技术交流汇报

React 之 React Hooks 从入门到“放弃”？

什么是React Hooks？

汇报人 老人与海鲜

首先让我们来回想一下开发过程

组件的开发

在当前业务中最常用的几种组件

类组件

通过本地状态管理与生命周期方法决定业务逻辑的书写。

无状态组件

无状态组件的行为不依赖它的状态，可以是类组件也可以是函数组件。

高阶组件

通过传入一个组件并返回一个新的组件。

什么是React Hooks?

1. 如果我们想要将一个无状态组件重构为类组件，我们需要重写该组件并创建其状态

2. 我们常常需要处理复杂的生命周期关系，并伴随着设置与清理阶段，例如：如果未移除监听事件可能带来React性能上的影响。

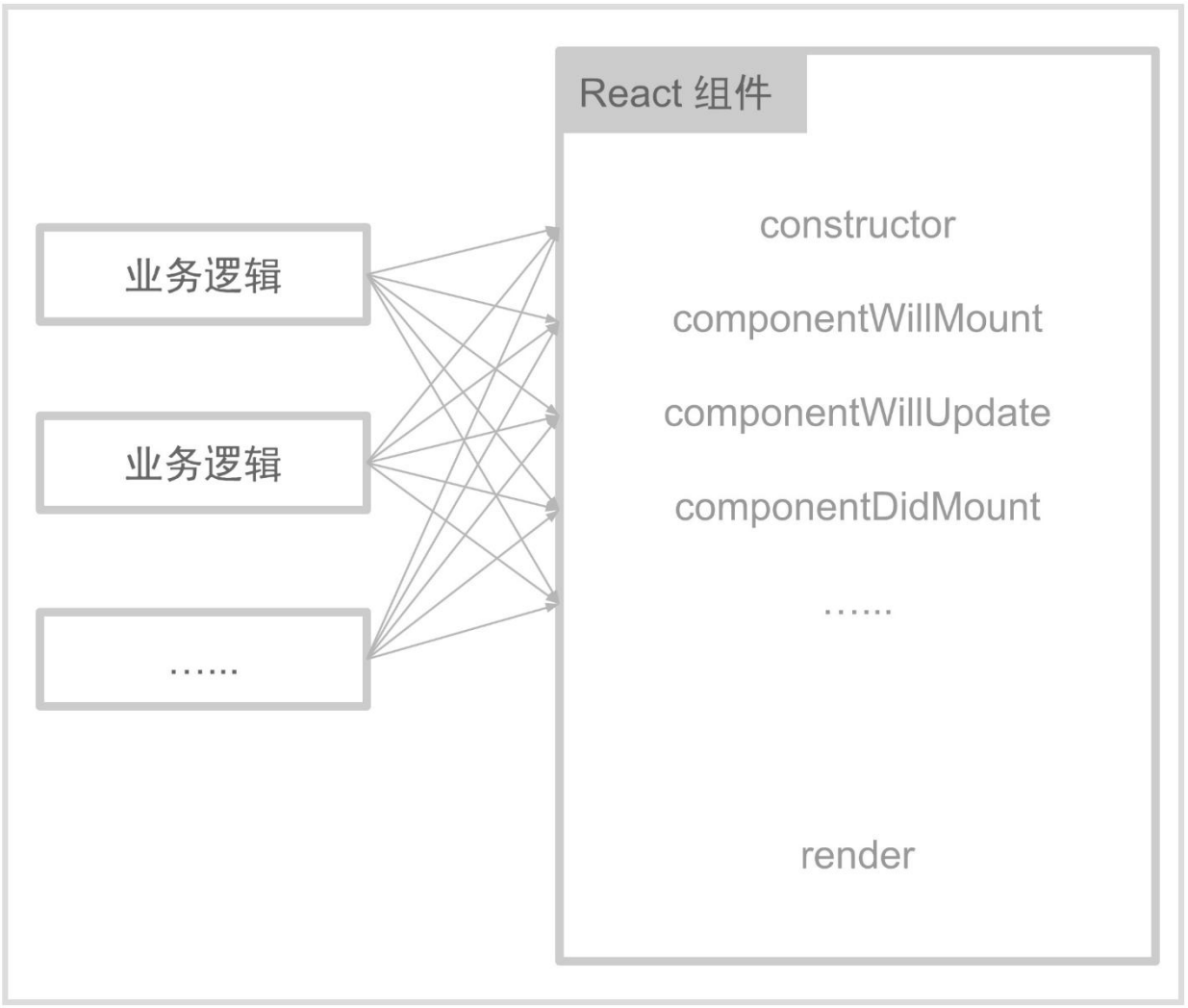
3. 在 React 中使用 higher-order-component（HOC）而带来的嵌套地狱。

