React & Redux

Link: - [GitHub](https://github.com/jrzerr/react-redux-tsheets-app) [https://github.com/jrzerr/react-redux-tsheets-app]

# What is React?

* Within the MVC design pattern, it is for building your View, but also to a lesser degree, your Controller
* Efficient DOM updating when data changes by using a Virtual DOM to compute differences before changing the actual DOM
* Components use one way data flow, data flow is explicit
* Leads to using pure functions
* Debugging is easier
* Testing generally is easier on pure functions
* Typically see component mark-up built in JSX
* Sponsored by Facebook

# What is Redux?

* Redux is a framework for managing the state for a web application, React components render that state
* A single data store contains the state for your app
* Your application emits an action that defines something that just happened that will affect the state Reducers specify how to change the state when the action is received
* Hot reloading of code changes
* State changes can be tracked, and replayed

# Why React + Redux?

* The props for React components come from the Redux store that tracks the state.
* React components react to user input and emit actions, either directly or indirectly.
* Redux handles the action by running the appropriate reducers which transform the current state into a new state.
* React components react to the new state and update the DOM.
* React components themselves are stateless (most of the time), all of the state is kept in the Redux store, one common place, for simplicity.

# Terminology

- (REACT)

* Component
* Props
* propTypes
* render
* JSX

- (REDUX)

* Action
* Action Creator
* Dispatch
* Reducer
* Store

-(React + Redux)

* mapStateToProps
* mapDispatchToProps
* connect
* from View component src/views/TimecardView.js



