



React类组件生命周期学习总结

Created and last modified by 王文磊 on Jan 13, 2022

旧

新

Initialization

setup props and state

Mounting

componentWillMount

render

componentDidMount

Updation

props

componentWillReceiveProps

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

states

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

Unmounting

componentWillUnmount

一、生命周期的作用

生命周期是每一个React类组件都拥有的

方法，官方提供这些方法让我们重写，以便于结合业务场景在运行过程中特定的阶段执行这些方法

@稀土掘金技术社区

二、各个生命周期的触发时机与职责

常用：

render

触发时机：

挂载时在 constructor > getDerivedStateFromProps 之后

更新时在 constructor > getDerivedStateFromProps > shouldComponentUpdate (返回true) 后被调用

职责：

1. 根据返回类型渲染DOM

规避badcase：

1. render函数应该为纯函数，即state与props不变时保证相同输出
2. 补充1，避免在render中实现交互，保证纯函数
3. 不要在render中调用setState，否则会触发死循环导致崩溃

constructor

触发时机：React组件的构造函数，在组件挂载之前被调用

职责：

1. 定义state，为state赋初值
2. 为事件处理函数绑定实例

规避badcase:

1. 在所有语句之前调用super(props)，否则this.props在构造函数中可能会出现未定义的问题
2. 在constructor()中不要调用setState()方法，因为此时第一次render()还未执行，也就意味着DOM节点还未挂载
3. 避免将props的值复制给state，因为这不符合state的设计，如果需要将prop作为state某个属性的初值，请使用独立的变量名

• componentDidMount

触发时机: 在组件挂载后（插入DOM树中）立即调用

职责:

1. 发送网络请求
2. 启用事件监听

规避badcase:

1. componentDidMount中可以调用setState在浏览器更新屏幕之前触发额外渲染，除非你的渲染依赖于DOM节点的大小或位置，否则尽量避免这种使用，因为它会导致性能问题

• componentDidUpdate

触发时机: 更新后会被立即调用，首次渲染不会执行此方法

职责:

1. 比较更新前后的props
2. 进行DOM操作
3. 发送网络请求

规避badcase:

1. componentDidUpdate中使用setState与发送请求时必须被包裹在一个条件语句中（如：比较props），否则会导致死循环，谨慎使用setState，因为它会导致额外的渲染，影响组件性能。
2. 不要将props复制给state，因为这种场景下父组件导致的更新会使子组件缓存的state丢失（重新赋为props）

• componentWillUnmount

触发时机: 在组件卸载即销毁之前被调用

职责:

1. 取消网络请求（AbortController）
2. 移除监听事件
3. 清理DOM元素
4. 清理定时器

规避badcase:

1. componentWillUnmount中不应调用setState

不常用:

• shouldComponentUpdate

触发时机: 在组件更新之前调用，首次渲染或使用forceUpdate时不会调用该方法

职责:

1. 控制组件是否进行更新

规避badcase:

1. 不要在该方法中进行深层比较或使用JSON.stringify(), 这样非常影响效率, 且会损害性能
2. 不要在该方法中调用setState, 否则会导致无限循环调用更新、渲染, 直至浏览器内存崩溃

• getDerivedStateFromProps (static)

触发时机: 每次调用render方法前都会被调用

职责:

1. 返回一个对象用来更新state

规避badcase:

1. 该方法无法执行副作用, 需要响应props变更执行副作用请改用componentDidUpdate

• getSnapshotBeforeUpdate

触发时机: 在最近一次渲染输出 (提交到DOM节点) 之前调用, 即render之后, 即将对组件进行挂载时

职责:

1. 使组件能在发生更改之前从DOM中捕获一些信息 (如滚动信息)

规避badcase:

1. 该方法的任何返回值将传递给componentDidUpdate的第三个参数, 如果不需要请返回null

• getDerivedStateFromError

触发时机: 后代组件抛出错误后被调用

职责:

1. 处理后代组件抛出的错误, 并返回一个值更新state

规避badcase:

1. 该方法会在渲染阶段调用, 因此不允许出现副作用

• componentDidCatch

触发时机: 后代组件抛出错误后被调用

职责:

1. 处理后代组件抛出的错误

规避badcase:

1. 该方法会在提交阶段被调用, 因此允许执行副作用

过时:

由于React纤程的出现, 导致 **Reconciliation** 这个过程有可能暂停然后继续执行, 所以挂载和更新之前的生命周期钩子就有可能不执行或者多次执行

• UNSAFE_componentWillMount

触发时机：在组件挂载、render之前被调用

职责：

1. 调用setState修改state，但更建议在constructor中进行初始化

规避badcase：

1. 避免在此方法中引入任何副作用或订阅，建议改用componentDidMount

• UNSAFE_componentWillReceiveProps

触发时机：在已挂载的组件接收新的props之前被调用

职责：

1. 使用this.setState()执行state转换

规避badcase：

1. 处理更改时确保进行当前值与变更值的比较

• UNSAFE_componentWillUpdate

触发时机：组件收到新的props或state时，在渲染之前调用，初始渲染不会调用此方法

职责：

1. 更新时的转场动画

规避badcase：

1. 不能在该方法中使用this.setState或执行其他会触发React组件更新的操作

Like Be the first to like this