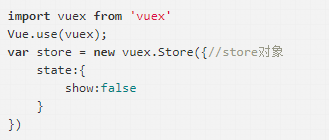
**安装、使用vuex**

1、**安装**

在命令行窗口输入：npm install vuex --save

2、**基本用法**

在main.js中加入



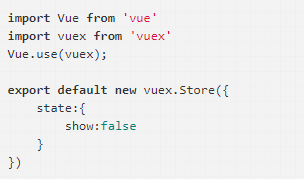
然后，在实例化的vue对象时加入store对象：



完成到这一步，上述例子中$store.state.show就可以使用了。

3、**modules**

为了便于日后的维护，通常我们不会将store.js写到main.js中，我们可以在src目录下新建一个名为store的文件夹，然后在里面新建一个index.js



所以相应的，main.js里的代码应该改为

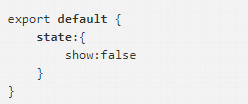


这样就把 store 分离出去了 , 那么还有一个问题是 : 这里 $store.state.show 无论哪个组件都可以使用 , 那组件多了之后 , 状态也多了 , 这么多状态都堆在 store 文件夹下的 index.js 不好维护怎么办 ?

我们可以使用 vuex 的 modules , 把 store 文件夹下的 index.js 改成 :

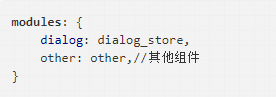


这里我们引用了一个 dialog\_store.js , 在这个 js 文件里我们就可以单独写 dialog 组件的状态了 :



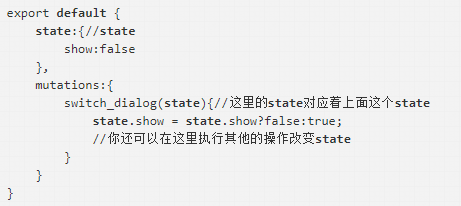
做出这样的修改之后 , 我们将之前我们使用的 $store.state.show 统统改为 $store.state.dialog.show 即可。

如果还有其他的组件需要使用 vuex , 就新建一个对应的状态文件 , 然后将他们加入 store 文件夹下的 index.js 文件中的 modules 中。



**mutations：**

mutations是改变store中状态的

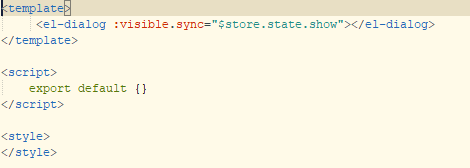


使用 mutations 后 , 原先我们的父组件可以改为 :



使用 $store.commit('switch\_dialog') 来触发 mutations 中的 switch\_dialog 方法。

在components文件夹下创建了一个dialog.vue文件，只是一个组件，没有其他的意思：



在这里需要**注意：**

①mutations 中的方法是不分组件的 , 假如你在 dialog\_stroe.js 文件中的定义了

switch\_dialog 方法 , 在其他文件中的一个 switch\_dialog 方法 , 那么

$store.commit('switch\_dialog') 会执行所有的 switch\_dialog 方法。

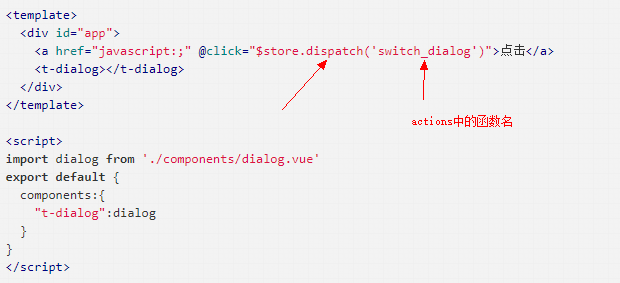
②mutations里的操作必须是同步的。

**actions（执行异步操作）:**

多个 state 的操作 , 使用 mutations 会来触发会比较好维护 , 那么需要执行多个 mutations 就需要用 action 了:



那么 , 在之前的父组件中 , 我们需要做修改 , 来触发 action 里的 switch\_dialog 方法:

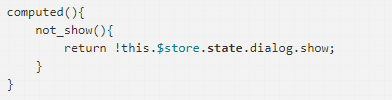


使用 $store.dispatch('switch\_dialog') 来触发 action 中的 switch\_dialog 方法。

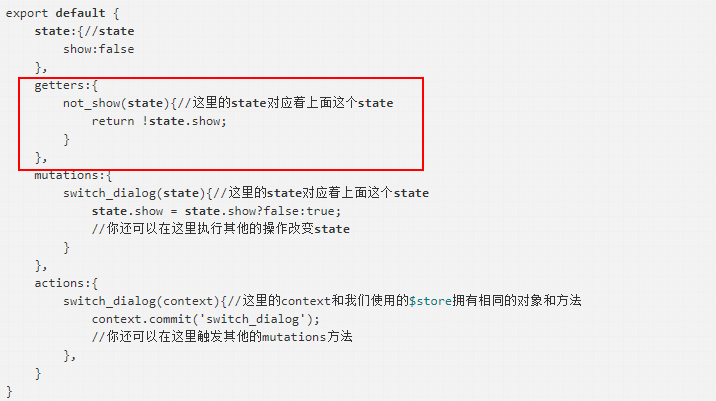
**getters:**

getters 和 vue 中的 computed 类似 , 都是用来计算 state 然后生成新的数据 ( 状态 ) 的。

还是前面的例子 , 假如我们需要一个与状态 show 刚好相反的状态 , 使用 vue 中的 computed 可以这样算出来 :



那么 , 如果很多很多个组件中都需要用到这个与 show 刚好相反的状态 , 那么我们需要写很多很多个 not\_show , 使用 getters 就可以解决这种问题 :



我们在组件中使用 $store.state.dialog.show 来获得状态 show , 类似的 , 我们可以使用 $store.getters.not\_show 来获得状态 not\_show 。

注意 : $store.getters.not\_show 的值是不能直接修改的 , 需要对应的 state 发生变化才能修改。

mapState、mapGetters、mapActions我们后面来说。

秀丽，我爱你，么么哒！！！