Connecting React to Redux



Cory House

@housecor

reactjsconsulting.com



Here's The Plan



Container vs Presentation Components

React-Redux

- Provider
- Connect

A Chat with Redux



Two Component Types

Container

Aware of Redux
Subscribe to Redux State
Dispatch Redux actions
Generated by react-redux

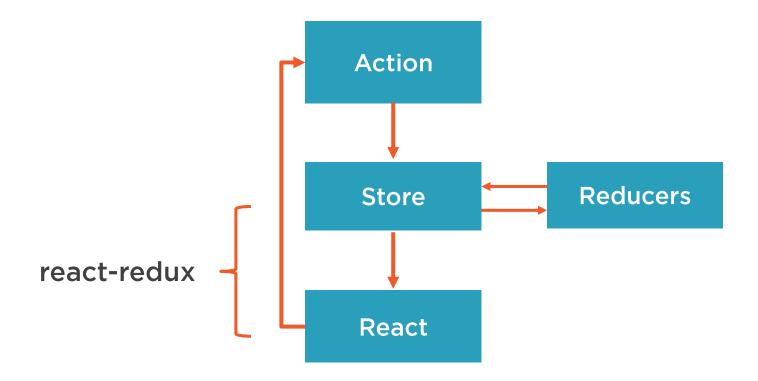
Presentational

Focus on how things look
Unaware of Redux
Read data from props
Invoke callbacks on props
Written by hand



Connecting React to Redux





React-Redux

Provider

Connect

Attaches app to store

Creates container components



React-Redux Provider

```
<Provider store={this.props.store}>
    <App />
    </Provider>
```



Connect

Wraps our component so it's connected to the Redux store.

export default connect(mapStateToProps, mapDispatchToProps)(AuthorPage);



Flux

```
componentWillMount() {
  AuthorStore.addChangeListener(this.onChange);
componentWillUnmount() {
   AuthorStore.removeChangeListener(this.onChange);
onChange() {
   this.setState({ authors: AuthorStore.getAll() });
```



Redux

```
function mapStateToProps(state, ownProps) {
  return { authors: state.authors };
}
export default connect(mapStateToProps, mapDispatchToProps)(AuthorPage);
```

Benefits:

- 1. No manual unsubscribe
- 2. Declare what subset of state you want
- 3. Enhanced performance for free



React-Redux Connect

```
connect(mapStateToProps, mapDispatchToProps)
      What state should I expose as props?
function mapStateToProps(state) {
  return {
    appState: state
```



React-Redux Connect

```
connect(mapStateToProps, mapDispatchToProps)
      What state should I expose as props?
function mapStateToProps(state) {
  return {
    users: state.users
```



Reselect

Memoize for performance

```
const getAllCoursesSelector = state => state.courses;

export const getCoursesSorted = createSelector(
  getAllCoursesSelector,
  courses => {
    return [...courses].sort((a, b) =>
       a.title.localeCompare(b.title, "en", { sensitivity: "base" })
    );
});
```



React-Redux Connect

```
connect(mapStateToProps, mapDispatchToProps)
                         What actions do I want on props?
function mapDispatchToProps(dispatch) {
  return {
    actions: bindActionCreators(actions, dispatch)
```



4 Ways to Handle mapDispatchToProps

- 1. Ignore it
- 2. Wrap manually
- 3. bindActionCreators
- 4. Return object



Option 1: Use Dispatch Directly

```
// In component...
this.props.dispatch(loadCourses())
```

Two downsides

- 1. Boilerplate
- 2. Redux concerns in child components



Option 2: Wrap Manually

```
function mapDispatchToProps(dispatch) {
  return {
    loadCourses: () => {
      dispatch(loadCourses());
    createCourse: (course) => {
      dispatch(createCourse(course));
    updateCourse: (course) => {
      dispatch(updateCourse(course));
// In component...
this.props.loadCourses()
```



Option 3: bindActionCreators

```
function mapDispatchToProps(dispatch) {
  return {
    actions: bindActionCreators(actions, dispatch)
  };
                       Wraps action creators in dispatch call for you!
   In component:
this.props.actions.loadCourses();
```



Option 4: mapDispatchToProps as Object

```
const mapDispatchToProps = {
   loadCourses
};
   Wrapped in dispatch automatically

// In component:
this.props.loadCourses();
```



4 Ways to Handle mapDispatchToProps

```
this.props.dispatch(loadCourses());
                                                      Ignore it
function mapDispatchToProps(dispatch) {
                                                       Manually wrap in dispatch
  return {
    loadCourses: () => dispatch(loadCourses();
  };
function mapDispatchToProps(dispatch) {
                                                      Use bindActionCreators
  return {
    actions: bindActionCreators(actions, dispatch)
  };
const mapDispatchToProps = {
                                                       Return object
  incrementCounter
```



A Chat With Redux

React Hey CourseAction, someone clicked this "Save Course" button.

Action Thanks React! I will dispatch an action so reducers that care can update

state.

Reducer Ah, thanks action. I see you passed me the current state and the action

to perform. I'll make a new copy of the state and return it.

Store Thanks for updating the state reducer. I'll make sure that all connected

components are aware.

React-Redux Woah, thanks for the new data Mr. Store. I'll now intelligently determine

if I should tell React about this change so that it only has to bother with

updating the UI when necessary.

React Ooo! Shiny new data has been passed down via props from the store! I'll

update the UI to reflect this!

Action Objects are Components Store Reducers creators dispatched invokes receive new generate create to the store reducers new state state objects



Time to code!





Next up: Let's code Redux!

