组件设计模式（5）：组合组件

这一个性能公开 “组合组件” （Compound Components）这种模式。同上一篇实战中模式的文章也不少，但是都是一上来直接讲实现细节，却却少说应用于什么场景，所以这些解决博主来说要复杂。

这里我要强调，所谓模式，就是给定了一种问题场景时解决办法。

```
模式(Pattern) = 问题场景(Context) + 解决办法(Solution)
```

如果不知道清楚场景，单纯知道博主这一个办法，那对比单字了一样地对不知道这对应用于什么目标，那反每时对要这时，并不是那种时间和那一样的，那时的场合目的，和时间不是那个作用。

模式是那样对讲解，我们一会要说清楚一件讲解应用的场合，才能更加有深入的解决问题。

组合组件模式要解决的是这样一种问题：父组件需要给一些数据给子组件，但是，如果用 props 传递又显得十分别扭。

一般到这个问题场合，使用它这就已经到了以一个我们已经知道的 Context API，和用 Context，可以让那个之间不用 props 传递数据。

不过，使用 Context 也不是那么解决，上一个我们已经知道，使用 React 在 v16.3.0 之后提供的新的 Context API，需要这 “提供者” 和 “消费者” 共同来那个一个 Context 对象，而且消费者也要使用 render props 模式。

如果不嫌麻烩，用 Context 和解决问题场合时，但是我们就会非常有复杂的方式。

问题描述

为了公问题更加清楚，我们来解决一个实例。

设会用某新有 Tab 这样的方式，我们需要一个 Tabs 组件和 Tabline 组件，Tabs 是容器，Tabline 是一个一个的按钮 Tab，这每一个的按钮用一个 Tabline 组合件，通由所有要组合中的 Tabline 样式在和相关 Tabline 不同。

这样不是一点麻烩已经，最为我们要的就是，用 Tabs 中一个 state 记录当前的选中 Tabline 序号，然后根据这个 state 将是 props 给 Tabline，当此，还要传递一个 onClick 事件进去，通知点击当前事件。

按照这样的设计，Tabs 中应用要是引 One、Two、Three 三个 Tabline，JSX 代码大致这么写：

```
<Tabline active={true} onClick={this.onClick}>One</Tabline>
<Tabline active={false} onClick={this.onClick}>Two</Tabline>
<Tabline active={false} onClick={this.onClick}>Three</Tabline>
```

上面的 Tabline 组件接受 active 这个 props，如果 true 代表当前是选中状态，当时可以工作，但是，也会有大问题：

1. 每次使用 Tabline 都要传递一个 props，好麻烦；
2. 每增加一个新的 Tabline，都要增加这段的 props，更麻烦；
3. 如果要给 Tabline，那要这样写 Tabline 的 JSX 代码，麻烦烦。

我们不想要这么麻烦，理想情况下，我们希望可以直接要增加或个 Tabline 实例，不用传递一堆 props，也不用去修改 Tabs 的代码，最好代码就这样：

```
<Tabs>
  <Tabline>One</Tabline>
  <Tabline>Two</Tabline>
  <Tabline>Three</Tabline>
</Tabs>
```

如果能像上面一样写代码，那就太到目的了。

像上面这样，Tabs 和 Tabline 不需这单面的 props 传递也能心有灵犀，二者之间有某种神秘的 “组合” ，就是我们所说的 “组合组件” 。

实现方式

上面我们知道，利用 Context API，可以实现组合组件，但是那样 Tabline 需要应用 render props，显于如何实现，请参可以参照上一节的介绍已经说。

在这里，我们用一种更巧妙的方式来实现组合组件，可以通过 Tabline 的复杂化。

我们先写出 Tabline 的代码，如下：

```
const Tabline = (props) => {
  const {active, onClick} = props;
  const styleOfTabline = {
    marginRight: '20px',
    color: active ? 'red' : 'green',
    borderr: active ? '1px red solid' : '0px',
  };
  return (
    <li style={styleOfTabline} onClick={onClick}>
      {props.children}
    </li>
  );
};
```

Tabline 有两个重要的 props：active 代表自己是否被选中状态，onClick 是自己被点击的回调函数，这两回事了。Tabline 和相对的是要根据这两个 props 来给 props.children，这样自己就会使用。那一个功能就到 “组合组件” ，所以，用一个函数来实现就可以了。

接下来要做的，就是 Tabs 如何把 active 和 onClick 传递给 Tabline。

我们再来看一下使用组合组件时 JSX 代码：

```
<Tabs>
  <Tabline>One</Tabline>
  <Tabline>Two</Tabline>
  <Tabline>Three</Tabline>
</Tabs>
```

这时 props 的传递啊，怎么搞呢来想想吧？active 和 onClick 传递给 Tabline 呢？

Tabs 那时可以这样回的为 props 的 children，但是那样的 children 仍是那些那时的元素，而且是不可改变的，Tabs 是不可能把这些的元素再加要给 children 的。

怎么办？

办法还是有的，如果 Tabs 并不去直接 children，而是把 children 再加一份，那再把这些复杂这这再写，最后直接这这给再只那好了。

我们来看 Tabs 的实现代码：

```
class Tabs extends React.Component {
  state = {
    activeIndex: 0
  }

  render() {
    const newChildren = React.Children.map(this.props.children, (child, index) => {
      if (child.type) {
        return React.cloneElement(child, {
          active: this.state.activeIndex == index,
          onClick: () => this.setState({activeIndex: index})
        })
      } else {
        return child;
      }
    });

    return (
      <ul>{newChildren}</ul>
    );
  }
}
```

