

32 条评论

默认 最新

Redux 状态更新为什么会从根结点开始全部重新渲染？只有订阅过状态更新的组件会重新跑一遍回调函数，跑完这个函数，组件的状态也未必改变，最终就是该渲染的渲染，不该渲染的不渲染。

准确的说应该是Redux状态每次更新，所有subscribe的回调都会跑一遍，这才是问题所在，整个系统只要有一个回调写得烂，整个系统都会被拖垮，典型的一粒老鼠屎害了一锅粥。

inner 上面用 react 原本的 state 就行
引入 redux 每次更新，从整个 App
开始渲染，每改动一次就会全部渲染
到爆。为啥改动页面局部的一个
刷新整个网站甚至 layout 呢？再

2020-05-04

● 回复 ● 喜欢

**Rabibabo** · 于江水在银河系

store是一个constant，它在react-redux中用context传给整个应用，但因为store其自身的引用并不会变化，所以不会触发context的变化而导致组件更新。

其实完全可以用`export const store = createStore()`来让整个应用可以import到。react-redux之所以用context来传，我估计还是为了简化API，否则connect函数和useSelector就要手动指定store了。

2020-05-05

● 回复 ● 1

**安兹** · 于江水在银河系

确实是根节点订阅，推送整个页面的connect组件，但是connect高阶组件会做性能优化，只有state改了才会更新

2020-05-05

● 回复 ● 喜欢

展开其他 3 条回复 >

**莫等闲**

一直在用mobx，好用，而且性能好。

2020-05-04

● 回复 ● 喜欢

**黄cJ**

MobX 的文档入门比较困难...

2020-05-07

● 回复 ● 喜欢

**PunCha Feng**

加的时候都说为了将来扩展，而且业界都在用，无力反驳

2020-05-04

● 回复 ● 喜欢

**于江水在银河系** 作者

引入很简单，知道为什么引入却很难，需要针对实际情况衡量。就我那个 portal 的例子，将来是 container 横向扩展自己维护内部的 state 即可，甚至也可以在 container 级别加 redux 如果很复杂，不引入全局 redux 也并没有扩展问题。

2020-05-04

● 回复 ● 喜欢

**幻魂**

欢迎参与讨论

32 条评论

默认 最新

Redux 状态更新为什么会从根结点开始全部重新渲染？只有订阅过状态更新的组件会重新跑一遍回调函数，跑完这个函数，组件的状态也未必改变，最终就是该渲染的渲染，不该渲染的不渲染。

准确的说应该是Redux状态每次更新，所有subscribe的回调都会跑一遍，这才是问题所在，整个系统只要有一个回调写得烂，整个系统都会被拖垮，典型的一粒老鼠屎害了一锅粥。

inner 上面用 react 原本的 state 就行
引入 redux 每次更新，从整个 App
开始渲染，每改动一次就会全部渲
染到爆。为啥改动页面局部的一个
要刷新整个网站甚至 layout 呢？再

2020-05-04

**Rabibabo** · 于江水在银河系

store是一个constant，它在react-redux中用context传给整个应用，但因为store其自身的引用并不会变化，所以不会触发context的变化而导致组件更新。

其实完全可以用`export const store = createStore()`来让整个应用可以import到。react-redux之所以用context来传，我估计还是为了简化API，否则connect函数和useSelector就要手动指定store了。

2020-05-05

**安兹** · 于江水在银河系

确实是根节点订阅，推送整个页面的connect组件，但是connect高阶组件会做性能优化，只有state改了才会更新

2020-05-05

**莫等闲**

一直在用mobx，好用，而且性能好。

2020-05-04

**黄cJ**

MobX 的文档入门比较困难...

2020-05-07

**PunCha Feng**

加的时候都说为了将来扩展，而且业界都在用，无力反驳

2020-05-04

**于江水在银河系** 作者

引入很简单，知道为什么引入却很难，需要针对实际情况衡量。就我那个 portal 的例子，将来是 container 横向扩展自己维护内部的 state 即可，甚至也可以在 container 级别加 redux 如果很复杂，不引入全局 redux 也并没有扩展问题。

2020-05-04

**幻魂**

欢迎参与讨论

32 条评论

默认 最新

Redux 状态更新为什么会从根结点开始全部重新渲染？只有订阅过状态更新的组件会重新跑一遍回调函数，跑完这个函数，组件的状态也未必改变，最终就是该渲染的渲染，不该渲染的不渲染。

准确的说应该是Redux状态每次更新，所有subscribe的回调都会跑一遍，这才是问题所在，整个系统只要有一个回调写得烂，整个系统都会被拖垮，典型的一粒老鼠屎害了一锅粥。

inner 上面用 react 原本的 state 就行
引入 redux 每次更新，从整个 App
开始渲染，每改动一次就会全部渲染
到爆。为啥改动页面局部的一个
要刷新整个网站甚至 layout 呢？再

2020-05-04

**Rabibabo** · 于江水在银河系

store是一个constant，它在react-redux中用context传给整个应用，但因为store其自身的引用并不会变化，所以不会触发context的变化而导致组件更新。

其实完全可以用`export const store = createStore()`来让整个应用可以import到。react-redux之所以用context来传，我估计还是为了简化API，否则connect函数和useSelector就要手动指定store了。

2020-05-05

**安兹** · 于江水在银河系

确实是根节点订阅，推送整个页面的connect组件，但是connect高阶组件会做性能优化，只有state改了才会更新

2020-05-05

**莫等闲**

一直在用mobx，好用，而且性能好。

2020-05-04

**黄cJ**

MobX 的文档入门比较困难...

2020-05-07

**PunCha Feng**

加的时候都说为了将来扩展，而且业界都在用，无力反驳

2020-05-04

**于江水在银河系** 作者

引入很简单，知道为什么引入却很难，需要针对实际情况衡量。就我那个 portal 的例子，将来是 container 横向扩展自己维护内部的 state 即可，甚至也可以在 container 级别加 redux 如果很复杂，不引入全局 redux 也并没有扩展问题。

2020-05-04

**幻魂**

欢迎参与讨论