age: 26,
808      description: "Quality Assurance",
809      thumbnail: "https://goo.gl/1KNpiy"
810    },
811    {
812      id: 102,
813      first: "Javier",
814      last: "Walker",
815      age: 38,
816      description: "Business Development",
817      thumbnail: "https://goo.gl/1KNpiy"
818    },
819    {
820      id: 103,
821      first: "Kenneth",
822      last: "Harris",
823      age: 22,
824      description: "UX Researcher",
825      thumbnail: "https://goo.gl/1KNpiy"
826    },
827    {
828      id: 104,
829      first: "Liam",
830      last: "Lopez",
831      age: 31,
832      description: "Product Designer",
833      thumbnail: "https://goo.gl/1KNpiy"
834    },
835    {
836      id: 105,
837      first: "Madison",
838      last: "Torres",
839      age: 29,
840      description: "Sales Manager",
841      thumbnail: "https://goo.gl/1KNpiy"
842    },
843    {
844      id: 106,
845      first: "Nathan",
846      last: "Harris",
847      age: 36,
848      description: "Business Analyst",
849      thumbnail: "https://goo.gl/1KNpiy"
850    },
851    {
852      id: 107,
853      first: "Olivia",
854      last: "Wright",
855      age: 23,
856      description: "Marketing Coordinator",
857      thumbnail: "https://goo.gl/1KNpiy"
858    },
859    {
860      id: 108,
861      first: "Parker",
862      last: "Lopez",
863      age: 39,
864      description: "Operations Director",
865      thumbnail: "https://goo.gl/1KNpiy"
866    },
867    {
868      id: 109,
869      first: "Quinn",
870      last: "Gonzalez",
871      age: 21,
872      description: "UX Researcher",
873      thumbnail: "https://goo.gl/1KNpiy"
874    },
875    {
876      id: 110,
877      first: "Ryder",
878      last: "Perez",
879      age: 34,
880      description: "Product Designer",
881      thumbnail: "https://goo.gl/1KNpiy"
882    },
883    {
884      id: 111,
885      first: "Sophia",
886      last: "Robinson",
887      age: 28,
888      description: "Sales Manager",
889      thumbnail: "https://goo.gl/1KNpiy"
890    },
891    {
892      id: 112,
893      first: "Theodore",
894      last: "Clark",
895      age: 37,
896      description: "Business Analyst",
897      thumbnail: "https://goo.gl/1KNpiy"
898    },
899    {
900      id: 113,
901      first: "Uma",
902      last: "Rodriguez",
903      age: 25,
904      description: "Marketing Strategist",
905      thumbnail: "https://goo.gl/1KNpiy"
906    },
907    {
908      id: 114,
909      first: "Victor",
910      last: "Walker",
911      age: 30,
912      description: "Systems Engineer",
913      thumbnail: "https://goo.gl/1KNpiy"
914    },
915    {
916      id: 115,
917      first: "Wendy",
918      last: "Gonzalez",
919      age: 27,
920      description: "Customer Success",
921      thumbnail: "https://goo.gl/1KNpiy"
922    },
923    {
924      id: 116,
925      first: "Xavier",
926      last: "Perez",
927      age: 32,
928      description: "Finance Manager",
929      thumbnail: "https://goo.gl/1KNpiy"
930    },
931    {
932      id: 117,
933      first: "Yara",
934      last: "Robinson",
935      age: 24,
936      description: "Product Marketing",
937      thumbnail: "https://goo.gl/1KNpiy"
938    },
939    {
940      id: 118,
941      first: "Zachary",
942      last: "Clark",
943      age: 35,
944      description: "Operations Manager",
945      thumbnail: "https://goo.gl/1KNpiy"
946    },
947    {
948      id: 119,
949      first: "Adrian",
950      last: "Rodriguez",
951      age: 26,
952      description: "Quality Assurance",
953      thumbnail: "https://goo.gl/1KNpiy"
954    },
955    {
956      id: 120,
957      first: "Bella",
958      last: "Walker",
959      age: 38,
960      description: "Business Development",
961      thumbnail: "https://goo.gl/1KNpiy"
962    },
963    {
964      id: 121,
965      first: "Cameron",
966      last: "Harris",
967      age: 22,
968      description: "UX Researcher",
969      thumbnail: "https://goo.gl/1KNpiy"
970    },
971    {
972      id: 122,
973      first: "Diana",
974      last: "Lopez",
975      age: 31,
976      description: "Product Designer",
977      thumbnail: "https://goo.gl/1KNpiy"
978    },
979    {
980      id: 123,
981      first: "Evan",
982      last: "Torres",
983      age: 29,
984      description: "Sales Manager",
985      thumbnail: "https://goo.gl/1KNpiy"
986    },
987    {
988      id: 124,
989      first: "Gabriella",
990      last: "Harris",
991      age: 36,
992      description: "Business Analyst",
993      thumbnail: "https://goo.gl/1KNpiy"
994    },
995    {
996      id: 125,
997      first: "Hudson",
998      last: "Wright",
999      age: 23,
1000     description: "Marketing Coordinator",
1001     thumbnail: "https://goo.gl/1KNpiy"
1002   },
1003   {
1004     id: 126,
1005     first: "Isabella",
1006     last: "Lopez",
1007     age: 39,
1008     description: "Operations Director",
1009     thumbnail: "https://goo.gl/1KNpiy"
1010   },
1011   {
1012     id: 127,
1013     first: "Javier",
1014     last: "Gonzalez",
1015     age: 21,
1016     description: "UX Researcher",
1017     thumbnail: "https://goo.gl/1KNpiy"
1018   },
1019   {
1020     id: 128,
1021     first: "Kenneth",
1022     last: "Perez",
1023     age: 34,
1024     description: "Product Designer",
1025     thumbnail: "https://goo.gl/1KNpiy"
1026   },
1027   {
1028     id: 129,
1029     first: "Liam",
1030     last: "Robinson",
1031     age: 28,
1032     description: "Sales Manager",
1033     thumbnail: "https://goo.gl/1KNpiy"
1034   },
1035   {
1036     id: 130,
1037     first: "Madison",
1038     last: "Clark",
1039     age: 37,
1040     description: "Business Analyst",
1041     thumbnail: "https://goo.gl/1KNpiy"
1042   },
1043   {
1044     id: 131,
1045     first: "Nathan",
1046     last: "Rodriguez",
1047     age: 25,
1048     description: "Marketing Strategist",
1049     thumbnail: "https://goo.gl/1KNpiy"
1050   },
1051   {
1052     id: 132,
1053     first: "Olivia",
1054     last: "Walker",
1055     age: 30,
1056     description: "Systems Engineer",
1057     thumbnail: "https://goo.gl/1KNpiy"
1058   },
1059   {
1060     id: 133,
1061     first: "Parker",
1062     last: "Gonzalez",
1063     age: 27,
1064     description: "Customer Success",
1065     thumbnail: "https://goo.gl/1KNpiy"
1066   },
1067   {
1068     id: 134,
1069     first: "Quinn",
1070     last: "Perez",
1071     age: 32,
1072     description: "Finance Manager",
107
```

