General Data Loading with Redux

Component gets rendered onto the screen

Component's 'componentDidMount' lifecycle method gets called

We call action creator from 'componentDidMount'

Action creator runs code to make an API request

API responds with data

Action creator returns an 'action' with the fetched data on the 'payload' property

Some reducer sees the action, returns the data off the 'payload'

Because we generated some new state object, redux/react-redux cause our React app to be rerendered

Action Creators

fetchPosts

Action

type: 'FETCH_POSTS'

payload: response

Reducers

postsReducer

Maintains an array of all fetched posts

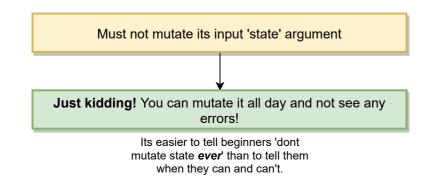
Rules of Reducers

Must return any value besides 'undefined'

Produces 'state', or data to be used inside of your app using only previous state and the action

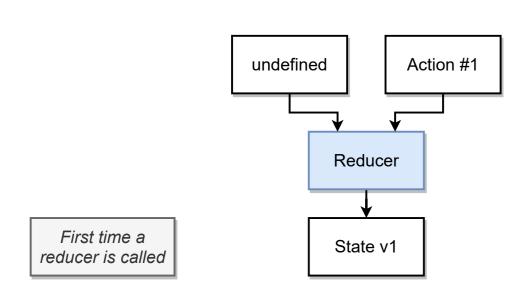
Must not return reach 'out of itself' to decide what value to return (reducers are pure)

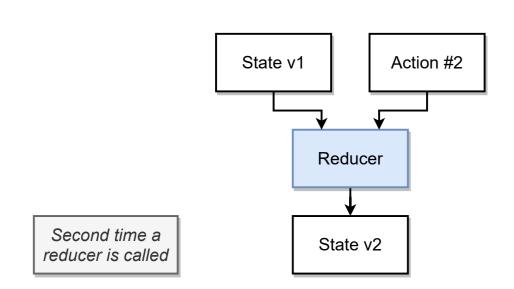
MISLEADING >> Must not mutate its input 'state' argument

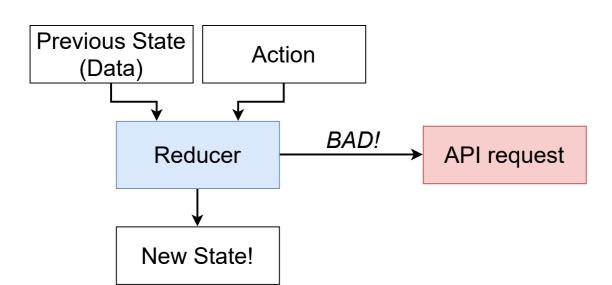


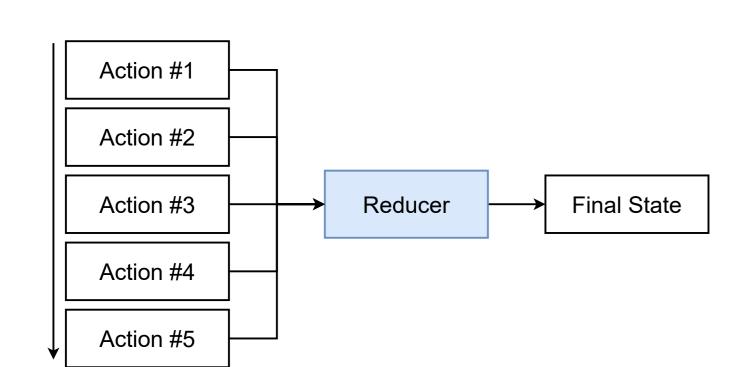
We are not going to mutate state ever

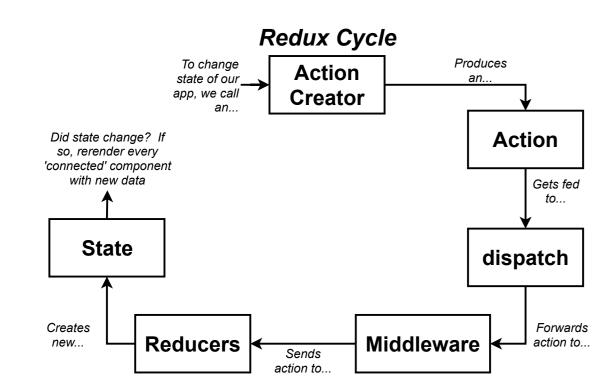
I want you to understand the behind the scenes of this rule to help you better understand Redux

