Vuex

<https://segmentfault.com/a/1190000009404727>

<https://segmentfault.com/a/1190000012015742>

<https://vuex.vuejs.org/zh/guide/>

1.



2.action:



3.不懂：

https://vuex.vuejs.org/zh/guide/actions.html#%E5%9C%A8%E7%BB%84%E4%BB%B6%E4%B8%AD%E5%88%86%E5%8F%91-action



5.不懂：



6. 单状态树和模块化并不冲突——在后面的章节里我们会讨论如何将状态和状态变更事件分布到各个子模块中

7. 每当 store.state.count 变化的时候, 都会重新求取计算属性，并且触发更新相关联的 DOM。

8. 通过在根实例中注册 store 选项，该 store 实例会注入到根组件下的所有子组件中，且子组件能通过 this.$store 访问到

9. mapState 辅助函数

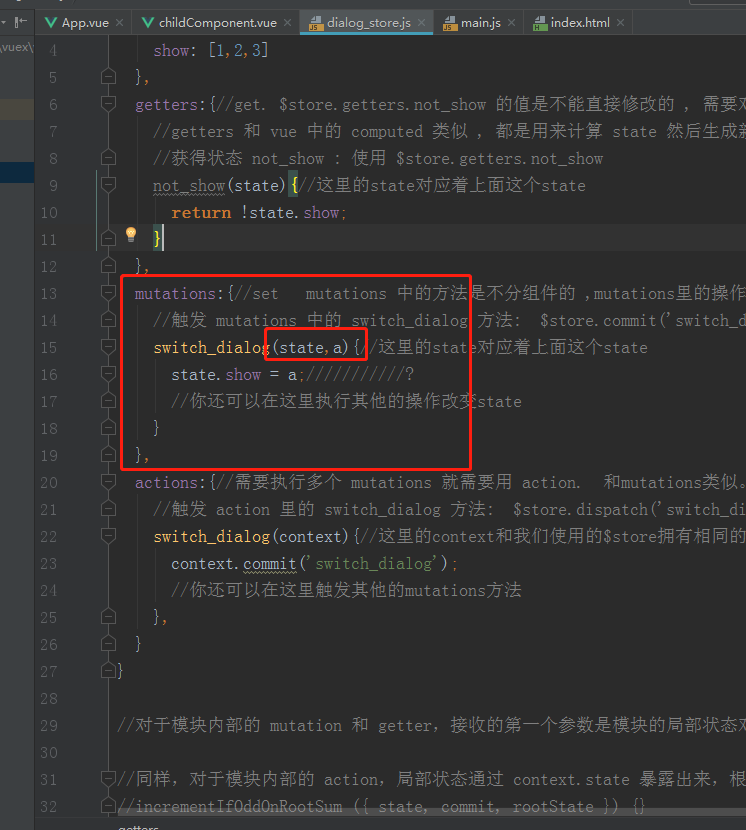
// 为了能够使用 `this` 获取局部状态，必须使用常规函数

10. Vuex Store是响应式的，当Vue组件从store中读取状态（state选项）时，若store中的状态发生更新时，他会及时的响应给其他的组件（类似双向数据绑定） 而且不能直接改变store的状态，改变状态的唯一方法就是，显式地提交更改（mutations选项）

11. 把getters看成是store的计算属性

12.要传参的话这里要加第二个参数：

<https://www.cnblogs.com/first-time/p/6815036.html>



13.action里如何嵌套触发action?

14.总结：

　　　　各个类型的 API各司其职，mutation 只管存，你给我（dispatch）我就存；action只管中间处理，处理完我就给你，你怎么存我不管；Getter 我只管取，我不改的。　action放在了 methods 里面，说明我们应该把它当成函数来用（讲道理，钩子函数也应该可以的） mutation是写在store里面的，这说明，它就是个半成品，中间量，我们不应该在外面去操作它。getter写在了 computed 里面，这说明虽然 getter我们写的是函数，但是我们应该把它当成计算属性来用。