zeolearn™

React-Redux

Hottest Pair

React - Redux - Bindings

- Redux is independent of React, it is a state management library for any JS applications
- React is a view library, a JS library
- React-Redux bindings is a glue between react and redux.

React - Redux - Bindings - Setup

zeolearn™

```
npm install react-redux --save
```

Presentational vs Container Components

zeolearn™

	Presentational Components	Container Components
Purpose	How things look (markup, styles)	How things work (data fetching, state updates)
Aware of Redux	No	Yes
To read data	Read data from props	Subscribe to Redux state
To change data	Invoke callbacks from props	Dispatch Redux actions
Are written	By hand	Usually generated by React Redux

React - Redux - API

zeolearn™

`<Provider store>`

Makes the Redux store available to the `connect()` calls in the component hierarchy below. Normally, you can't use `connect()` without wrapping the root component in `<Provider>`.

`connect([mapStateToProps], [mapDispatchToProps], [mergeProps], [options])`

Connects a React component to a Redux store. `connect` is a facade around `connectAdvanced`, providing a convenient API for the most common use cases.

React - Redux - In Action

zeolearn™

```
const store = createStore(rootReducer);

ReactDOM.render(
<Provider store={store}>
  <Router history={browserHistory}>
    <Route path="/" component={App}>
      <IndexRoute component={Home}/>
      <Route path="posts" component={PostList}/>
      ...
    </Route>
  </Router>
</Provider>,
document.getElementById('root')
);
```

```
import React from 'react';
import {connect} from 'react-redux';
import {bindActionCreators} from 'redux';

class Signup extends React.Component {
….
function mapStateToProps(state) {
return {
  userSignedUp: state.users.userSignedUp
}
}

function mapDispatchToProps(disptach) {
return {
  actions: bindActionCreators(UserActionCreators, disptach)
}
}
export default connect(mapStateToProps, mapDispatchToProps)(Signup);
```