

5.3 AppState(应用状态)

`AppState` 可以告诉我们应用当前是在后台运行还是前台运行，并且能在状态改变的时候通知我们。

`AppState` 通常在处理推送通知的时候用来决定内容和对应的行为。

App States

- `AppState` - 应用正在前台运行
- `background` - 应用正在后台运行。用户有可能在别的应用中或者主屏幕中
- `inactive` - 一个过渡状态，不会再正常的 **React Native** 应用中出现

更多信息，请查看 [Apple's documentation](#)。

基本用法

如果要获得当前的状态，我们需要使用 `AppState.currentState`，这边变量会一直保持更新，不过在启动的过程中，`currentState` 可能为 `null`，直到 `AppState` 从原生代码得到通知为止

```
getInitialState: function() {  
  return{  
    currentAppState: AppState.currentState,  
  };  
};
```

```
},
componentDidMount: function() {
  AppState.addEventListener('change',
this._handleAppStateChange);
},
componentWillUnmount: function() {
  AppState.removeEventListener('change',
this._handleAppStateChange);
},
_handleAppStateChange: function(currentAppState) {
  this.setState({ currentAppState, });
},
render: function() {
  return (
    <Text>Current state is: {this.state.currentAppState}</Text>
  );
},
```

上面的例子只会显示"Current state is: active"，因为应用只有处于 active 状态下才能被用户看到，

而 null 状态只会在一开始的瞬间出现..

方法

constructor(0)

addEventListener(type, handler)

添加一个监听函数，用于监听应用状态的变化, type 参数为 change.

TODO: 现在 AppState 是 NativeEventEmitter 的一个子类，我们可以弃用 addEventListener 和

`removeEventListener` 或者值使用 `addListener` 和 `listener.remove()`。这是一个重大改变，虽然两个方法和事件的名称不同，这将是一个突破性的改变(`addListener` 事件目前是全局唯一的)。

`removeEventListener(type, handler)`

删除一个监听函数, **type** 参数为 `change`。