



Created and last modified by 王文磊 on Jan 23, 2022

Questions:

Q: 为什么必须调用setState改变状态

A:

1. 因为React并不像Vue2中调用Object.defineProperty实现数据响应式或者像Vue3调用Proxy监听数据的变化，只有通过调用组件的原型方法setState才能触发React的updateComponent实现状态刷新
2. React遵循的Immutable原则，若直接修改state的值，在触发组件shouldComponentUpdate生命周期时oldState与newState就已经相同了，但组件的渲染结果并没有更新

Q: setState执行，触发了哪些生命生命周期

A: `componentWillReceiveProps => shouldComponentUpdate`
`=> componentWillUpdate => render => componentDidUpdate`

Q: setState的同步异步场景，及产生原因

A:

场景：

- 在组件生命周期或React合成事件中，**setState**是异步
- 在setTimeout或者原生dom事件中，**setState**是同步

原因：

```
if (!batchingStrategy.isBatchingUpdates) {  
  batchingStrategy.batchedUpdates(enqueueUpdate, component);  
  return;  
}  
dirtyComponents.push(component);
```

解释：

如果处于批量更新模式，也就是isBatchingUpdates为true时，不进行state的更新操作，而是将需要更新的component添加到dirtyComponents数组中；如果不处于批量更新模式，对所有队列中的更新执行

batchedUpdates方法（事务更新）。

而在react的生命周期和合成事件中，react仍然处于他的更新机制中，这时`isBatchingUpdates`为true。按照上述过程，这时无论调用多少次`setState`，都不会执行更新，而是将要更新的state存入`_pendingStateQueue`，将要更新的组件存入`dirtyComponent`，从而展现出了异步的特征。

原生事件绑定不会通过合成事件的方式处理，自然也不会进入更新事务的处理流程。`setTimeout`也一样，在`setTimeout`回调执行时已经完成了原更新组件流程，不会放入`dirtyComponent`进行异步更新，其结果自然是同步的。

Q: PureComponent是怎么实现的

A: `React.PureComponent` 与 `React.Component` 很相似。两者的区别在于 `React.Component` 并未实现 `shouldComponentUpdate` 方法，而 `React.PureComponent` 中以浅层对比 `prop` 和 `state` 的方式来实现了该函数。

浅层对比即判断`props / state` 键数量，以及值是否为同一引用

Usage:

设计组件state

1. 把整个状态按照领域分成若干个子状态，子状态之间不能保存重复数据。
2. `state`以键值对的结构存储，以记录的`key`或者`id`作为记录的索引，记录中的其他字段都依赖于索引。
3. `state`中不能保存可以通过`state`中以有的字段计算二来的数据，既`state`中的字段不能相互依赖。

setState使用注意事项

1. 不能直接对 `this.state` 进行修改
2. 什么时候修改，什么时候对 `state` 操作，不要提前操作
3. 不能直接对数组进行 `pop`、`push`、`splice` 等操作（对象 `delete` `setter` `assign` 等）
4. 直接传递对象的`setState`会被合并成一次，使用函数传递`state`不会被合并，因为函数接收的参数`preState`永远是上一次`setState`执行后的`state`
5. 在组件生命周期或React合成事件中，`setState`是异步。在`setTimeout`或者原生`dom`事件中，`setState`是同步

PureComponent应用场景

当组件的渲染内容只依赖`props`与`state`且该组件的`props`与`state`结构简单时（大多数情况下为纯展示组件），可以使用`pureComponent`来提升性能

此外，由于 `pureComponent` 的 `shouldComponentUpdate` 钩子会跳过子组件`props`的变化，因此需要确保`pureComponent`的`childComponent`也是`pureComponent`

Like Be the first to like this