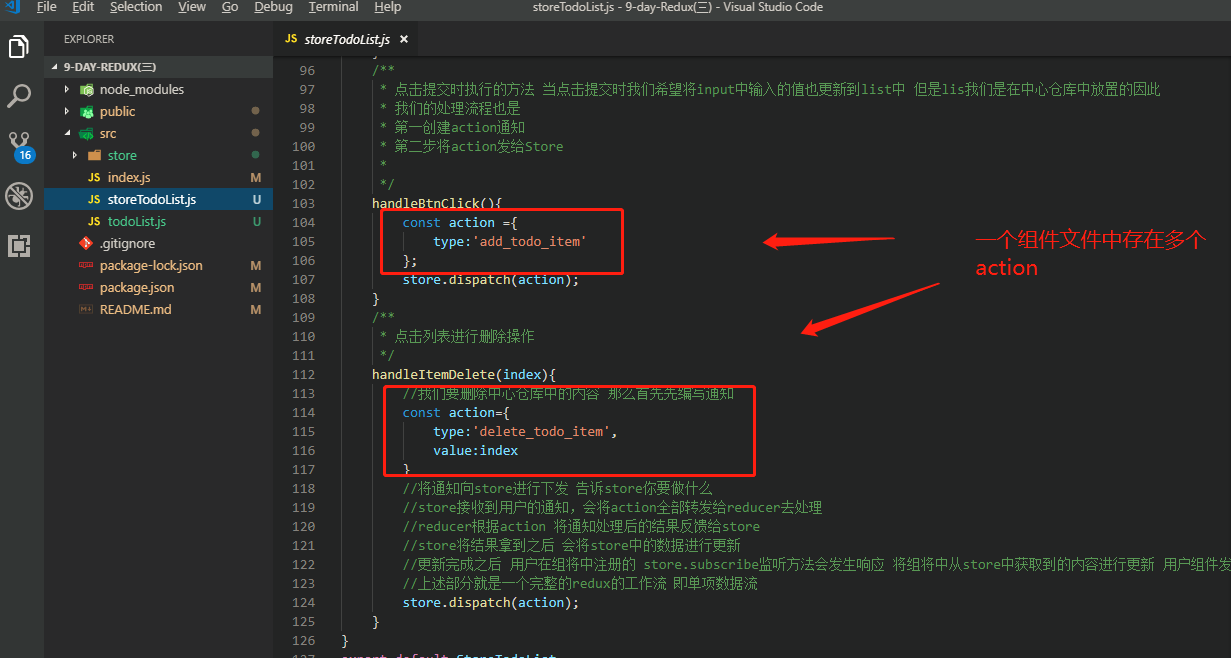
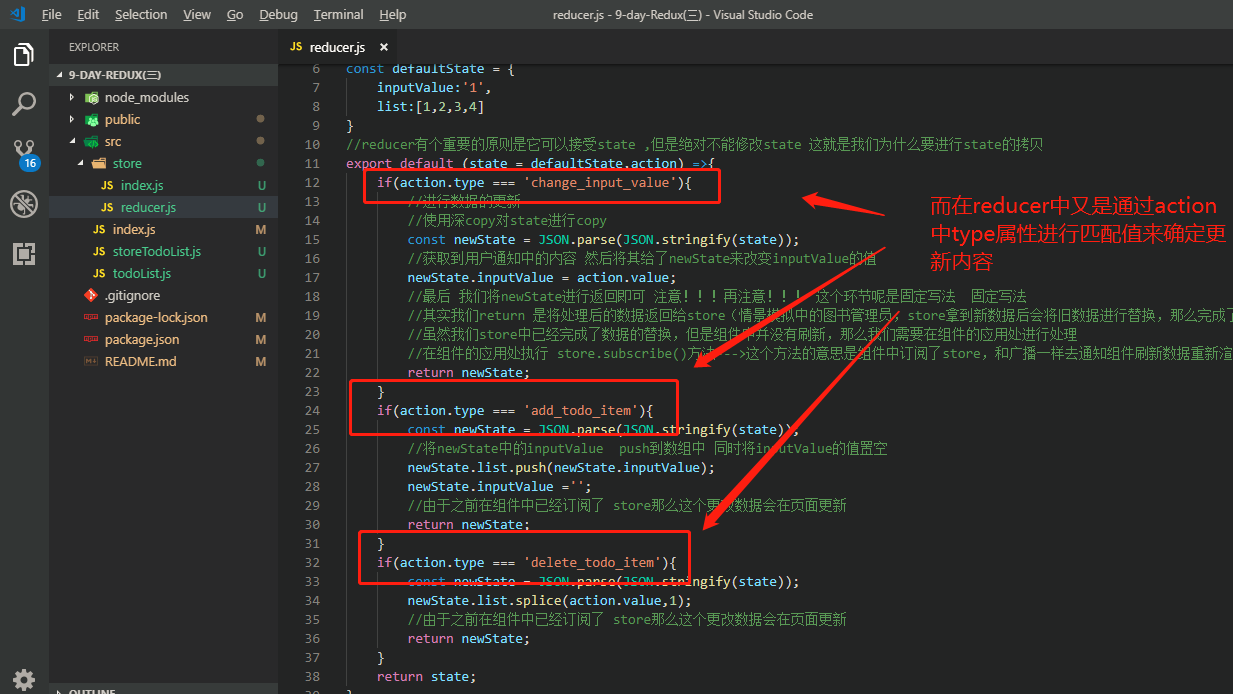
Redux---ActionTypes

