

React-Redux for front end applications

Vedashri R Mungekar

Computer Science, MET– Institute of Computer Science
Mumbai, India

Abstract: With web technology rapidly expanding, there is a significant trend that Hypertext Markup Language (HTML) turns into a worldwide web consortium and leads the front-end development to stand on the front stage of internet history. However, there are numerous front-end development frameworks and libraries such as React, Angular and Vue. This paper starts with introducing an overview of the leading framework and library in the field of front-end development i.e. REACT. This paper will list the pros of REACT integration with REDUX which leads to development of front-end applications by taking control over data flow. Finally, the paper summarizes the advantages of REDUX.

Keywords: Redux, React.

1. INTRODUCTION

With the increase in the technology worldwide, many organizations have come forward to utilize efficient and faster UI frameworks. Redux is a library that can be used with any UI framework, including React, Angular, Vue and vanilla JS. Even though Redux works with all frameworks, Redux and React are commonly used together, they are independent of each other. When we are using Redux with any other form of UI framework, we will normally use a "UI binding" library or a connecting library to tie Redux together with our UI framework, instead of directly interacting with the store from your UI code. React Redux is the original UI binding library for React.

Front-end Frameworks			
Angular1	Angular2	React	Vue1
1.4 million	2.6 million	9.2 million	1.5 million

Table 1: Github front-end frameworks usage statistics in May 2018

Redux itself is a standalone library that can be used with any UI layer or framework, including React, Angular, Vue, Ember, and vanilla JS. Although Redux and React are

commonly used together, they are independent of each other. If you are using Redux with any kind of UI framework, you will normally use a "UI binding" library to tie Redux together with your UI framework, rather than directly interacting with the store from your UI code. **React Redux is the official Redux UI binding library for React.** If you are using Redux and React together, you should also use React Redux to bind these two libraries. To understand why you should use React Redux, it may help to understand what a "UI binding library" does.

2. OBJECTIVE

This paper describes that REACT is the leading front-end development library in developing web applications and list REDUX as a potential solution for developing a web application to make it more manageable and consistent to be run in different environments. By analyzing the comparison between REACT and REDUX.

Shift towards React Redux:

As the requirements for JavaScript single-page applications have become increasingly complicated, our code must manage more state than ever before. This state can include server responses and cached data, as well as locally created data that has not yet been persisted to the server. UI state is also increasing in complexity, as we need to manage active routes, selected tabs, spinners, pagination controls, and so on. As if this weren't bad enough, consider the new requirements becoming common in front-end product development. As developers, we are expected to handle optimistic updates, server-side rendering, fetching data before performing route transitions, and so on. We find ourselves trying to manage a complexity that we have never had to deal with before. Libraries like React attempt to solve this problem in the view layer by removing both asynchrony and direct DOM manipulation. However, managing the state of your data is left up to you. This is where Redux enters.

Requisite for React Redux Application:

- ***Single source of truth :***

The global state of your application is stored in an object tree within a single store. This makes it easy to create universal apps, as the state from your server can be serialized and hydrated into the client with no extra coding effort. A single state tree also makes it easier to debug or inspect an application; it also enables you to persist your app's state in development, for a faster development cycle. Some functionality which has been traditionally difficult to implement - Undo/Redo, for example - can suddenly become trivial to implement, if all of your state is stored in a single tree.

- ***State is read-only:***

The only way to change the state is to emit an action, an object describing what happened. This ensures that neither the views nor the network callbacks will ever write directly to the state. Instead, they express an intent to transform the state. Because all changes are centralized and happen one by one in a strict order, there are no subtle race

conditions to watch out for. As actions are just plain objects, they can be logged, serialized, stored, and later replayed for debugging or testing purposes.

- **Changes are made with pure functions**

To specify how the state tree is transformed by actions, you write pure reducers. Reducers are just pure functions that take the previous state and an action, and return the next state. Remember to return new state objects, instead of mutating the previous state. You can start with a single reducer, and as your app grows, split it off into smaller reducers that manage specific parts of the state tree. Because reducers are just functions, you can control the order in which they are called, pass additional data, or even make reusable reducers for common tasks such as pagination.

