Section Notes

Section 5 : Actions, Reducers, Redux

See the Prezi presentation (https://prezi.com/view/QAfXVHFWHPTz8OyGzM1W/)

Brief Notes

- Data flow
  - Unidirectional, Components -> Actions -> Reducers -> Store -> Container -> Component
  - Use this flow to your advantage
- Get used to installing boilerplate that sets up the wiring you need and the folder structure outlined in the videos
- Actions
  - Do AJAX calls or send information that should be shared across whole app
  - Can put some logic in here or in a Container with mapDispatchToProps() (next section)
  - Avoid too much logic here if possible, keep it minimal
  - Actions fired from components, and wrapped in a dispatch function
- Reducer
  - Need to create individual reducers for combineReducer() function
  - Contains state variable relative to that reducer, which is connected to the global state
  - Is interlinked with the Store
- Container
  - Wraps the component and manages component props
  - Can contain dispatch() functions that fire actions

Review Assignment

- Fix some bugs in a set of boilerplate actions and reducers
- See solution code for the solutions

Quiz

See page 2 for answers.

1. What does a reducer do?
2. You should avoid putting logic code into reducers. True or false?
3. Where do AJAX calls go?
4. Containers handle rendering and component state. True or false?
5. Redux is the only library that works with React data flow.

Quiz Answers

1. It’s complicated. But on a simple level, it listens for the action and receives information from an action,
2. True, if it’s anything that doesn’t directly have to do with taking care of the global state.
3. Almost always in Actions.
4. False. Containers do have a role in rendering, but most of that work (and internal state) is done within a component.
5. False — Flux and many others also exist and are used in many projects.