

01/07

What is React redux



IBTISSAM
OULAHCHOUM

SWIPE





02/07

IMAGINE YOUR APP IS LIKE A GROUP OF FRIENDS PLANNING A VACATION. INSTEAD OF EVERYONE TALKING AT ONCE, THEY WRITE DOWN THEIR IDEAS (ACTIONS) AND GIVE THEM TO ONE ORGANIZED FRIEND (REDUX). THAT FRIEND LOOKS AT THE IDEAS, CHECKS THE RULES (REDUCERS), AND UPDATES THE PLAN (STATE). NOW, EVERYONE'S ON THE SAME PAGE, AND THE VACATION IS SMOOTH!



KEEP SWIPING



Why Use Redux?

- ✓ **CENTRALIZED STATE:** ALL YOUR APP'S DATA IS IN ONE PLACE,
- ✓ **PREDICTABLE:** CHANGES HAPPEN IN A CLEAR, TRACEABLE WAY.
- ✓ **DEBUGGABLE:** TRACK EVERY ACTION, MAKING BUG-FIXING SIMPLER.

ONE MORE

Key Concepts:

Store : Think of the store as the central brain of your application. It's where all the data (state) lives. Whether it's user information, app settings, or anything else, the store keeps it all in one place, making it easy to access and manage.

Actions: Actions are like sending a letter to the store. They're simple objects that describe what you want to do, like "ADD_TODO" or "DELETE_ITEM." Actions don't do the work themselves; they just say what needs to happen.

ONE MORE



Key Concepts:

Reducers: Reducers are the decision-makers. When an action arrives, the reducer looks at the current state and the action, then decides how to update the state. For example, if the action is “ADD_TODO,” the reducer adds that to-do to the list in the state.

Dispatch: Dispatch is the method used to send an action to the store. It’s like dropping your letter in the mailbox. Once you dispatch an action, the store uses the reducers to figure out how to update the state.

ONE MORE



When to Use Redux?



LARGE APPS: MANAGING LOTS OF COMPONENTS AND STATE.



COMPLEX STATE: WHEN YOUR STATE HAS MULTIPLE INTERDEPENDENT PARTS



PREDICTABILITY NEEDED: WHEN DEBUGGING OR CONSISTENCY IS CRUCIAL

ONE MORE



07/07

**Thank you for your
time! Keep coding,
keep growing!**

COMMENT BELOW



**IBTISSAM
OULAHCHOUM**