

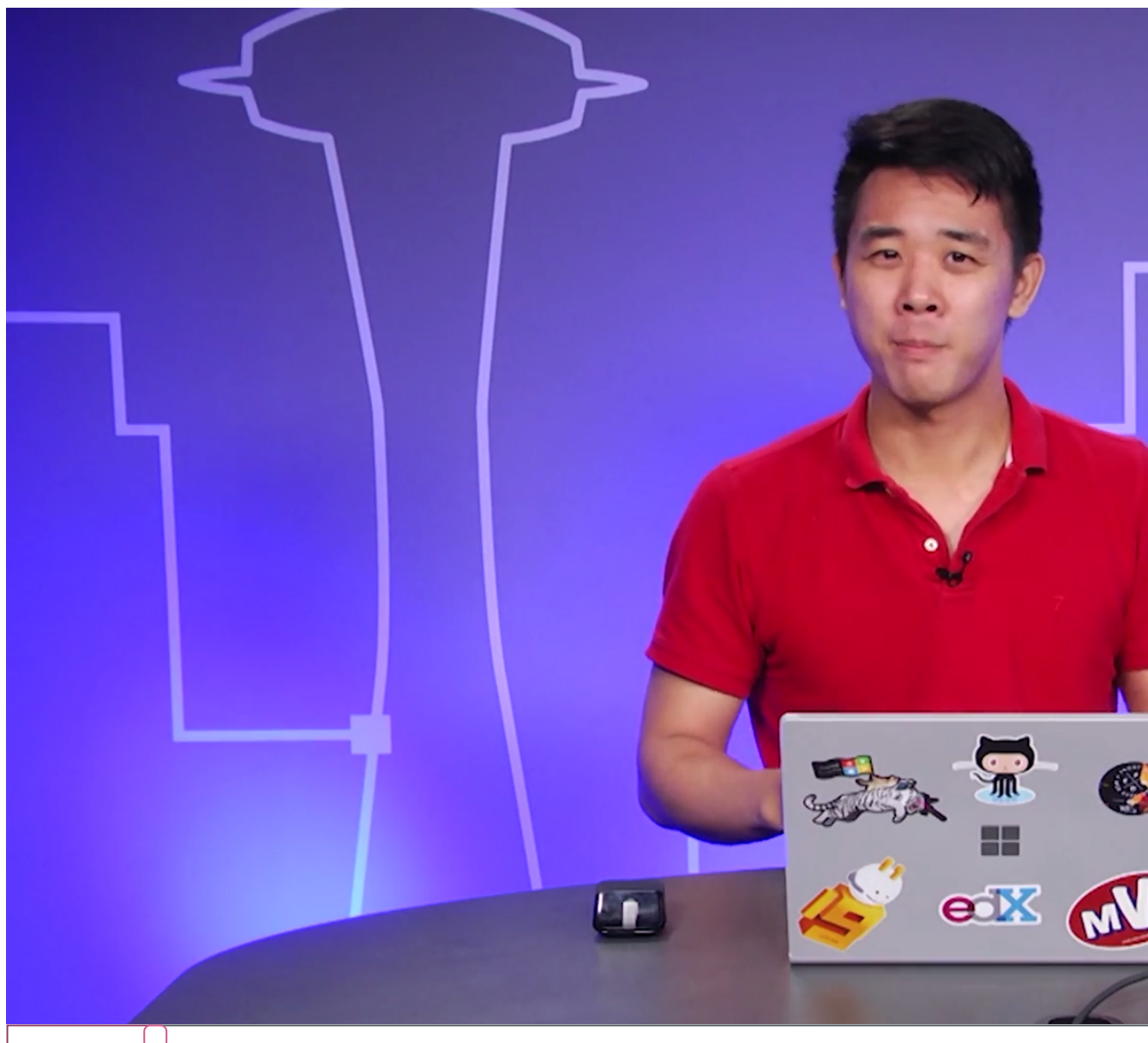


[Course](#) > [Redux](#) > [What is...](#) > What is...

Audit Access Expires Jun 20, 2020

You lose all access to this course, including your progress, on Jun 20, 2020.

