

# React + Redux

## Questions

- I have a Redux app, how do I add React?
- What is the `react-redux` npm module?
- What is a Provider?
- What is a "Dumb Component"?
- What is a "Smart Container"?
- What is the `connect` function?
- What is `mapStateToProps`?
- What is `mapDispatchToProps`?
- Why do I bring in my action creators into a Smart Container?

# I have a Redux app, how do I add React?

## Setup

- make a separate create-react-app project
- in that project, `npm install redux react-redux`
- copy over your `store.js` file from your redux project
- change your `store.js` requires to imports and your `module.exports` to `export`

## What is the `react-redux` npm module?

- it's "glue" module that lets react and redux talk to each other

## What is a Provider?

- it's a component that is part of the `react-redux` npm module
- you wrap your whole react + redux app in it
- you give it a reference to your redux store
- it makes the store available to everybody

## What is a "Dumb Component"?

- it's just a function component
- it accepts props
- it returns some JSX

## What is a "Smart Container"?

- it's a file
- it has two functions that describe how to translate redux state into react props
- it returns a new, enhanced version of a Dumb Component

## What is the connect function?

- wires together a redux store and a Dumb Component
- it returns a new, enhanced version of the Dumb Component



## What is `mapStateToProps`?

- a function
- receives a copy of the current state
- it returns an object that says what React props equal what Redux state

## What is `mapDispatchToProps`?

- a function
- receives a reference to Redux's `store.dispatch`
- it returns an object that lists React props, whose values are our helper functions
- the helper functions call `store.dispatch`

## Why do I bring in my action creators into a Smart Container?

- you need to call them and pass the result to `dispatch`