1. actionTypes

从上一讲Redux的工作流程中我们得知，当要更改中心仓库中数据时，首先是要创建action通知，那么在一个组件中有多个action时稍有操作不对就会造成action中type属性配置错误，进而造成调试成本的提高。故此提出actionTypes的概念

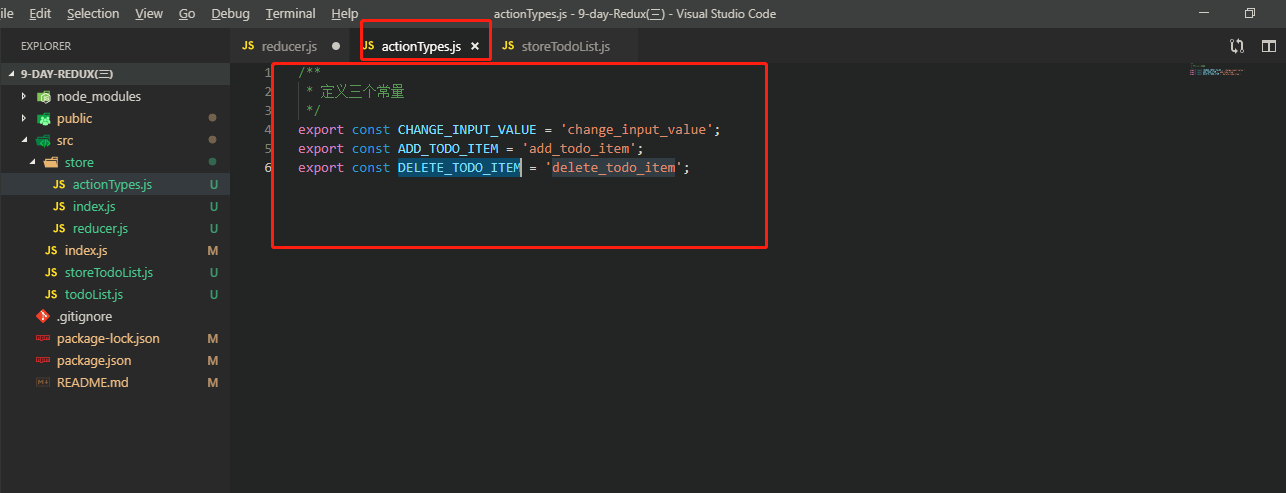