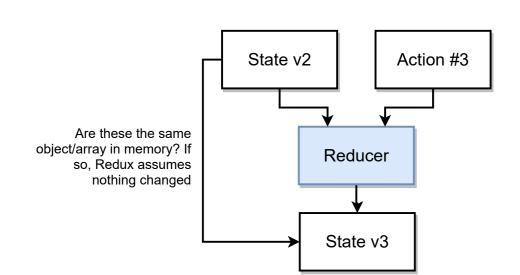




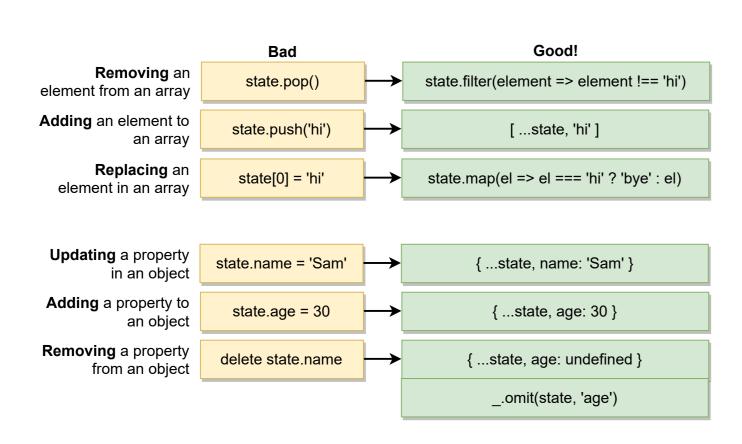


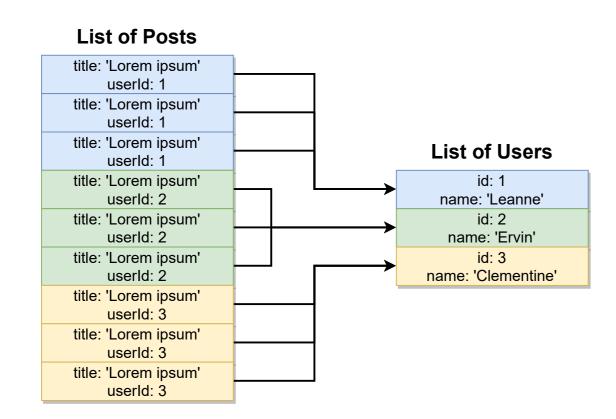


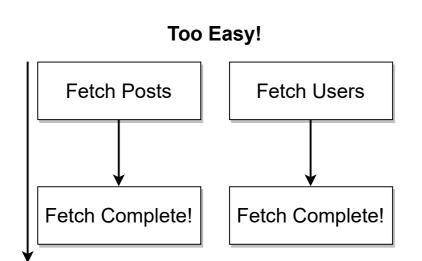


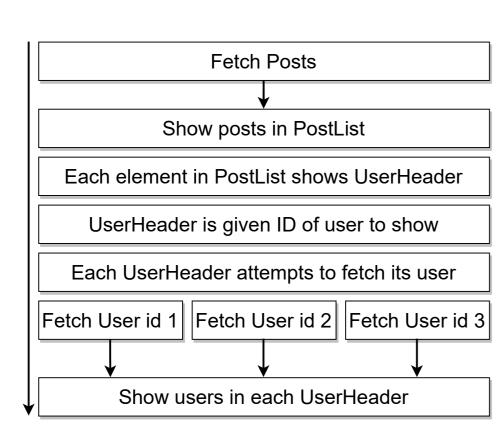


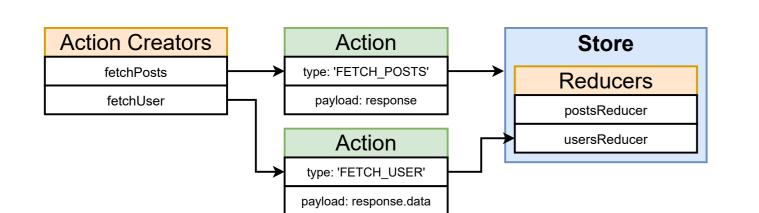
github.com/reduxjs/redux/











fetchPostsAndUsers()

Call 'fetchPosts'

Get list of posts

Find all unique userId's from list of posts

Iterate over unique userId's

Call 'fetchUser' with each userId