

# Importance of React-Redux in Front-End Development Framework

Anuj Kesarwani

1DS18MCA07

*Master of Computer Application*

*Dayananda Sagar College of Engineering, Bengaluru*

**Abstract:** In this research paper, I am going to mention the advantages of JavaScript based front-end library. As we know that in market there are plenty of front-end libraries available. In which React is one among the foremost popular front-end library. It mainly focuses on the MVC pattern and is being used widely for big scale application development. Wherein Redux is also a JavaScript library. Which is used to manage the state management throughout the application. Redux can be used with any other JavaScript framework or library.

## I-INTRODUCTION:

Developing the web-application is a very difficult task for a developer. According to the developers, it is very important to develop one application for the IOS based mobile and one for android based mobile. In which developers should know the domain of both the applications. For the android app, we should know the Java or Kotlin and for the IOS based mobile, we should know the Swift or Objective-c. Apart from these many hybrid web-application frameworks were developed by the developers and it was not successful to create the same experience as native platforms. To find out the solution of this problem, this paper will be showing the importance of React and Redux framework. Which will be helpful for the developers to build the hybrid web application for the cross platforms. To implement complex application, it uses the Redux library which is used to manage the state throughout the application.

## i. Proposed Application

This type of application is used to be developed by the React and Redux framework. Which can be run on both IOS and Android mobiles. In this type of application, it will store the data in backend as compared to other applications. But by using Redux, we can have data on the client side or front-end side and in the same way we can access it.

## ii. Why React and Redux

We know that our IT industry is growing very fast in terms of Technology and each IT company wants to use the best and fastest User Interface Frameworks. For example, React, Angular, Vue. These are the very popular frameworks for user interface, and now a days React is very popular in the market. Where Redux is the library, which is used to maintain the state in the application.

## iii. Market Trends of the framework

- It allows us to read the data by using the Redux store and it does dispatch the action. Developer can manage the states by using the use Selector () hooks.
- Redux permit “time-travel debugging” experience.
- Due to tightly coupled with root component, we cannot use the component again. Redux can be

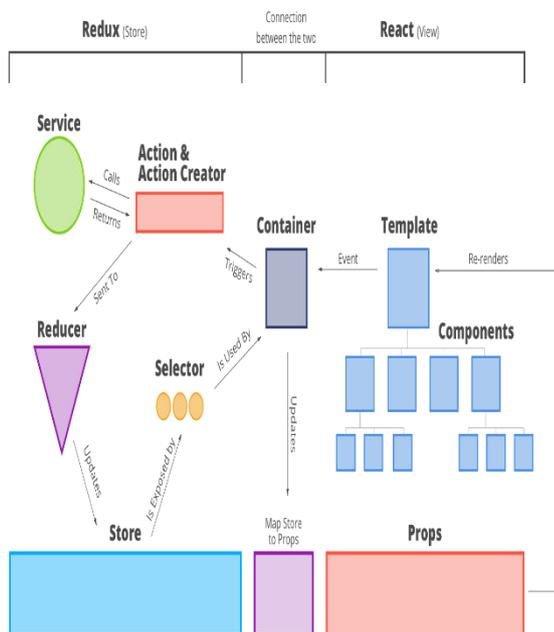
helpful to reduce this complexity and provides global accessibility.

## II. PROBLEM STATEMENT

When we are using the small web-app, mobile-app, in that case managing the state is quite easy, but if we are going to develop a large-scale application, in that case managing the state throughout the application is very hard. Because sometimes we need the data from another component, so every time hitting the backend is not the good practice. So, Redux can be helpful out from this problem, it will store the backend data in the reducer and we can easily fetch it, instead of hitting the backend the server.

## III STRUCTURE OF THE REACT-REDUX BASED APPLICATION

It contains the store, dispatch, selector, components and user interface.



As we can see from the above figure, it has following components-

**Component:** It contains the class and which is written in the JavaScript. It has props, state for

performing the several operations such as Taking input, showing messages etc. It also defines the layout of the UI. In the react we can pass the data from one component to another component, which can be handle using the props. React does support High Order Component. It means we can reuse the single component anywhere in application. This is the best feature of the react.

**Template:** This is the predefined component. Which can help developers to boost up the development process. It allows the developer to use or modify the template according to the requirements. For example, if we want to have a table in our application and which should show row, column, heading and pagination. If we are going to create the code for the Data table with pagination functionality, which will take much time and it will not be able to boost up the development process. Instead of creating from scratch, we can use the Data table component from the react library, which allows to modify the UI according to our requirements.

**Container:** This is used to connect with the redux store. It also gets the state and dispatch it to action. There are two components in react Presentational Component and Container Component. Where presentational Component contains the layout of the application which can be seen by the client side and container component is the main component of the react application. It is the root of the application.

**Actions & Action Creators:** Action, which contains the payload and sends by the dispatch to the reducer. It will be executing while calling the APIs because it will execute when any event will be called such as button click or focus on textbox. Whenever we called the post method then that time, we do need to pass the body. Action contains the body as the payload and send it to the reducer.

**Reducer:**It does accept the payload from the action and send back the payload to the services. It is crucial component of the redux because this component will decide whether it should the payload or not. There are separate reducer for each action which will be called by the dispatch.

**Selector:**As per the name we can understand that it will select the data from the store within container. It store the state which is coming from the reducer. It can boost up the performance of the Application.

**Store:** It is use to hold the state of the application throughout the application. It's not come under any category. it is just an object.

#### IV REDUX BASED APPLICATION

Due to many advantages of this framework, there are several organisations which are using this framework for the development.

##### i. Huge front-end data:

Redux is especially used where an inexpensive amount of knowledge gets changed over time. If the info states aren't changing frequently then Redux usage is restricted. it's used as a best practice for managing application state.

##### ii. State interpretation for complex applications:

There are some front-end technologies like React having their own application state management, so while using these libraries one should learn its inbuilt capabilities. Sometimes after developing the appliance, it becomes complex to know and code, it's hard to understand how the state has been changed. for this scenario, Redux is useful and used.

##### iii. Scalable applications:

Some applications can scale in size and when redux is employed we will simply add values to redux to proportion or down. The performance of the system isn't affected with the rise in data.

#### V FURTHER ENHANCEMENT

In upcoming time, redux is going to very strong framework in the development because with a normal redux store we can do only synchronous updates by dispatching an action. But in future we work on large project then we have to use the Asynchronous updates, which is possible by using the middleware libraries such as Thunk and Saga. These are the middleware libraries which can be use with the redux for asynchronous updates to front-end UI.

#### VII CONCLUSIONS

Due to many advantages of this framework, Redux is widely being used in various organisation for the development and Its very suitable for both Small and Large Application. Redux support both Synchronous and Asynchronous updates, so this is best thing in the redux, which is like by all the front-end developers. If we talk about benefits of this framework such as: Easy Debugging, Code for the testing is very simple, Easy to maintain the data rate and performance rate and server rendering.

**REFERENCES:**

[1] Redux from Tutorials Point

<https://www.tutorialspoint.com/redux/index.htm>

[2] React from W3Schools

<https://www.w3schools.com/react/>

[3] Redux from Redux Official

<https://redux.js.org/>

[4] React JS from React Official

<https://reactjs.org/>

[6] Core Concept of Redux

<https://redux.js.org/introduction/core-concepts>

[7] React Redux Release

<https://github.com/reduxjs/redux/releases>