30 Days Of React: Higher Order Component

LinkedIn Follow @asabeneh 776

Author: Asabeneh Yetayeh

October, 2020

<< Day 15 | Day 17 >>

16
DAY
30
DAYS OF REACT
1 - 30 Oct 2020
THEORIES | EXERCISES | PROJECTS
CodingForAll.io
<CodingForAll title = 'Everyone can code' />

Higher Order Component

The term higher order component is similar to higher order function in JavaScript. In JavaScript, a higher order function is a function that takes another function as a parameter or return another function.

Similar to higher order function, a higher order component takes a component and return another component. This definition will make sense with examples. Look at the example below for better understand.

```
// One way of writing a Higher Order Component(HOC)
import React from 'react'
const higherOrderComponent = (Component) => {
  return (props) => {
    return <Component {...props} />
  }
}
```

Most of the time third party libraries use higher order component. For instance redux, react-router-dom and material-u use higher order component.

```
import React from 'react'

const Button = ({ onClick, text, style }) => {
  return (
    <button onClick={onClick} style={style}>
      {text}
    </button>
  )
}

const buttonWithStyle = (CompParam) => {
  const buttonStyles = {
    backgroundColor: '#61dbfb',
    padding: '10px 25px',
    border: 'none',
    borderRadius: 5,
    margin: 3,
    cursor: 'pointer',
    fontSize: 18,
    color: 'white',
  }
  return (props) => {
    return <CompParam {...props} style={buttonStyles} />
  }
}
const NewButton = buttonWithSuperPower(Button)

class App extends Component {
  render() {
    return (
      <div className='App'>
        <Button text='No Style' />
        <NewButton text='Styled Button' />
      </div>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

Let's make the buttonWithStyle higher order take more parameter in addition to the component.

```
import React from 'react'

const Button = ({ onClick, text, style }) => {
  return (
    <button onClick={onClick} style={style}>
      {text}
    </button>
  )
}

const buttonWithStyles = (CompParam, name = 'default') => {
  const colors = [
    {
      name: 'default',
      backgroundColor: '#e7e7e7',
      color: '#000000',
    },
    {
      name: 'react',
      backgroundColor: '#61dbfb',
      color: '#ffffff',
    },
    {
      name: 'success',
      backgroundColor: '#4CAF50',
      color: '#ffffff',
    },
    {
      name: 'info',
      backgroundColor: '#2196F3',
      color: '#ffffff',
    },
    {
      name: 'warning',
      backgroundColor: '#ff9800',
      color: '#ffffff',
    },
    {
      name: 'danger',
      backgroundColor: '#f44336',
      color: '#ffffff',
    },
  ]
  const { backgroundColor, color } = colors.find((c) => c.name === name)

  const buttonStyles = {
    backgroundColor,
    padding: '10px 45px',
    border: 'none',
    borderRadius: 3,
    margin: 3,
    cursor: 'pointer',
    fontSize: '1.25rem',
    color,
  }
  return (props) => {
    return <CompParam {...props} style={buttonStyles} />
  }
}

const NewButton = buttonWithSuperPower(Button)
const ReactButton = buttonWithSuperPower(Button, 'react')
const InfoButton = buttonWithSuperPower(Button, 'info')
const SuccessButton = buttonWithSuperPower(Button, 'success')
const WarningButton = buttonWithSuperPower(Button, 'warning')
const DangerButton = buttonWithSuperPower(Button, 'danger')

class App extends Component {
  render() {
    return (
      <div className='App'>
        <Button text='No Style' onClick={() => alert('I am not styled yet')} />
        <NewButton
          text='Styled Button'
          onClick={() => alert('I am the default style')}
        />
        <ReactButton text='React' onClick={() => alert('I have react color')} />
        <InfoButton
          text='Info'
          onClick={() => alert('I am styled with info color')}
        />
        <SuccessButton text='Success' onClick={() => alert('I am successful')} />
        <WarningButton
          text='Warning'
          onClick={() => alert('I warn you many times')}
        />
        <DangerButton
          text='Danger'
          onClick={() => alert('Oh no, you can not restore it')}
        />
      </div>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

The is above example is one use case of Higher Order Component. However, its use case is is more than just styling simple button. It has enormous use cases, it allow us to reuse component and enhance a component with style and functionality. In the coming sections, we will cover React Router and we will use HOC and you will not be surprised when you see one component wrap another component.

Exercises

Exercises: Level 1

1. What is higher order function
2. What is Higher Order Component
3. What is the difference between higher order function and higher order component?
4. A higher order component can allow us to enhance a component. (T or F)

Exercises: Level 2

1. Make a higher order component which can handle all the input type.

Exercises: Level 3

coming

🎉 CONGRATULATIONS ! 🎉

<< Day 15 | Day 17 >>