4. 在React中， class只是声明，而组件的实际用法是它的实例化，既 this 与 setState 的交互，我们需要在很多位置处理this的指向

我们先假想一个常见的需求，一个 Modal 里需要展示一些信息，这些信息需要通过 API 获取且跟 Modal 强业务相关。

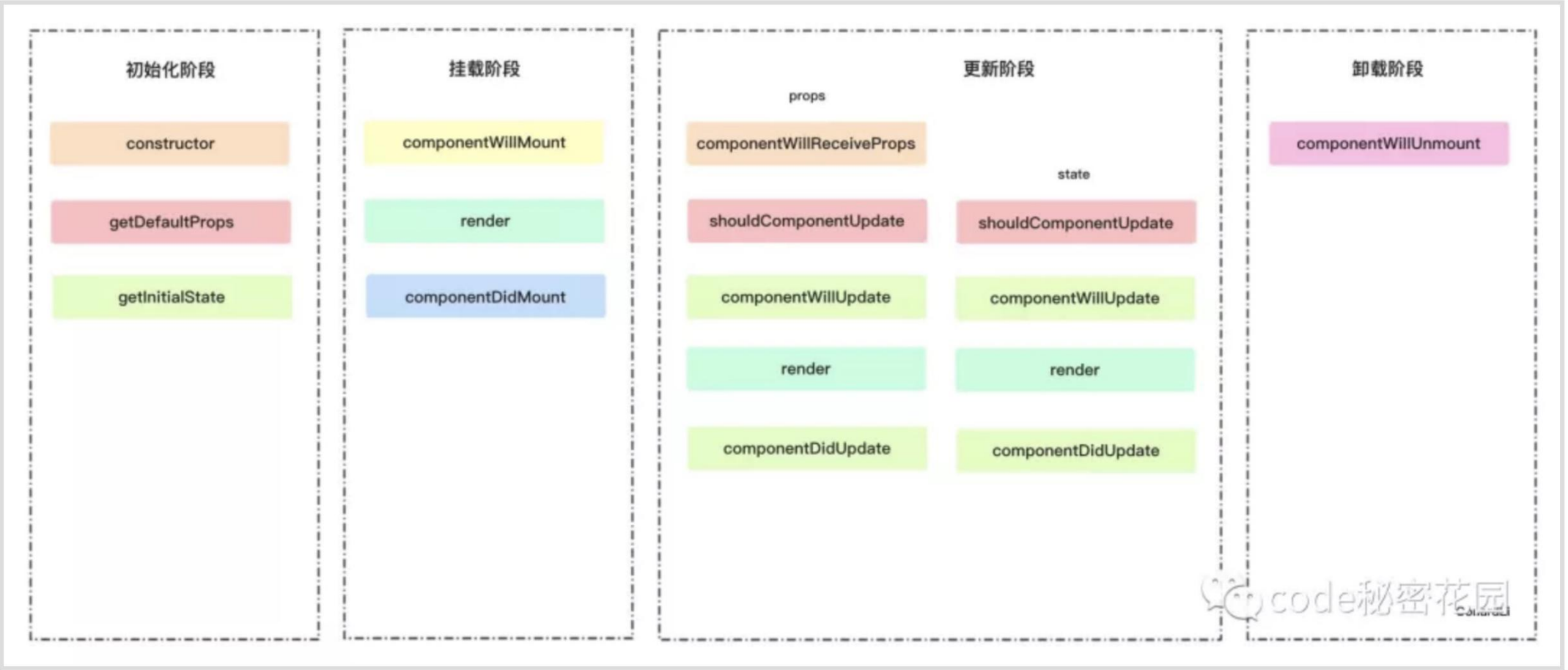
1. 因为业务简单，没有引入额外状态管理库
2. 因为业务强相关，并不想把数据跟组件分开放
3. API 数据会随机变动，因此需要每次打开 Modal 才获取最新数据
4. 为了后期优化，不可以有额外的组件创建和销毁

