

2.2 What problem do we solve?

Redux is getting becoming the de facto way to build React apps. But should Redux be used in all React apps? At the very least, not all apps will need an ambitious state management solution from the beginning.

Today's front-end development trends are **component-based**. Components can have data state and UI state, and the state that they need to manage becomes more and more complicated as your app grows.

State management solutions have emerged to solve the following problems, and Redux is becoming popular as a standard among other solutions.

- components share state
- state should be accessible from anywhere
- components need to mutate the state
- components need to mutate the state of other components

Redux is a **state management library**, which is a tool that allows you to store the state of our app somewhere, mutate the state, and receive the updated state.

In other words, with Redux, we have one place where we can refer the state, change the state, and get the updated state.

Redux was written with React in mind, but it is also **framework-agnostic** and even can be used with Angular or jQuery applications.

*I recommend you read **Dan Abramov's** [You Might Not Need Redux](#) before choosing Redux.*