What is Redux



0:24 / 3:24



1.50x



Video

[Download video file](#)

Transcripts

[Download SubRip \(.srt\) file](#)

[Download Text \(.txt\) file](#)

What is Redux

What is Redux

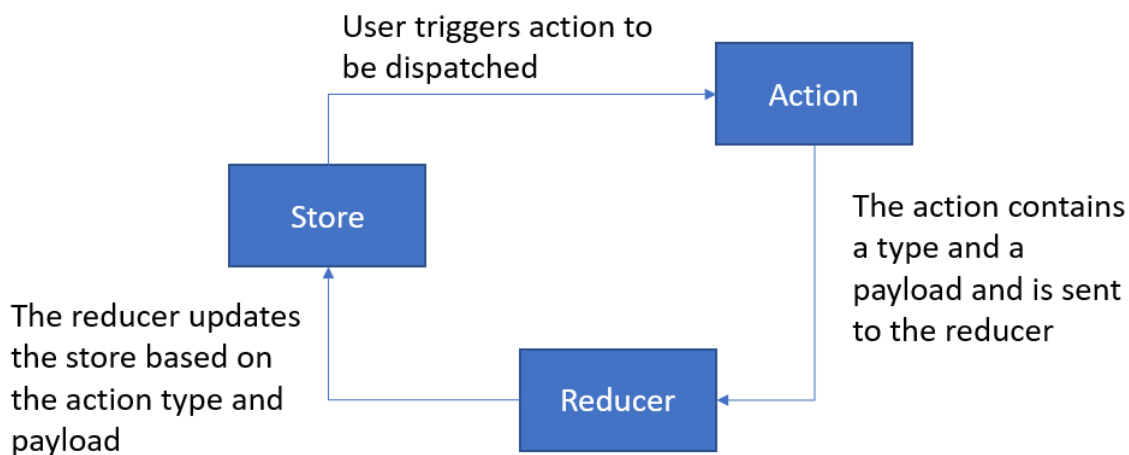
Redux is a third party state management library that makes it easier to access and modify your application state. Redux is often used with React, but it can be used with any other front end library as well.

Redux simplifies state management by using the following 3 concepts:

- Store - a single object that holds all your application state
- Actions - categorized events that trigger a change to your application state
- Reducers - functional definitions of how the application state should change

These concepts work together in the following way:

1. All application data is held inside a store
2. A user or timed event triggers an action to be dispatched.
3. The reducer figures out what type of action was dispatched and updates the store accordingly
4. The store now contains updated data.



Why should you use Redux with React?

Globally accessible state

One annoying thing about React is that state has to be passed down from parent component to child component in order for the child component to use it. This gets increasingly frustrating when you have many components nested within each other. Imagine passing state from a parent component down to through five components to get to the last child component. It would be a bit tedious to manually set the state as the

props of each of the child components. This is one of the problems that Redux solves. Redux has a single store that holds all of the application data that can be accessed by any component. Redux does this by providing plug in that allows you to pass in part of store data as props to any of your React components.

Globally accessible event handlers

One thing that is even more annoying than passing down state across multiple components is passing down event handlers, since they are a bit more complex. However, you no longer have to do that with Redux. Redux has a plug in that allows you to pass any dispatched `action` as a prop to your components.

Easier troubleshooting and organization

Redux makes it much easier to organize all your state management logic. Instead of having a bunch of helper functions spread out across multiple class components, all of your state management logic is kept inside a `reducer`. This also makes it much easier to troubleshoot. If something doesn't behave as it should, you can check what `action` caused it to go wrong. Once you figure out what `action` went wrong, simply check the appropriate section in the `reducer` and find the bug.

When should you use Redux and when should you not?

My advice is to not use Redux if you can accomplish your project with plain old React. If it gets too annoying to pass down state through parent components to child components, or if you have too many types of event handlers to keep track of, then it might be a good idea to use Redux. Redux was designed for complicated applications like Facebook or Twitter. If you are making a simple todo app then you probably don't need Redux.

Try out this live example on CodeSandbox: <https://codesandbox.io/s/1zn20o3y13>