

MobX vs Redux: Choosing the Best State Management Library For Your Next React Project

TechRounder PDF Edition

Live article:

<https://www.techrounder.com/development/mobx-vs-redux-choosing-the-best-state-management-library-for-your-next-react-project/>

By Vipin PG | Published March 29, 2023 | Updated January 4, 2026 | Format: Comparison | 3 min read

Bottom line

State management is the biggest challenge that React developers face in their applications. This process involves handling the different components involved in developing a complex and efficient user experience.

State management is the biggest challenge that React developers face in their applications. This process involves handling the different components involved in developing a complex and efficient user experience.

Some developers use hooks, which help them access and share various state components. However, they can't handle the complexity of dealing with many of them. That is why, to solve the problem, they should use state management libraries.

Fortunately, many different state management solutions and frameworks are becoming more popular. But between MobX vs. Redux, which external library gives the best framework for managing the state of an application? Let's find out!

What Exactly is a MobX?

The state management of the MobX system helps app developers to manage their local state.

The framework of MobX is a popular yet very simple state management tool that can be used to manage the different elements of an application.

MobX was first released in the year 2015 and is written with JavaScript. Its main features are Derivations, States, and Actions. These features allow the application to react to events and modify the state.

The React and MobX combination provides an excellent output, which can render the app state into components.

Furthermore, with MobX, you can also update and store the application state used in React, which makes it easy to manage the local state of your application.

Fundamental Principles of MobX

- The state of the application in MobX is stored in one object tree.
- Whenever the state changes, the derivatives from the state are updated automatically
- Different types of stores can hold the state of an application.
- It can easily change the code state of an application in a way that is simple to test without a browser.

The framework of MobX provides a flexible yet simple way to build applications that are easy to maintain and implement. Moreover, its code is decoupled and is available for testing.

Moreover, it can automatically store various types of data in its stores. This makes it ideal for developers looking to update their applications' state continuously.

What Exactly is a Redux?

With the help of Redux, many developers can quickly develop an application that performs well in various environments. It also features tools like debuggers, live code editing, etc.

Redux is a great state management solution combining functional programming and Flux capabilities.

React introduced Redux in 2015, a framework that provides state management for any application. On the other hand, it is also a combination of the Elm language's functional programming and the architecture's Flux design.

Fundamental Principles of Redux

- Unchangeable state in the store.
- State updates reducers.
- It has a penetrative one-store with complete details.
- Actions can also be taken to ask for alterations to the store.

Furthermore, the most critical factor a developer needs to consider when developing applications is clear coding patterns.

Also, three main components are involved in the development of a store: the Event handler, the UI, and the Reducers. The UI takes action from the user while the Event handler retrieves the state from the store.

The software Redux manages different states in a way that allows them to keep changing. It can also develop a user interface for frameworks such as React and Angular, offering good support for multi-platform development.

MobX or Redux: Which is the Best Option?

With the help of MobX, you can develop multiple stores. These stores can then be separated to ensure the application's many states stay separate.

For instance, most applications have at least two stores, one for the UI and one for the domain. Moreover, the advantage of this approach is that it allows you to reuse the domain in different applications.

Redux, however, only has a singular one-store repository. Its immutable nature makes locating the state or data we need easy.

So, since there is only one object for the repository, developers can easily branch out the code into different reducers to separate the various concerns related to the repository.

Final Verdict

If you are asking about MobX vs. redux, Redux is far better than MobX because of its scalability and popularity. However, if you want to build simple apps with minimal code, the better alternative is MobX.

Furthermore, when you need to manage an application's state, you can always go with MobX. Meanwhile, if it's a basic application that needs to be fast delivered, you can choose Redux. So there are a lot of advantages to choosing either of these platforms, and it's a good deal.

References

1. redwerk.com - blog / mobx-vs-redux-choosing-the-right-state-management-solution - <https://redwerk.com/blog/mobx-vs-redux-choosing-the-right-state-management-solution/>
2. simform.com - blog / angular-vs-react - <https://www.simform.com/blog/angular-vs-react>