UCSC Silicon Valley Extension

Web Application Development Using React, Redux, and Typescript

Dr. Min Wu
minwu.training@gmail.com

Libraries that We Need

- React and ReactDOM
- “ReactJS, as the V in MVC, allows you to build reusable UI components and makes maintaining changes in your data’s state effortless by abstracting the DOM.
- “Combined with a bundler utility like Webpack, ReactJS greatly simplifies building and maintaining SPAs.”

Libraries that We Need (cont.)

- React, ReactDOM
  - Component-based (easy to understand)
  - One-way data flow (easy to code and debug)
  - Virtual DOM (efficient)
- Redux, Redux-thunk, Redux-logger
- ImmutableJS
- Typescript

React and ReactDOM - Component-based

- React is an engine for building composable user interface using JavaScript and XML
- React supports component-oriented development using pure JavaScript.
    - In React, everything is made of components, which are self-contained, concern-specific building blocks.
    - Components are kept small. It is easy to create complex and more feature-rich components made of smaller components.
    - A separation of concerns in a component approach

React and ReactDOM - One-Way Data Flow

- Two-way data flow is messy.
- One-way data flow is clean and robust.
- React + Redux form a one-way data flow cycle.

React and ReactDOM - Virtual DOM

- React lets us write in a declarative way how components should look and behave.
- When data changes, React **conceptually** renders the whole interface again.
- React updates the in-memory, lightweight virtual DOM.
- React then compares the updated virtual DOM with the real DOM in the browser to decide which real DOM elements need to update.
    - This step is highly optimized by React library

Redux Principles

- Single source of truth: the state of your whole application is stored in an object tree within a single store.
  - Designing the state structure is the critical job
- State is read-only: the only way to change the application state is to emit an action object.
- Changes are made with pure functions: reducers are pure functions that take the current state and an action object, and return the next state.
  - Always return a new state object, instead of mutating the current state object.

Redux Middleware

- Redux-logger
  - Redux makes the state changes predictable and transparent.
  - Log and display every action that happens in the App, together with the computed state.
  - Easy to understand the action flow and debug
- Redux-thunk
  - Deal with async flow with server

Immutable JS

- Makes the application state immutable
  - Immutable data cannot be changed once created
  - List (ES6 Array), Stack, Map (ES6 Map), OrderedMap, Set (ES6 Set), OrderedSet, Record (JS Object)
- Immutable JS has been tuned for good performance.

TypeScript

- A strict superset of JavaScript
- Add static typing and class-based OOP to JavaScript
- Help code refactoring and debugging