What is React-redux

React Redux is the official Redux UI binding library for React. If you're using Redux and React together, you should also use React-Redux to bind these two libraries.

As the official Redux binding for React, React-Redux is kept up-to-date with any API changes from either library to ensure that your React components behave as expected. Its intended usage adopts the design principles of React - writing declarative components. The followings are some benefits of using React-Redux library

- **It Encourages Good React Architecture:** As a general architectural principle, we want to keep our own components "unaware" of Redux. They should simply receive data and functions as props, just like any other React component. This ultimately makes it easier to test and reuse your own components.
- **It Implements Performance Optimizations:** React Redux implements many performance optimizations internally, so that your own component only re-renders when it actually needs to.
- **Community Support:** As the official binding library for React and Redux, React Redux has a large community of users. This makes it easier to ask for help, learn about best practices, use libraries that build on top of React Redux, and reuse your knowledge across different applications.

Redux vs react-redux

Redux gives you a store, and lets you keep states in it and extract them out. It responds when the state changes. but that’s all it does.

It’s actually **react-redux** that lets you connect pieces of the state to React components.

That’s right: **redux** knows nothing about React at all.

These libraries are like two peas in a pod, though. 99.999% of the time, when anyone mentions “Redux” in the context of React, they are referring to both of these libraries in tandem. So that's what you should keep in mind when you see Redux mentioned on StackOverflow, Reddit or any place else.

The **redux** library can be used outside of a React app too. It'll work with Vue, Angular, and even backend Node/Express apps.

Installing React-Redux

To start using Redux along with React, only you need to do is following these steps:

1. Create a new react application

```
$ npx create-react-app react-redux-counter
```

2. Install redux

```
$ npm i redux
```

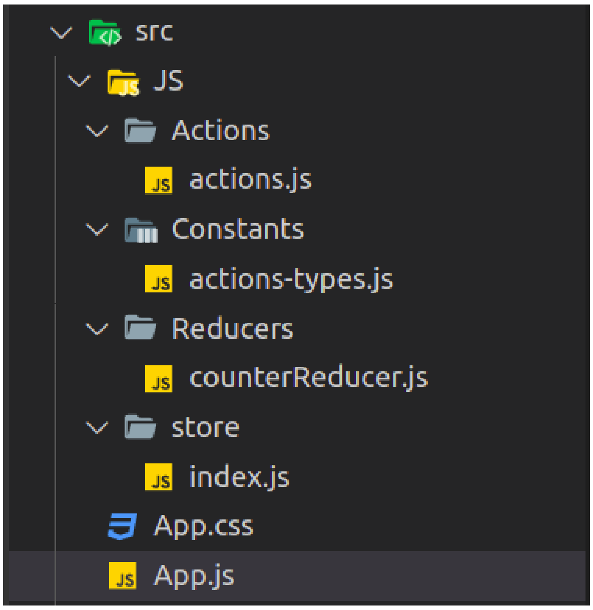
3. install react-redux

```
$ npm i react-redux
```

Redux folder architecture

Redux can make your life as a developer much easier or much harder depending on the architecture that you follow.

Searching the web you can find a lot of architecture or different methods to implement redux into your react application. The following is the one that we recommend using it.



```
src
  JS
    Actions
      actions.js
    Constants
      actions-types.js
    Reducers
      counterReducer.js
    store
      index.js
  App.css
  App.js
```

This architecture is simple and organized. You have to create a folder called JS (or any name that you prefer)

Under the JS folder, there will be four subfolders:

-> **Actions**: This folder will contain all the actions

-> **Constants**: This folder will include all the actions-creator

-> **Reducers**: it will hold the reducers

-> **Store**: it will contain the store creation.

‹ Previous next ›