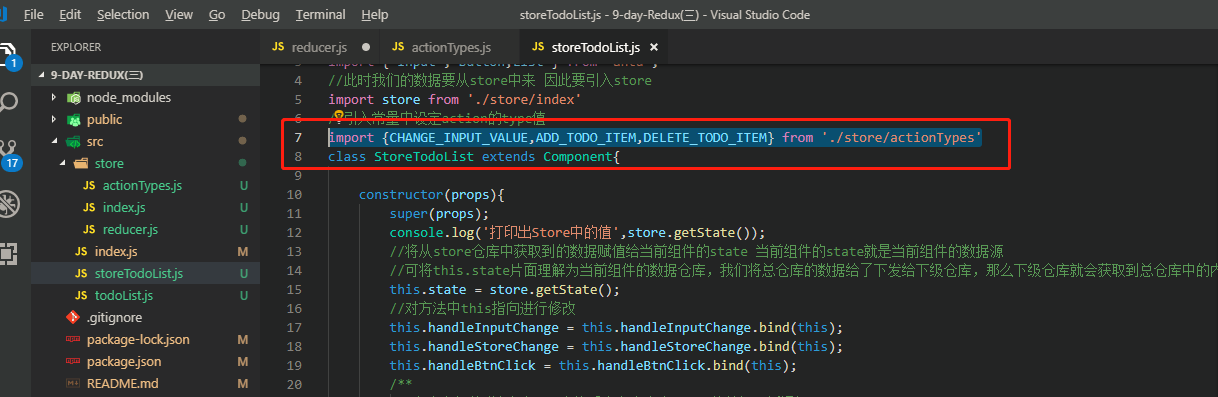
这样会导致维护成本很高，出现bug很难排错

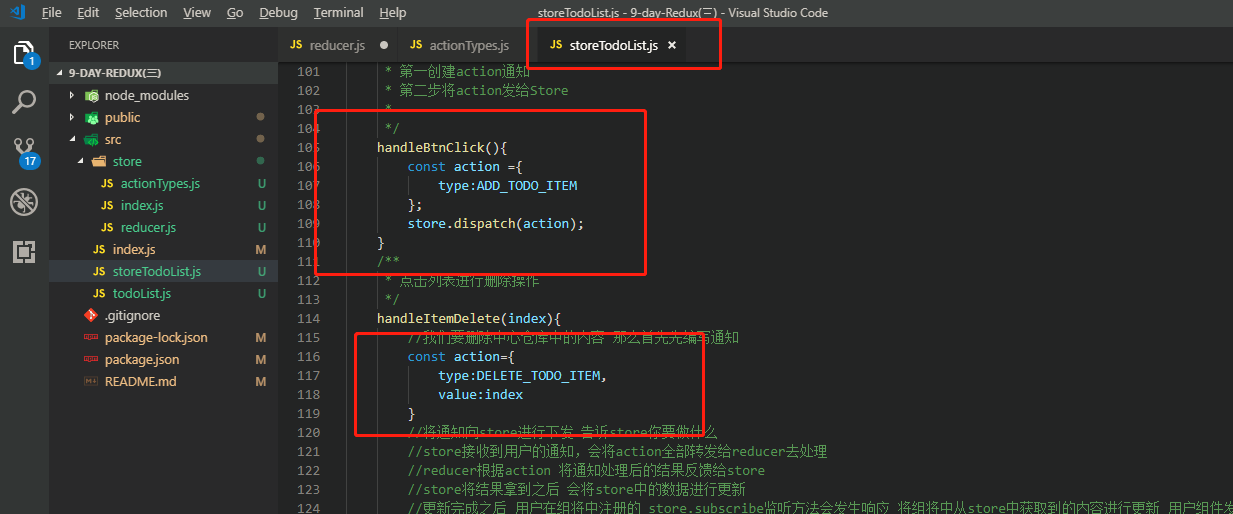
为了解决上面的困境，我们在store中创建actionTypes文件

1. 在actionTypes文件中声明action中需要的常量并且抛出export

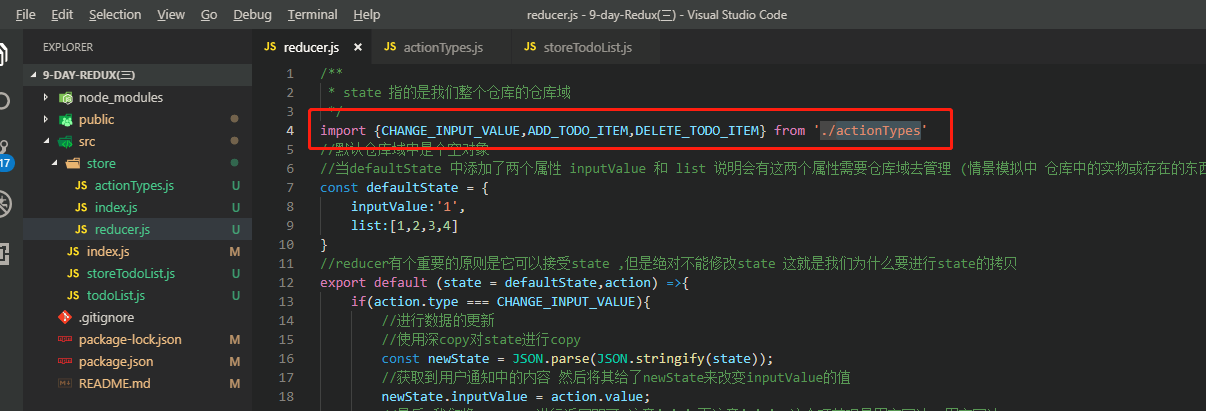


