Redux

WHAT IS REDUX?

Redux is, at its core, an incredibly simple pattern. It:

1. saves a current value
2. runs a single function to update that value, when needed
3. notifies any subscribers that something has changed

3 PRINCIPLES OF REDUX

I

REPRESENT THE WHOLE STATE (DATA AND UI STATE) OF YOUR APPLICATION IN A SINGLE JAVASCRIPT OBJECT.

a.k.a. the state, the state tree, the store (state + methods)

- The state is a minimal representation of the data in your app
- Single source of truth - simple data flow, UI can be pure function of state

II

THE STATE TREE IS READ ONLY - TO CHANGE IT YOU MUST DISPATCH AN ACTION

- an action: the minimal representation of a change to the state
- all mutations to the state are **explicit**, so you can keep track of them (debugging :))
- separation of concerns - components don't need to know how the state works/is structured, they just dispatch an action
- implementation detail: a requirement that every action's type property is not undefined

III

STATE MUTATIONS ARE PERFORMED BY PURE FUNCTIONS, CALLED REDUCERS

- the functions that change the state (reducers) must be pure - they take the current state and the action being dispatched, and return the next state (a new object)
- the next state is therefore a pure function of current state and the dispatched action
- why? enables time-travel debugging

REDUCER DETAILS

- If it receives an unrecognised action, returns the existing state
- If the state is undefined, returns an initialState
- optimisations

DISPATCHING ACTIONS - HOW DOES THIS WORK?

The dispatch function simply calls the reducer function on the action and current state, and saves whatever value it returns

mapDispatchToProps:

- a function that takes dispatch and returns functions that dispatch actions, i.e. it binds action creators with the dispatch function
- Shorthand - if the desired function takes the same arguments as the action creator, you can use the object shorthand instead of explicitly binding action creators with dispatch, that maps the names of the callback props to the corresponding (bound) action creators

THANKS!