

Why use Redux?

A few reasons for using redux in your applications.

As you already know, questions like, why should you use A over B boils down to your personal preferences.

That being said, there are still some very decent reasons to learn Redux.

For example, in larger apps with a lot of moving pieces, state management becomes a huge concern and Redux ticks that off quite well without performance concerns or trading off testability.

One other reason a lot of developers love Redux is the developer experience that comes with it. A lot of other tools have begun to do similar things, but big credits to Redux.

Some of the nice things you get with using Redux includes logging, hot reloading, time travel, universal apps, record and replay - all without doing so much on your end as the developer. These things will likely sound fancy, but wait till you use them and see for yourself.

If you don't consider yourself a senior engineer, I advise you to learn Redux - largely because of some of the principles it imbibes. You'll learn new ways of doing common things, and this will likely make you a better engineer.

Everyone has different reasons for picking up different technologies. In the end, the call is yours, but it definitely doesn't hurt to add Redux to your skillset.