# Routing in React (React Router)

**1. What is React Router? How does it handle routing in single-page applications?**

Redux is a state management library for JavaScript applications, commonly used with React. It helps manage application state in a predictable way, making it easier to handle complex state interactions and debugging.

WHY USE :

entralized State Management – Redux stores the entire application state in a single store, making state management more structured and predictable.

Predictability – Since state changes occur through pure functions (reducers), it is easier to debug and test the application.

Efficient Updates – Components only re-render when necessary, improving performance.

Easier Debugging – Redux DevTools allow tracking of actions and state changes over time.

Better State Sharing – Useful for large applications where state needs to be shared across multiple components.

ACTION : ctions are plain JavaScript objects that describe what needs to be done.

They contain a type property (describing the action) and optionally a payload (data for updating the state).

REDUCERS : Reducers are pure functions that specify how the state should change in response to an action.

They take the current state and an action as arguments and return a new state.

STORE : The store holds the global state of the application.

It provides methods to dispatch actions, get the current state, and subscribe to changes.

**2. Explain the difference between BrowserRouter, Route, Link, and Switch components in React Router.**

Less Boilerplate – Recoil uses atoms (state units), no actions/reducers needed like Redux. Better Performance – Only components using specific atoms re-render, unlike Redux's global state updates. Easier Async Handling – Recoil handles async state (e.g., API calls) with selectors, no middleware needed. Simple Setup – Just wrap your app in , no complex store setup like Redux. Use Recoil for a lightweight, React-friendly approach. Use Redux for complex state logic and debugging tools.