3. PROBLEM STATEMENT

Integration of REDUX with REACT leads to development of front-end applications with proper control over dataflow.

4. RESULT

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

React boasts excellent performance due to its hierarchical approach to data storage and rendering web elements. Unfortunately, this very benefit makes data management complicated and can quickly lead to code bloat.

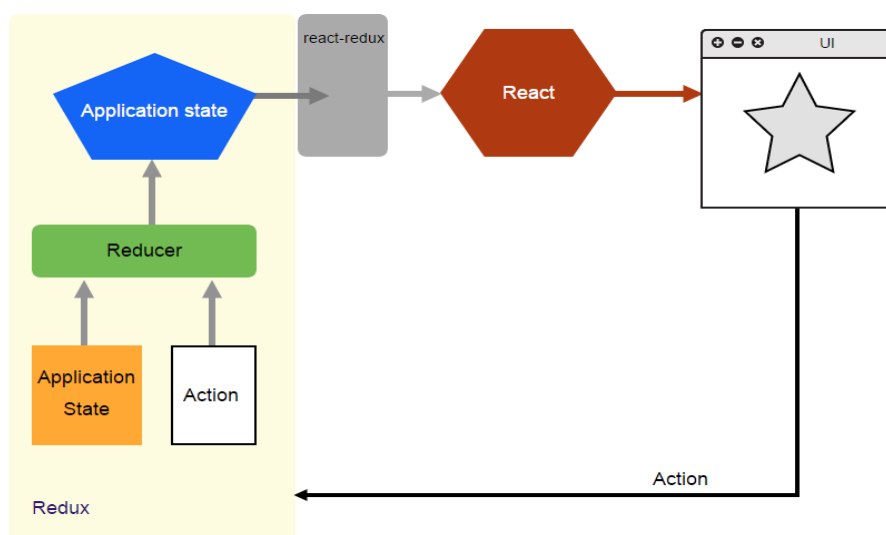


Fig.1 React-Redux architecture

This is where Redux can help. Redux manages the data between components by existing separately from the React hierarchy. Redux's architecture is built around unidirectional data flow, which pairs

nicely with React's rendering pattern. Since the data flow is one direction, we don't have to worry about side effects and can trust that the component will render or re-render in a predictable, React way.

5. RELATED RESEARCH

Web technology provides one means to develop apps that span platforms. To ensure that the comparison and evaluation of the frameworks were as objective as possible, only frameworks leveraging JavaScript as their main programming language were chosen. Due to the possibility of underlying differences in frameworks using other languages, those using anything, but JavaScript were excluded. Three frameworks for cross-platform development were chosen due to their novelty, recentness, and (perceived) developer interest. All of these frameworks have its own advantages and disadvantages, and selecting a particular framework is up to the user and their particular need for the application. The rapid modernization of new technologies ,has led to increase in generation of data in UI, hence multiple companies are working towards implementing React-Redux.

Reasons:

- Redux is mainly used where huge amount of data is generated .Best practice for managing application state.
- Redux is also helpful in state interpretation .
- Incase of scalable applications . the performance of system is not affected with increase in data.

6. CONCLUSION

The presented study describes how normally with React, you manage state at a component level, and pass state around via props. With Redux, the entire state of your application is managed in one immutable object. Every update to the Redux state results in a copy of sections of the state, plus the new change. At the end of this survey we can clearly see the advantages of using react redux for certain circumstances. Management of states and props is also in the most efficient way in terms of data rate and response time.

REFERENCES

- [1] Vipul Kaushik, Kamal Gupta, Deepali Gupta(2018). *React Native Application Development*, 10.1007/978-981-10-6620-7_59.
- [2] Tim A. Majchrzak, *Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks*.
- [3] Blogs : <https://react-redux.js.org/introduction/why-use-react-redux>
- [4] Research Paper: <http://www.ijcrt.org/papers/IJCRT2004607.pdf>
- [5] Redux Book: <http://thereduxjsbooks.com/>
- [6] Blogs on React-Redux:<https://www.telerik.com/blogs/control-your-frontend-data-before-it-controls-you-when-to-implement-redux-in-your-react-app>
- [7] Research Paper: <https://www.jstor.org/stable/815280>