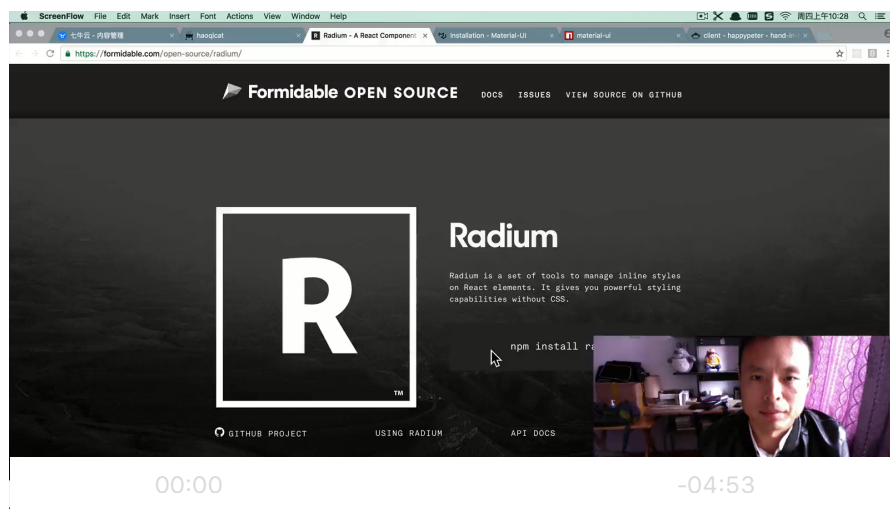


[首页](#) [目录](#)[退出](#)[认证方式对比](#)[JWT 知识科普](#)[API 调试工具](#)[实现用户认证接口](#)[实现基于 JWT 的用户认证接口](#)

## 借助 Radium 工具集，编写导航栏的样式

### 安装 radium 包

```
npm install --save radium
```

若采用内联样式编写 React 组件样式的时候，一些便捷的 CSS 特性就不能使用了，比如说 CSS 支持的伪选择器，媒体查询，还要手动给样式添加厂商前缀，这样看来，采用内联样式的确挺不方便的。不过，通过 **Radium** 工具集，就能让 React 组件的内联样式支持这些功能。

## 添加一个导航栏

新建文件 `src/ui/shared/Header.js`，添加代码：

```
import React, { Component } from 'react';
import Radium from 'radium';
import ActionHome from 'material-ui/svg-icons/';

class Header extends Component {
  render() {
    const styles = {
      header: {
        position: 'fixed',
        zIndex: '100',
        top: 0,
        right: 0,
        left: 0,
        backgroundColor: '#00bcd4',
        borderBottom: '1px solid #0079aa',
        height: '47px',
        paddingLeft: '16px',
        paddingRight: '16px',
        display: 'flex',
        alignItems: 'center',
        justifyContent: 'space-between'
      },
      nav: {
        color: 'white',
        opacity: '.8',
        fontWeight: '600',
        fontSize: '1em',
        textDecoration: 'none',
        ':hover': {
          cursor: 'pointer',
          textDecoration: 'underline'
        }
      }
    };
  }
}
```

```

    }
  }
}

return (
  <header style={styles.header}>
    <div>
      <a to="/" style={styles.nav} key='/'>
    </div>
    <div>
      <a to="/login" style={styles.nav} >
    </div>
  </header>
);
}
}

export default Radium(Header);

```

## 渲染导航栏

然后，修改 `src/ui/App.js` 文件，首先导入组件：

```

import Radium, { StyleRoot } from 'radium';
import Header from './shared/Header';

```

然后加载 `Header` 组件：

```

<StyleRoot>
  <div style={styles.root}>
    <Header />
  </div>
</StyleRoot>

```

Radium 的 `StyleRoot` 组件是用来支持组件内联样式的媒体查询功能的，导航栏组件并没有用到，后续新增的组件会用到内联样式的媒体查询功能。

最后，把 `App` 组件用 `Radium` 包裹起来：

```
export default Radium(App);
```

到浏览器中访问 `localhost:8080` 端口，会看到一个蓝色的导航栏。

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3