Review on React-Redux Development Framework

Rozer Kshetrimayum
Student, B.Tech, Department of Computer Science and Engineering
Dronacharya College of Engineering, Gurugram, Haryana, India

Abstract: *With Redux, you can create dynamic data flow graphs. As a program executes, these graphs record every value-producing operation that occurs, building up an accompanying computational history of every value produced. The generated framework is compatible with Windows and iOS machines and produces satisfactory results for users on both platforms (IOS and Android). JavaScript ES6 has been used to implement the proposed methodology.*

Keywords: Redux

I. INTRODUCTION

Redux is a predictable state container for JavaScript apps.

Web development is often a tough job for a web developer because the developer has to make two applications, one for Android platform and one for iOS platform, which require knowledge of two different domains, i.e. for Android, knowledge of different browser interpreters and libraries (programming language: Kotlin or JavaScript) and for iOS mobile applications (programming language: Swift or JavaScript and IDE: Xcode). Redux will manage the state in all these platforms using ReactJS for browsers, React Native for iOS, and React Native for Android.

Redux is a dependency that manages state in react JS in order to implement complex applications. The research paper tends to achieve the following objectives:

- To discuss the architecture of how Redux work.
- To understand various component of Redux to be in react framework.

II. ARCHITECTURE OF REACT-REDUX

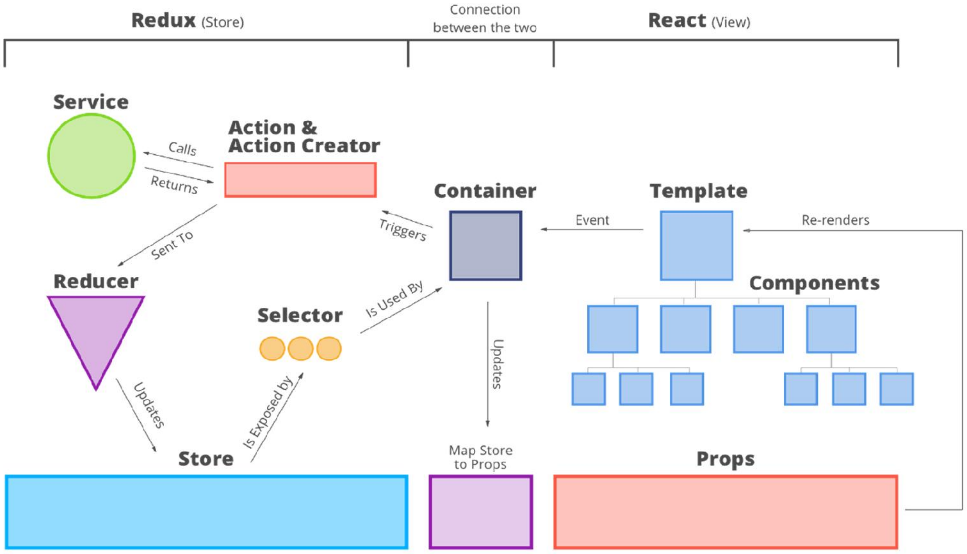

Figure 1: Architecture of React -Redux.

DOI: 10.48175/568

The React -Redux architecture and layout, how each connection is made, is described in the figure 1.

From the Above figure,

Services, Actions & Action Creators, Reducers, and Selectors are the component of Redux.

Templates and Components are the component of React.

Containers allow React and Redux to be loosely coupled. The store and the view can be updated independently, which is really important because it allows them to grow and change as they see fit.

React displays data through components from Redux's store, which maps the data to props. All of that is state.

III. PARTS AND CONNECTION

- **Component:** The component is rendered using a set of props that are passed down from their parent components. Here, the template is a set of code that uses properties and can call other components.
- **Template**: It is simply a component, we can say it as a special component that does think more specifically than the lower component in the figure 1.
- **Container**: The container connects React and Redux, allowing store and the view to be changed and grown independently of each other. The container also perform the generation of reports.
- **Actions and Action Creator**: This is used to implement the Redux services in the React framework.
  - **Action:** This is an object that contains the type of action and the state that was changed because of the action.
  - **Action Creator:** This is the code that is called to create an action and send it along to the reducer.
- **Reducer**: Whenever a dispatch is made, every reducer is called and given the dispatched action. Whether it is passed or handled is now up to the reducer. In simple words, this creates a mapping between the source (Redux) and the target (React).
- **Selector**: Data is passed from the store into the container and from there to React states through this function. It simply creates workflow and manage the execution.

REFERENCES

[1]. https://cyberleninka.org/article/n/412164
[2]. https://irejournals.com/paper-details/1702778
[3]. https://www.ijtsrd.com/papers/ijtsrd42490.pdf
[4]. https://www.theseus.fi/bitstream/handle/10024/130495/FInal_Year_Thesis.pdf?sequence=1&isAllowed=y
[5]. https://redux.js.org/