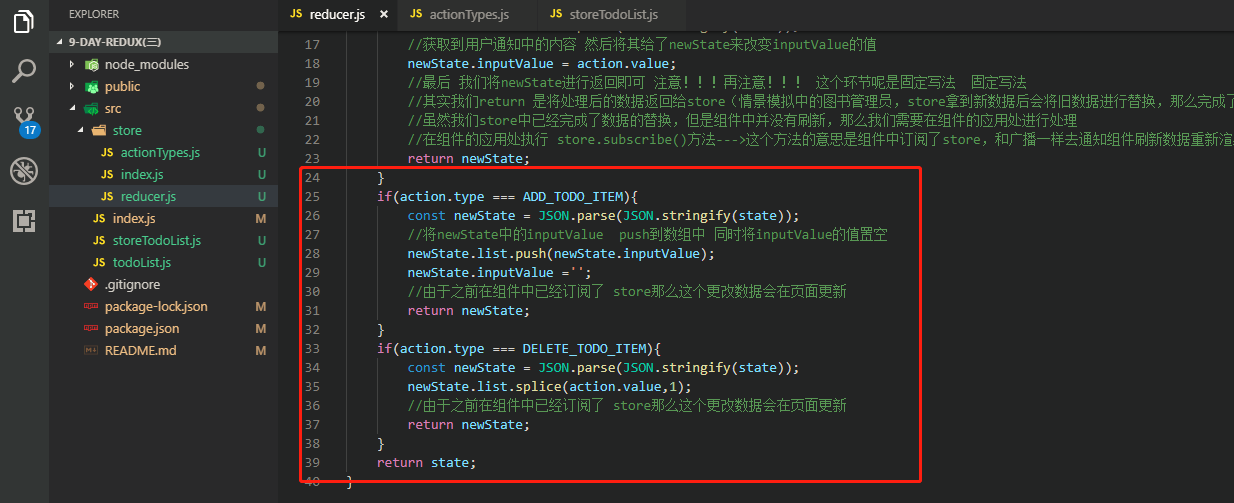
1. 在组件中创建通知时进行引用





1. 在reducer中利用action的type属性进行比对时引用常量

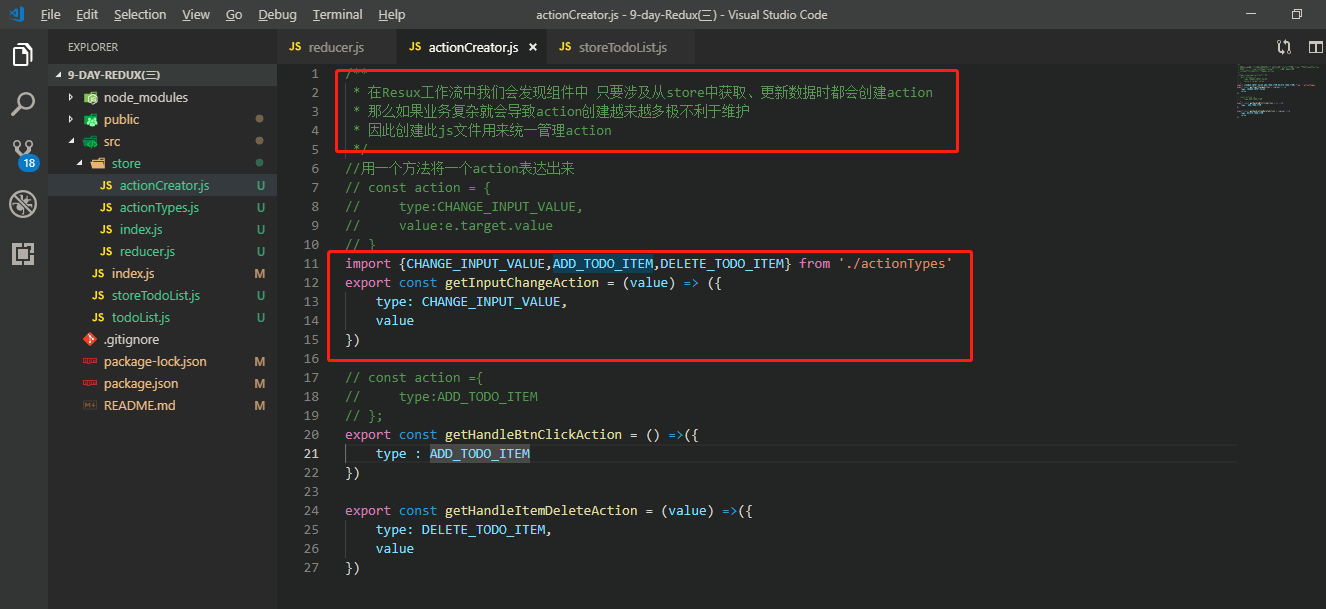




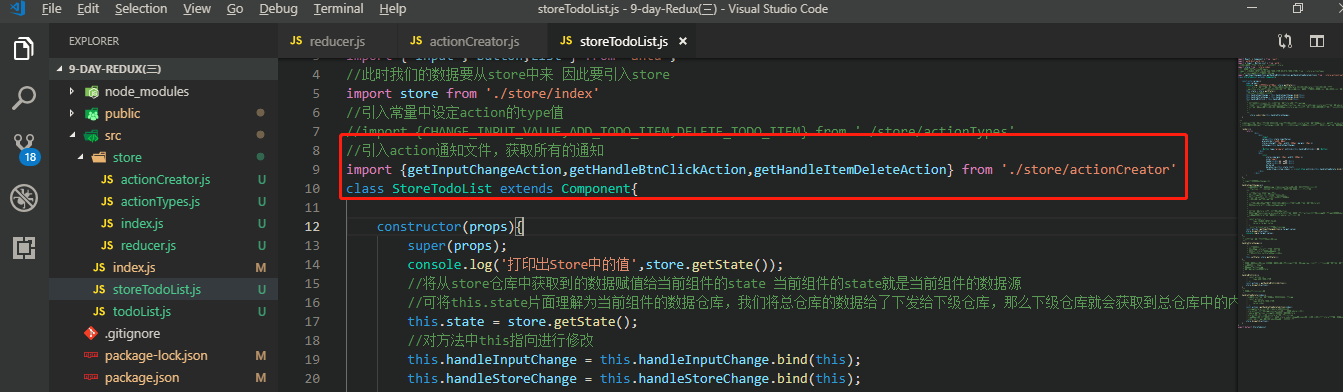