```

2   return [
3     {
4       id: 1,
5       first: "Maxx",
6       last: "Flinn",
7       age: 17,
8       description: "Loves basketball",
9       thumbnail: "https://goo.gl/1KNpiy"
10    },
11    {
12      id: 2,
13      first: "Allen",
14      last: "Matt",
15      age: 25,
16      description: "Food Junky.",
17      thumbnail: "https://goo.gl/rNLgww"
18    },
19    {
20      id: 3,
21      first: "Kris",
22      last: "Chen",
23      age: 23,
24      description: "Music Lover.",
25      thumbnail: "https://goo.gl/EVbPHb"
26    }
27  ]
28 }

```

**STEP 7:** Now within **reducers** folder create a **reducer-active-user.js** file.

```

1  export default function (state = null, action) {
2    switch (action.type) {
3      case 'USER_SELECTED':
4        return action.payload;
5        break;
6    }
7    return state;
8  }

```

**STEP 8:** Now you need to create **index.js** in the **root** folder.

```

1  import 'babel-polyfill';
2  import React from 'react';
3  import ReactDOM from 'react-dom';
4  import {Provider} from 'react-redux';
5  import {createStore, applyMiddleware} from 'redux';
6  import thunk from 'redux-thunk';
7  import promise from 'redux-promise';
8  import createLogger from 'redux-logger';
9  import allReducers from './reducers';
10 import App from './components/App';
11
12 const logger = createLogger();
13 const store = createStore(
14   allReducers,
15   applyMiddleware(thunk, promise, logger)
16 );
17
18 ReactDOM.render(
19   <Provider store={store}>
20     <App />
21   </Provider>,
22   document.getElementById('root')
23 );

```

**STEP 9:** Now that you are done writing the codes, launch your application at **localhost:3000**.

GET TRAINED BY INDUSTRY  
EXPERTS

This brings us to the end of the blog on React Redux tutorial. I hope through this React Redux tutorial blog I was able to clearly explain what is Redux, its components and why we use it with React. You can refer to this blog on **ReactJS Tutorial**, in case you want to learn more about React.

*If you want to get trained in React and wish to develop interesting UI's on your own, then check out the **ReactJS with Redux Certification Training***

*by Edureka, a trusted online learning company with a network of more than 250,000 satisfied learners spread across the globe.*

more about React.

*If you want to get trained in React and wish to develop interesting UI's on your own, then check out the **ReactJS with Redux Certification Training***

*by Edureka, a trusted online learning company with a network of more than 250,000 satisfied learners spread across the globe.*

*Got a question for us? Please mention it in the comments section and we will get back to you.*



**About Swatee Chand (13 Posts)**

)

Research Analyst at Edureka. A techno freak who likes to explore different technologies. Likes to follow the technology trends in market and write about them.

Share on

PREVIOUS

NEXT