

react + redux + router总结

react

1. react是一个用于构建用户界面的 JavaScript 库
2. react的特点
 - 声明设计-----React采用声明范式，可以轻松描述应用
 - JSX，是一个 JavaScript 的语法扩展
 - 虚拟DOM节点，二次渲染React 只更新它需要更新的部分
 - 组件灵活复用
 - 单项响应的数据流-----React实现了单项响应式的数据流
3. 生命周期
 - 挂载阶段：`componentDidMount()` 挂载结束时被执行
 - 更新阶段：`componentDidUpdate` 更新结束时被执行
 - 卸载阶段：`componentWillUnmount` 卸载之前执行
4. Hook
 - Hook是 React 16.8 的新增特性。它可以让你在不编写 class 的情况下使用 state 以及其他的 React 特性。
 - `useState`：函数组件里调用它来给组件添加一些内部 state
 - `useEffect`：函数组件增加了操作副作用的能力
副作用（它跟 class 组件中的 `componentDidMount`、`componentDidUpdate` 和 `componentWillUnmount` 具有相同的用途，只不过被合并成了一个 API）
 - 自定义 Hook：将组件逻辑提取到可重用的函数
 - `useReducer`：形如`(state, action) => newState`，返回当前的state以及与其配套的dispatch方法

redux

由于单页面应用开发日渐复杂，需要管理的state（状态）越来越多，变化和异步也容易混淆在一起，redux是解决这些问题的一种方法。

redux的三大原则

- 单一数据源：整个应用的state被存储在一棵对象tree中，并且这个对象tree只存在于唯一一个store中
- state是只读的：唯一改变state的方法就是触发action，action是一个用于描述已发生事件的普通对象
- 使用纯函数来执行修改：编写reducers来描述action如何改变state tree

router

router是管理路由，所有页面的路径在router中配置，再由router分发，实现了自由的页面跳转。

1. 静态路由：运行之前就配置好的路由结构
2. 动态路由：应用运行的时候可以动态改变路由结构
3. 路由有两种方式，`BrowserRouter`和`HashRouter`