当前的开发模式，当业务足够庞大时，我们会有着相当量的组件，并且业务逻辑分布在各个生命周期中，每一个生命周期扮演着重要的角色，但是它们之间的关系又难以处理。

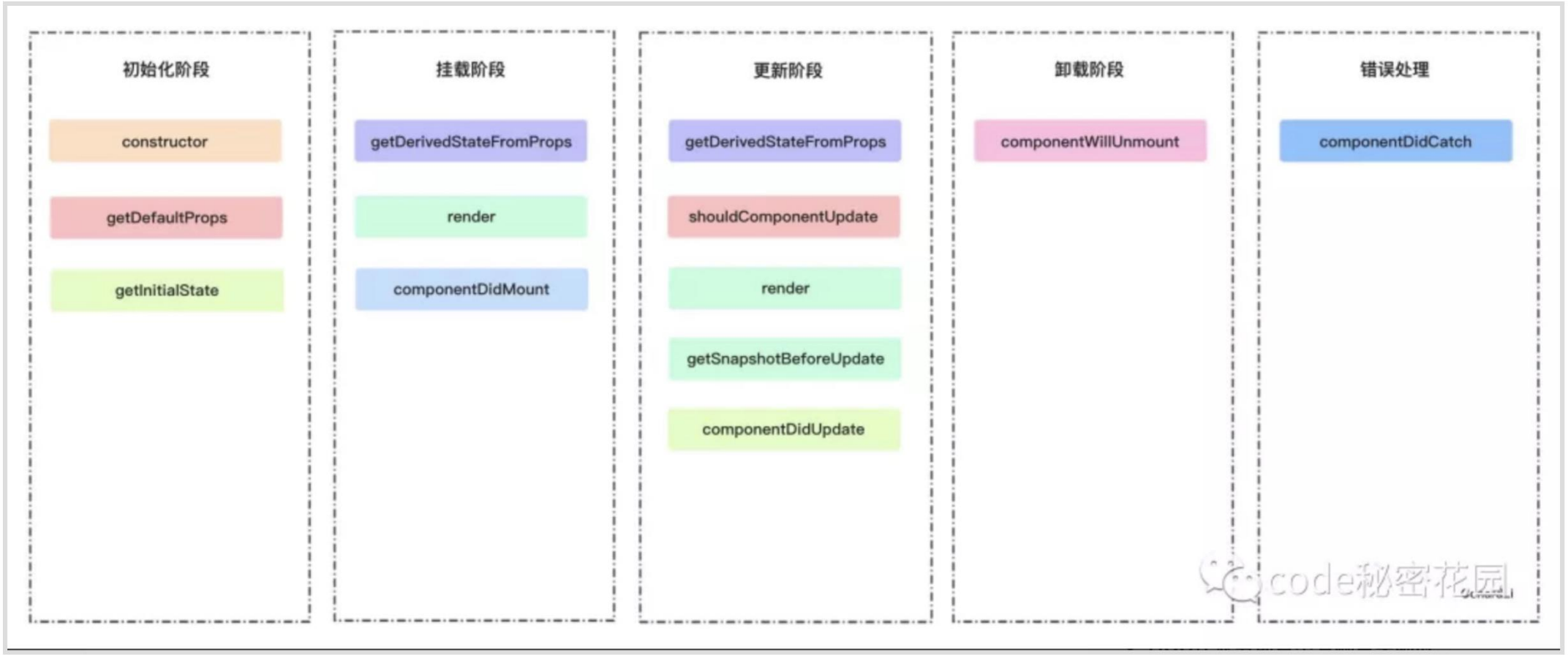


React Hooks 从入门到放弃

React复杂而又繁多的生命周期



React复杂而又繁多的生命周期



React Hooks 从入门到放弃

高阶组件的终极状态

```
▼ <BasicVerificationContainer> == $r
 ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))))))))) stepTwoVerified={false}>
  ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))))))) stepTwoVerified={false}>
   ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))))))) stepTwoVerified={false}>
    ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))))) stepTwoVerified={false}>
     ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))))) stepTwoVerified={false}>
      ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))) stepTwoVerified={false}>
       ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))) stepTwoVerified={false}>
        ▼ <withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))) stepTwoVerified={false}>
         ▼ <withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))) stepTwoVerified={false}>
          ▼ <withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))) stepTwoVerified={false}>
           ▼ <withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))) stepTwoVerified={false}>
            ▼ <withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))) stepTwoVerified={false}>
             ▼ <withState(withState(withState(withState(Apollo(withState(BasicVerification)))))) stepTwoVerified={false}>
              ▼ <withState(withState(withState(Apollo(withState(BasicVerification))))) stepTwoVerified={false}>
               ▼ <withState(withState(Apollo(withState(BasicVerification)))) stepTwoVerified={false}>
                ▼ <withState(Apollo(withState(BasicVerification))) stepTwoVerified={false}>
                 ▼ <Apollo(withState(BasicVerification)) stepTwoVerified={false}>
                  ▼ <Mutation errorPolicy="all" ignoreResults={true}>
                   ▶ <withState(BasicVerification) stepTwoVerified={false}>…</withState(BasicVerification)>
                  </Mutation>
                 </Apollo(withState(BasicVerification))>
                </withState(Apollo(withState(BasicVerification)))>
