

#### Redux

### 

- What is Redux?
  - Definition: A JavaScript library for state management .
  - Why Redux?
    - Stores all application data in a single place.
    - State changes are controlled using actions and reducers.
    - Avoids unnecessary re-renders in React.
- What is the Core Principles of Redux?
  - 1- Single Source of Truth.
    - The entire application state is stored in a single object (store).
  - 2- State changes are controlled using actions and reducers.
    - State is read-only, meaning it cannot be modified directly.
    - updates happen through actions dispatched to reducers.
  - 3- Avoids unnecessary re-renders in React.
    - Reducers specify how the state changes in response to actions.

#### ■ (2). Definitions

- ✓ Reducer → A function that takes the current state and an action, then returns the updated state.
- ✓ Store 

  The place that holds the entire Redux state and allows access via useSelector, updates via dispatch(), and listens for changes.
- ✓ Subscribe → A method that registers a listener to be called whenever the state changes.

- ✓ Action → An object that describes a change in state, Come with a type and an optional payload.
- ✓ Payload 
  → The data or information sent with an action, typically used to update the state.

------

## **I** (3). Redux Workflow

(User Clicks Button)



dispatch(action)



Redux Store Receives Action



Reducer Updates the State



Store Saves the New State



Component Reads Updated State



(UI Updates)

- 1. Download Redux Toolkit
- 2. Create Slice (Reducer & Actions)
- 3. Configure Store
- 4. Set Up Provider
- 5. Redux Hooks (useSelector & useDispatch)

\_\_\_\_\_

# 

#### 1. Download Redux Toolkit

• npm install @reduxjs/toolkit react-redux

### 2. Create Slice (Reducer & Actions)

- What is createSlice?
  - is a higher order function
  - helps create reducers and actions easily
  - Takes:
    - name Of the slice.
    - initialState The starting state of the slice.
    - reducers Contains functions that modify the state.
  - Returns:
    - actions Generated action based on reducer names.
    - reducer A reducer function that will be used in store.

# 3. Configure Store

- What is Store?
  - The place where the entire application state is kept.
  - The store is created using configureStore

- Takes:
  - reducer Of the slice.
  - preloadedState (Optional) Load old data.
- 4. Set Up Provider
  - What is <Provider>?
    - wraps the app to give access to the Redux store.
    - <Provider store={store}>
- 5. Redux Hooks (useSelector & useDispatch)
  - What is useSelector?
    - its like getState(), return the state.
  - What is useDispatch?
    - its like setState(), update the state.

### 

• npm install @reduxjs/toolkit react-redux

------

## ■ (5). Auth Cycle

(User Enters Email & Password)



dispatch(loginUser(credentials))



Redux Store Receives Action



Reducer Handles Login



API Call to Validate Credentials



If Success → Store User Data in Redux



If Failure → Show Error Message



Component Reads User State & Updates UI



(User Clicks Logout)



dispatch(logoutUser())



User State is Cleared & UI Switches Back to Login