如此通过中间文件的引入会减少因为输入内容不匹配造成的bug

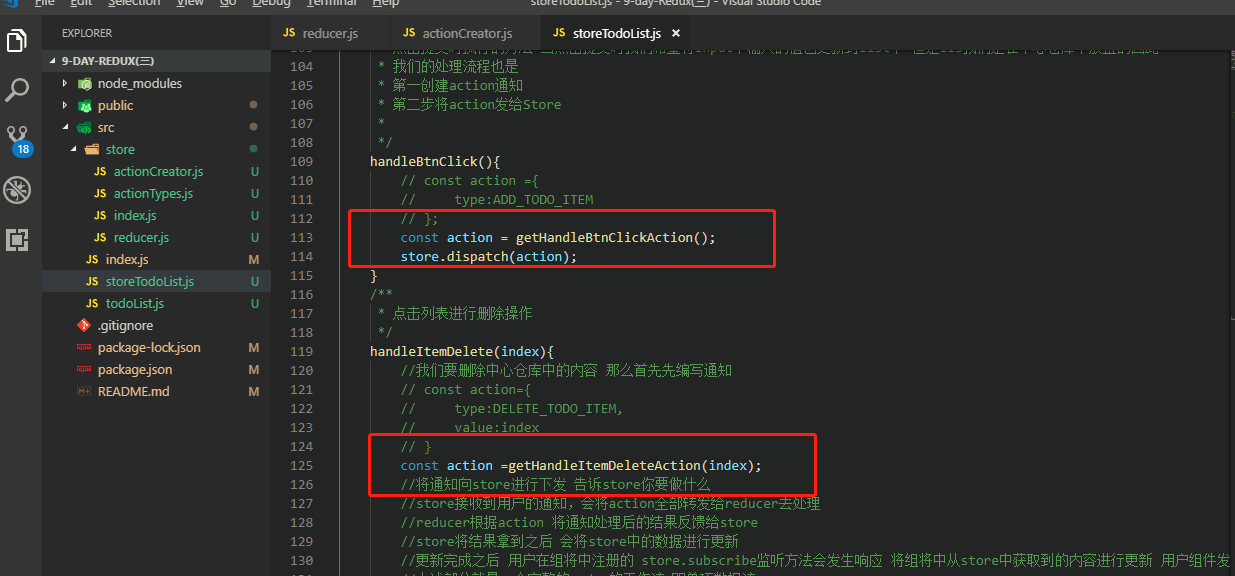
1. 通过actionCreator统一创建action

创建通知文件用来管控所有的通知



引入通知文件并使用通知文件中抛出的方法





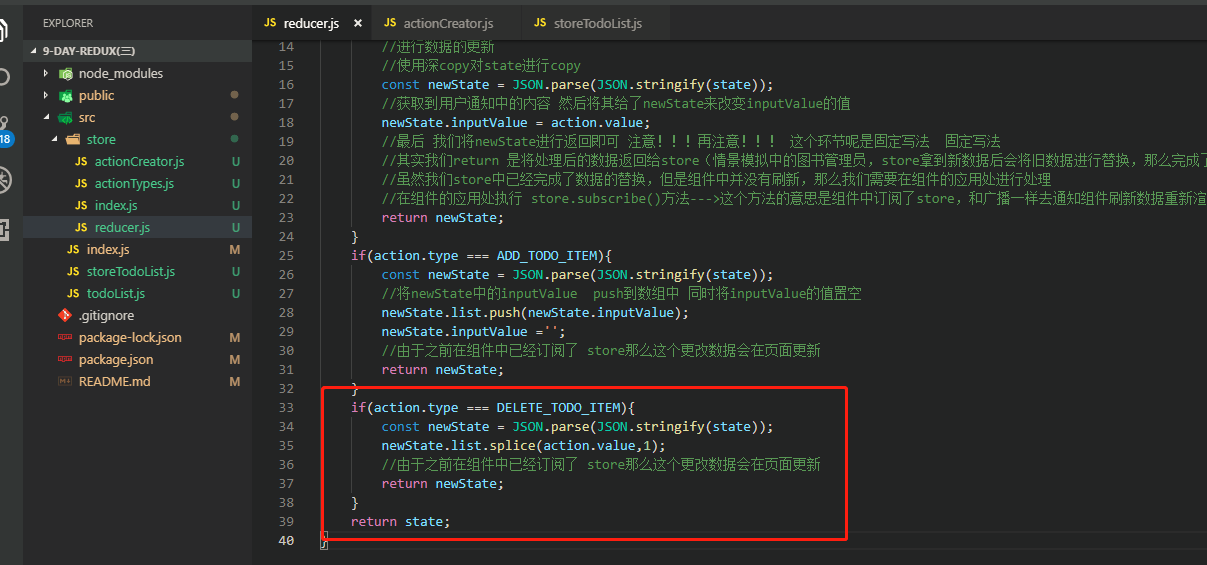