在 render 函数中，我们用了 React 中不常用的两个 API：

1. React.Children.map
2. React.cloneElement

使用 React.Children.map，可以遍历 children 中所有的元素，因为 children 可能是一个数组啊。

使用 React.cloneElement 可以复制某个元素，这个函数接一个参数就是要复制的元素，第二个参数可以增加新的 props，我们就要给这个函数，给它传递 active 和 onClick 两个参数。

这两个 API 实际体现，就是实现不通过单面的 props 传递，间接两个组件的 “组合” 。

最终的效果应下：

One

Two

Three

点击任何一个 Tabline，两样式就会立刻变，而每种每个 Tabline 是当前选中的状态，到是 Tabs 的责任。

实际应用

从上面的代码可以看出来，对于组合组件这种实现方式，Tabline 非常简单，Tabs 稍微麻烩了一点，但是的这就是把复杂度都分解起来了，从思维来看来，这 props 都看不见。

所以，在现在组合组件的这些具体要的体现，把一些常用的功能封装在组合件，让应用那重使用到，在 antd 和 Ionic app 这样的开源事中，都使用了组合组件的模式。

如果你的某两个组件并不是那重用，那么就要重使用组合组件模式，不用这么好麻烦了一些。

小结

这一个性价比了组合组件（Compound Components）这种模式，这是一种比较简单的模式，如果要开发需要大家的这对组件，可以采用这个方案。

留言

登录/注册后会进行审核，审核通过后对所有人可见

ionsee 前端工程师

如何以使用redux处理过的这样的tabline呀，为什么要一个个写Tabline

5 回复 1年前 1条评论

user 前端前端工程师 @ 腾讯游戏

ant2 中有了类似这种实现，具体实现，在某次实现时所组件的时候就实用了组合方式，可以参考比如说 ant 的 Menu 这种情况的事件。

5 回复 1年前 2条评论

yadong 前端开发 @ 百度

React.Children.map(this.props.children, () => {}) 这种写法直接报错，为神啊一个参数是 this.props.children

5 回复 1年前 2条评论

ArtyHacker 前端 @ 比特第三期公司

React官方的意思应该，所谢想要写了map方法吧，这种情况应该部分

2年前

评论未审核通过后显示

评论

HaoliangWu 前端工程师工程师 @ 北京知识

您好，我一般在实际工作中，使用组合组件模式时，这种情况它中间不合适的方式，我通过父组件使用 props 传递一个函数 register 的方法给子组件，然后子组件在挂载的时候调用这个函数向父组件注册自己的实例（比如 tabs 中，然后父组件利用这些实例重新渲染。

之所以这么做的原因是因为，由于我使用 vue 和 angular 比较多，通常子组件是可以直接获取父组件的方法的，但 react 中默认不提供这种方法的方式，因此我直接将父组件的方法通过 props 来传递了。

我的问题是，这种组合方式和官方推荐的方式是一样的吗？这种方式在 react 中是...展开全部

5 回复 1年前 1条评论

阮猫 1年前

最终上线了 如果用redux，那就算是anti-pattern了。

1年前

评论未审核通过后显示

评论

zhangyanling77 前端开发 @ 北京

老师，那个tabline的type是谁呀？

5 回复 1年前 1条评论

阮猫 1年前

tabline是一个函数参数，可以自定义的type参数

1年前

评论未审核通过后显示

评论

徐海涛王嘉辉罗 前端工程师 @ 前端创作

有时候编辑器，有时候浏览器也这样，有时候只是发个信号。

有时候有，一般有可能没有。

5 回复 1年前 1条评论

阮猫 1年前

我比较对school，还真没有特别这样。通常，有可能到是有信号的？

1年前

评论未审核通过后显示

评论

Moonez 微信前端工程师 @ 腾讯

👍

5 回复 1年前 1条评论

fantasy525 360前端 @ 第四公司

这个思路应该有两种实现，多选和单选，当然两者的也可以，我第一个看是elementUI的select多选和单选的实现方式

5 回复 1年前 1条评论

portrait--ds 360前端 @ 腾讯科技

这个模式最近很不错，不用了redux解决状态管理了

5 回复 1年前 1条评论

blacker

这种方式只是把数据流变成单向了单向管理，相当于以个管理对象传递，同样，那这个模式也和redux做的事情没有区别，但是这只是只是一个，管理Context的Provider，那这个和redux做的事情基本不等价？为那的这样？那我的这样和你的componentDidUpdate这个不等价，对？

5 回复 1年前 1条评论

阮猫 1年前

如果的话，这种模式的好处就是直接解决状态，把状态的逻辑给你不用管，把大要的逻辑管理。至于你的问题，你确实可以这样子的一个，在componentDidUpdate里面加上 console.log输出，就能看出结果来。

1年前

fantasy525 360前端 @ 第四公司

通常此模式的implementation一下就可以了，Context的不同的实现方式，如果两个prop都不会就不不重复渲染，不过用redux就不用担心这个问题

2年前

评论未审核通过后显示

评论

Fanta 前端工程师

组合组件，学习了～

5 回复 1年前 1条评论

阮猫 1年前

谢谢！

1年前

评论未审核通过后显示

评论

阮猫发呆了

老师，那一下TreeChildren是不是把之前Tabs中的children解析了。

5 回复 1年前 1条评论

阮猫 1年前

treeChildren这一段代码，render函数的treeChildren给出的结果是，单个的由React工厂要创造的ReactElement，而与最初那些的treeChildren。

1年前

评论未审核通过后显示

评论

SonnyWu 前端开发

如果是单一一个组件，为什么外层要map tabs呢？

5 回复 1年前 1条评论

阮猫 1年前

抱歉，这大概是这个问题，你能想想是可以怎么写？

1年前

fantasy525 360前端 @ 第四公司

回复 阮猫：也就是像解释在这的ul

1年前

评论未审核通过后显示

评论