Școala informală de IT

React Redux

What is Redux?

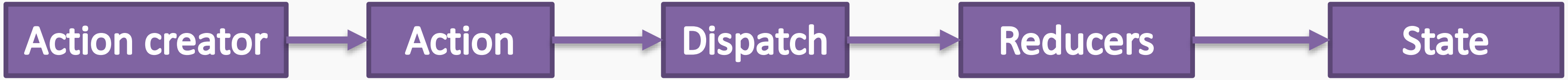
- A library for state management
- State management - a way sharing of data across components.
- Used in complex applications to make the state management easier
- The state of the app is added to a store
- Each component can access each state from the store

State

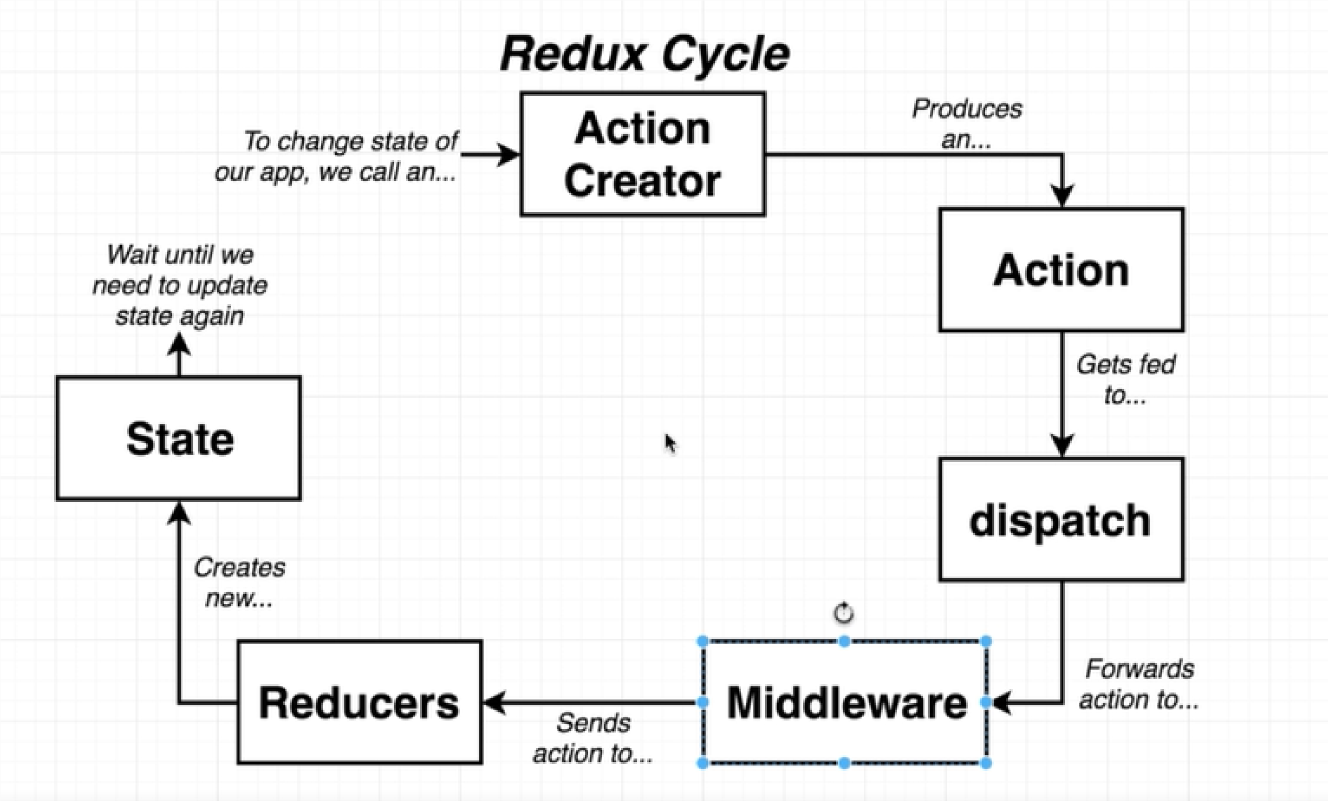
- **Local State**
  - State that belongs to a single component (e.g. user input)
- **Cross-component State**
  - State that affects multiple components (e.g. a button that opens a dialog – different component → props chain/prop drilling)
- **App State**
  - State that affects the entire app (e.g. user authentication → navigation is affected and some component will show more or less data depeding on user authorization)

How Redux works

- One central data(state) **store** for the entire application
- The data from the store could be used by any component
- The component subscribe to the store, and the store notifies component when the data is changed
- The states for store are NEVER directly manipulated by the components
- Reducer functions → the way to change the states from the store
- Components dispatch actions (simple js objects that describes the action that reducers should perform)
- Actions are forwarded to reducers, the reducers change the state from the store


Action creator
Action
Dispatch
Reducers
State

Redux cycle


Redux Cycle
To change state of our app, we call an...
Action Creator
Produces an...
Action
Gets fed to...
dispatch
Forwards action to...
Middleware
Sends action to...
Reducers
Creates new...
State
Wait until we need to update state again

React | Redux | React Redux

- Redux is not designed to work only with React
- React-Redux package connect the React app with Redux
- let’s create a new react app: “companies”
  - Install redux and react redux packages