链滴

React.StrictMode 会导致多次渲染

作者：zjhch123
原文链接：https://ld246.com/article/1590670561545
来源网站：链滴

React.StrictMode 会导致多次渲染

故事

今天在研究一个第三方库的时候发现一个很简单的组件在 render时被渲染了两次。精简后的组件代大致如下：

```
// App.js
import React, { useState } from 'react';

function App() {
  const [count] = useState(666);
  console.log('render!!');
  return (<div>{count}</div>);
}

export default App;


// index.js
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

在控制台中会看到输出了两次 render!!，使用的包版本如下：

```
{
  "react": "^16.13.1",
  "react-dom": "^16.13.1",
  "react-scripts": "3.4.1"
}
```

仔细看了 React的文档以及参考了很多Google上的问题，发现是因为使用了 React.StrictModerende导致的。

在开发环境中使用 StrictMode的目的是为了：

1. 识别不安全的生命周期
2. 检测过时的API和废弃的方法
3. 检测意外的副作用

React在检测意外的副作用时可能重复调用某些生命周期方法、hooks或者render方法，其包括：

1. class 组件的 constructor，render以及 shouldComponentUpdate方法

2. class 组件的生命周期方法 getDerivedStateFromProps
3. 函数组件
4. 状态更新函数 (即 setState的第一个参数)
5. 函数组件中 useState，useMemo或者 useReducer中的函数

当React发现在重复调用这些方法时出现了内存泄漏、无限循环或者其他奇怪的表现时会在控制台输错误，以供程序员快速定位错误。

在Github 的某条issue中，Dan Abramov也提到：

It is expected that setState updaters will run twice in strict mode in development. This helps e sure the code doesn’ t rely on them running a single time (which wouldn’ t be the case if an async render was aborted and alter restarted). If your setState updaters are pure functions (as hey should be) then this shouldn’ t affect the logic of your application.

简单来说就是我们在使用hooks或者在某些生命周期函数中不应该使用有副作用的代码。在开发模式的trictMode下，React会帮助我们发现这些不好的代码并给予提示。

StrictMode的这个重复调用的特性只使用于开发模式，在生产模式下不会触发多次调用。

案例

github/ReactTraining/react-media/modules/Media.js

89行使用了 useState并在 useState中调用 setUpMQLs用以注册事件监听器。由于在 StrictMode下seState中的函数会被执行2次（或多次），因此会导致某些事件监听器没有被cancel而一直存在，最由于内存泄漏而被React检测到，控制台会输出 Can't perform a React state update on an unmou ted component(issue#139)。

参考资料

1. Strict Mode
2. Double-Invoke of State Functions in React