1. redux设计和使用的三个原则

整个应用之间我们只能创建一个store,就是代码中store文件下indexe.js文件

Store是唯一的

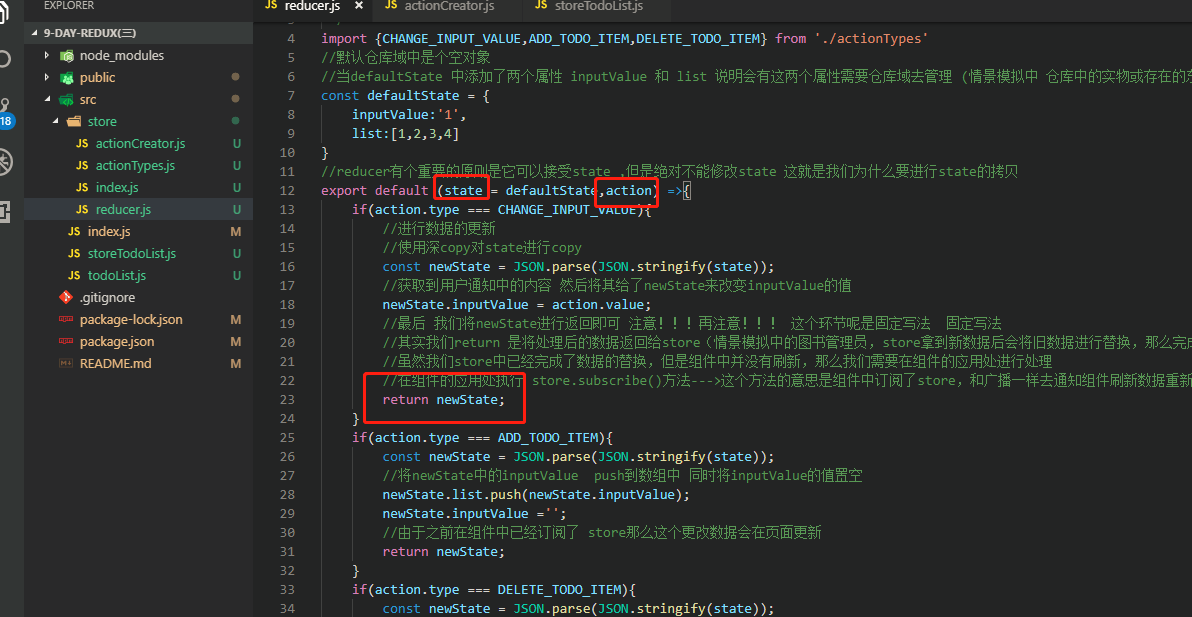
在代码实现中reducer并没有改变store,它只是将新的store返回给了Store

只有Store能够改变自己的内容



reducer必须是纯函数

纯函数指的是，给定固定的输入，就一定会有固定的输出，而且不会有任何副作用



1. redux的核心api

createStore--->创建store

store.dispatxh--->将创建的action传递或者派发给store

store.getState--->帮助获取store中的所有数据类型

store.subscribe--->订阅store的改变去执行的内容