               </withState(withState(Apollo(withState(BasicVerification))))>
              </withState(withState(withState(Apollo(withState(BasicVerification)))))>
             </withState(withState(withState(withState(Apollo(withState(BasicVerification))))))>
            </withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))>
           </withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))>
          </withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))>
         </withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))>
        </withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))>
       </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))>
      </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))))>
     </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))))>
    </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))))))>
   </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))))))>
  </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification)))))))))))))))))>
 </withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(withState(Apollo(withState(BasicVerification))))))))))))))))))>
</BasicVerificationContainer>
```

Render Props

它的实现思路很简单，把原来该放「组件」的地方，换成了回调，这样当前组件里就可以拿到子组件的状态并使用。

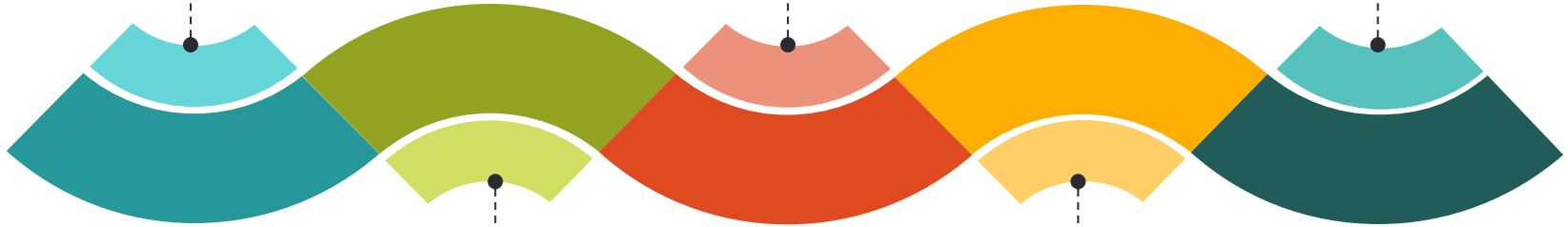
```
class App extends React.Component {
  render() {
    return (
      <ThemeProvider>
        <ThemeContext.Consumer>
          {val => <div>{val}</div>}
        </ThemeContext.Consumer>
      </ThemeProvider>
    )
  }
}
```

在 React Hooks 中，我们更加专注业务逻辑的书写，我们不需要将大量的业务逻辑分散在各个生命周期中，这使得业务逻辑更加集中，我们可以专注于业务。

useState
替换传统类组件中的State

useForceUpdate
强制渲染组件

灵活的操作
可以通过对hooks的封装，使其更容易被使用



useEffect
替代了传统类组件的部分生命周期

useRef
可以用来存储Dom或者存储当前的状态并在组件中的任何位置使用

协同开发

UI组件 与 业务组件 的拆分

减少组件的重构

通过 hooks 我们可以减少对
无状态组件的重构

面向业务进行开发

我们不需要在业务中过多顾虑生命周
期所带来的的影响

多元化的组件变换

无状态组件 与 hooks 之间的
完美转换

React Hooks 从入门到放弃

Hooks 存在的意义

Hooks 可能是 React 的未来，但还是无法完全替代原始的 Class。

Hooks做出了哪些改变

Hooks本质是把面向生命周期改为面向业务逻辑。
这里，做一个类比，await/async 本质是把 JS 里异步编程思维变成了同步思维，写法上表现出来的特点就是原来的 Callback Hell 被打平了。

不在生命周期上浪费时间

不在浪费大量的时间在生命周期的处理上，实际上工作中很多问题都出现在了生命周期的处理上

谢谢

感谢您的观看