

Introduction to Redux

Jogesh K. Muppala



THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系



香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

What is Redux?

- Predictable state container for JavaScript apps
- Inspired by Flux, Elm, Immutable
- Makes state mutations predictable

Main Principles of Redux

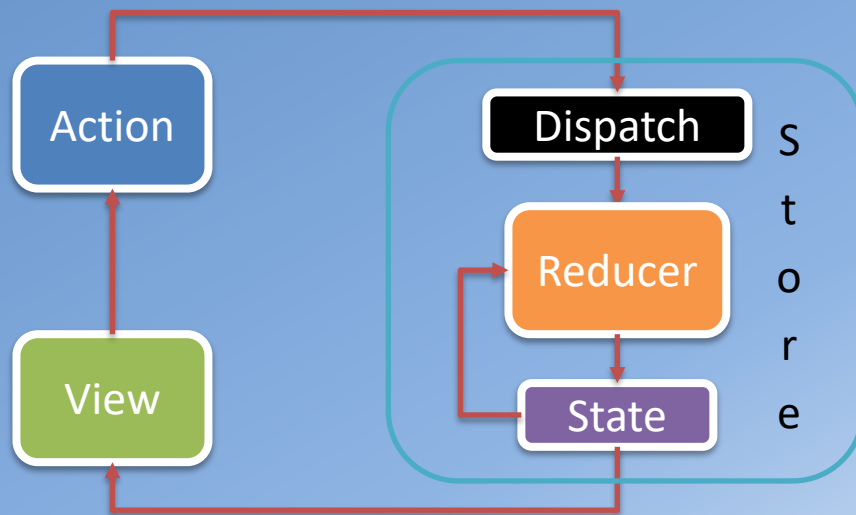
- Single source of truth
 - Single state object tree within a single store
- State is read-only (only getters, no setters)
 - Changes should only be done through actions
- Changes are made with pure functions
 - Take previous state and action and return next state
 - No mutation of the previous state

What is Redux

- Single store and single state tree enables powerful techniques:
 - Logging
 - API handling
 - Undo/redo
 - State persistence
 - “time-travel debugging”

Redux Data Flow

- Uni-directional data flow



Redux Concepts

- State: stored in plain JS object
- Action: plain JS object with a type field that specifies how to change something in the state
- Reducer: pure functions that take the current state and action and return a new state
 - Update data immutably (do not modify inputs)

Redux Store

- Holds the current state value
- Created using `createStore()`
- Supplies three methods:
 - `dispatch()`: states state update with the provided action object
 - `getState()`: returns the current stored state value
 - `subscribe()`: accepts a callback function that will be run every time an action is dispatched

React with Redux

- Use the react-redux package for bindings between React and Redux
 - connect(): generates a wrapper “container” component that subscribes to the store
 - Surround your App root with <Provider>
 - Takes the store as an attribute
 - Makes store accessible to all connected components

React with Redux

- The `connect()` function takes two optional arguments:
 - `mapStateToProps()`: called every time store state changes. Returns an object full of data with each field being a prop for the wrapped component
 - `mapDispatchToProps()`: receives the `dispatch()` method and should return an object full of functions that use `dispatch()`