

Higher Order Components

Mallory Bulkley

What are Higher Order Components?

- Functions that take a component as an argument and return another component
- Components that wrap other components

Why should I use HOCs?

- Composition > Inheritance
- Keep code DRY
- Define abstractions
- Extend the wrapped component
- Modify props
- Modify child components

You already know one!

- connect in react-redux takes a component and returns a new component that is connected to the Redux store

```
import { connect } from 'react-redux';  
  
export default connect(mapStateToProps, mapDispatchToProps)(Component);
```

```
1 import React from 'react';
2 import { merge } from 'lodash';
3
4 function getDisplayName(WrappedComponent) {
5   return WrappedComponent.displayName || WrappedComponent.name || 'Component';
6 }
7
8 export default function expandable(WrappedComponent, title, options) {
9   const defaults = { expanded: false, disabled: false };
10
11   class Expandable extends React.Component {
12     constructor(props) {
13       super(props);
14
15       this.state = merge(defaults, options);
16       this.toggleExpand = this.toggleExpand.bind(this);
17     }
18
19     toggleExpand() {
20       this.setState({ expanded: !this.state.expanded });
21     }
22
23     render() {
24       const expand = this.state.expanded && !this.state.disabled;
25       const arrow = <span style={{ transform: expand ? '' : 'rotate(-90deg)' }} className="section-arrow" />;
26
27       return (
28         <div className={this.state.disabled ? 'expandable disabled' : 'expandable'}>
29           <div className="header" onClick={this.toggleExpand}>{title} {arrow}</div>
30           {expand ? <WrappedComponent {...this.props} /> : null}
31         </div>
32       );
33     }
34   }
35
36   Expandable.displayName = `Expandable(${getDisplayName(WrappedComponent)})`;
37   Expandable.WrappedComponent = WrappedComponent;
38
39   return Expandable;
40 }
```

Tips & Tricks

- Manually set the displayName to make debugging easier
- Don't mutate the wrapped component!
- HOCs & "ref"s: ref will refer to the wrapper component rather than the wrapped component
 - You can use a callback to get at the inner ref if necessary

More resources

- <https://facebook.github.io/react/docs/higher-order-components.html>
- <https://medium.com/@franleplant/react-higher-order-components-in-depth-cf9032ee6c3e>
- <https://medium.com/javascript-scene/composing-software-an-introduction-27b72500d6ea>