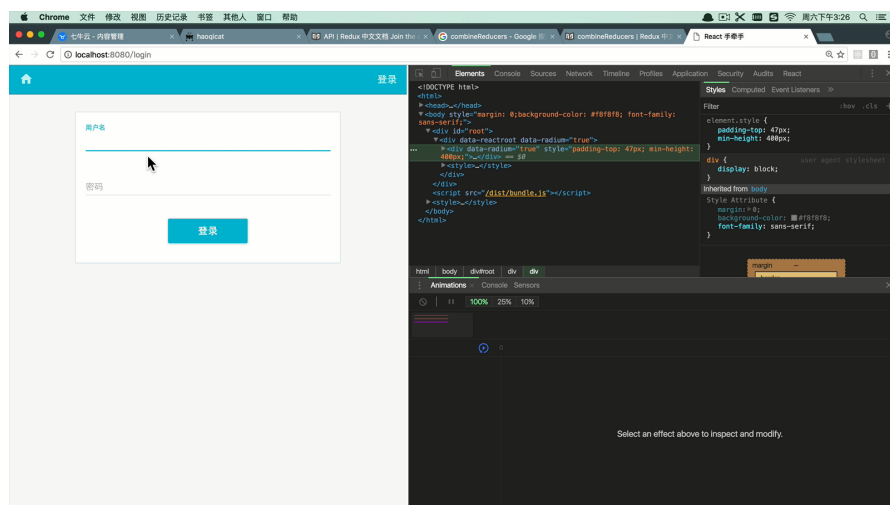


[首页](#) [目录](#)[退出](#)

引入 Material-UI 组件库美化页面

借助 Radium 工具集，编写导航栏的样式

通过 React Router 添加前端页面的路由功能

## 前端导航栏体现用户登录状态

打开 `src/shared/Header.js` 文件，导入 `connect` 方法：

```
import { connect } from 'react-redux';
```

借助 React 绑定库提供的 `connect` 方法读取 Redux store 中的 `auth state`，并注入到 `Header` 组件中。

检验传递给 `Header` 组件的 `auth` 属性是一个对象类型：

```
Header.propTypes = {  
  auth: React.PropTypes.object.isRequired
```

```
}
```

接下来，定义一个 `mapStateToProps` 方法，从 Redux store 中获得 `auth state`:

```
function mapStateToProps(state) {  
  return {  
    auth: state.auth  
  };  
}
```

把 `mapStateToProps` 方法作为参数传递给 `connect` 方法，得到 `auth state` 并以 `props` 的形式注入到 `Header` 组件中:

```
export default connect(mapStateToProps)(Radiu
```

## mapStateToProps

这样，我们就能够在 `Header` 组件中，使用传入的 `auth` 属性了:

```
render() {  
  const { isAuthenticated, currentUser } = this.props.auth  
  ...  
}
```

## 编写两个小组件

LogoutLink 组件体现用户登录后的导航栏状态,  
 LoginLink 组件体现用户退出登录后的导航栏状态:

```
render() {
  ...
  const LogoutLink = (
    <div>
      <span style={{color: 'rgb(255,226, 0)'},
      <Link to='/' style={styles.nav}>退出</L
    </div>
  );

  const LoginLink = (
    <div>
      <Link to='/login' style={styles.nav}>登
    </div>
  );
  ...
}
```

然后, 通过判断 isAuthenticated 的属性值, 若为 true, 说明用户已经登录了, 渲染 LogoutLink 组件; 反之, 渲染 LoginLink 组件, 最终 Header 组件的模样如下:

```
return (
  <header style={styles.header}>
    <div>
      <Link to='/' style={styles.nav}><Action
    </div>
    { isAuthenticated ? LogoutLink : LoginLink
  </header>
);
```

```
</header>  
);
```

欢迎添加 Peter 的微信: happypeter1983

冀ICP备15007992号-3