

2.5 定时器

定时器对应用来说是最重要的部分之一，React VR 实现的是 [browser timers](#).

定时器

- `setTimeout, clearTimeout`
- `setInterval, clearInterval`
- `setImmediate, clearImmediate`
- `requestAnimationFrame, cancelAnimationFrame`

`requestAnimationFrame(fn)` 和 `setTimeout(fn, 0)` 不一样，前者在每一帧刷新之后执行一次，后者会

尽可能快的执行（在 iPhone5s 上超过 1000 次每秒）。

`setImmediate` 则会在当前 JS 代码块执行结束以后执行，也就是在发送批量响应到原生之前。注意如

果你在 `setImmediate` 的回调函数中又执行了 `setImmediate`，它会紧跟着执行，而不会再调用之前等待原生的代码。

`Promise` 就是用了 `setImmediate` 执行异步回调。

TimerMixin

我们发现 React VR 的最主要的系统性错误就是定时器相关的，也就是在组件卸载之后定时器依旧生

效，为了解决这个问题，我们引入了 `TimerMixin`。如果你在组件中引入 `TimerMixin`，就可以把你原

本的 `setTimeout(fn, 500)` 改为 `this.setTimeout(fn, 500)` (只需要在前面加上 `this.`)，然后当你的组件

卸载时，所有的计时器事件也会被正确的清除。

这个库并没有跟着 React VR 一起发布。需要在项目文件夹下输入

```
npm i react-timer-mixin --save
```

来单独安装它。

```
import TimerMixin from 'react-timer-mixin';
```

```
var Component = React.createClass({
  mixins: [TimerMixin],
  componentDidMount: function() {
    this.setTimeout(
      () => { console.log('I do not leak!'); },
      500
    );
  }
});
```

用这个库可以规避许多难以排查的 BUG

注意：如果你的 react 组件用 ES6，内部[没有 mixins 的 API](#)。

在 ES6 里面用 `TimerMixin`，我们推荐用 [react-mixin](#)