# [vuex action 与mutations 的区别](https://www.cnblogs.com/wupeng88/p/8931728.html)

面试没说清楚。这个太丢人回来整理下：

事实上在 vuex 里面 actions 只是一个架构性的概念，并不是必须的，说到底只是一个函数，你在里面想干嘛都可以，只要最后触发 mutation 就行。异步竞态怎么处理那是用户自己的事情。

|  |  |
| --- | --- |
| 1  2  3  4  5 | Vuex.Store({    state,    actions,    mutation  }); |

|  |  |
| --- | --- |
| 1 | vuex 真正限制你的只有 mutation 必须是同步的这一点,在vue中，只有mutation才能正真改变VUEX stroe中的state, |

　Action 提交的是 mutation，而不是直接变更状态。

   Action 可以包含任意异步操作。 个人觉得这个action的 产生就是 因为mutation 不能进行异步操作，如果有异步操作那么就用action 来提交mutation

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | Mutation 必须是同步函数  一条重要的原则就是要记住 mutation 必须是同步函数。为什么？请参考下面的例子：    mutations: {    someMutation (state) {      api.callAsyncMethod(() => {        state.count++      })    }  }  现在想象，我们正在 debug 一个 app 并且观察 devtool 中的 mutation 日志。每一条 mutation 被记录，devtools 都需要捕捉到前一状态和后一状态的快照。<br>然而，在上面的例子中 mutation 中的异步函数中的回调让这不可能完成：因为当 mutation 触发的时候，回调函数还没有被调用，<br>devtools 不知道什么时候回调函数实际上被调用——实质上任何在回调函数中进行的状态的改变都是不可追踪的。 |

### **在组件中提交 Mutation**

你可以在组件中使用 this.$store.commit('xxx') 提交 mutation，或者使用 mapMutations 辅助函数将组件中的 methods 映射为 store.commit 调用（需要在根节点注入 store）。

[IMG_256](https://www.cnblogs.com/wupeng88/p/javascript:void(0);)

import { mapMutations } from 'vuex'

export default {

// ... methods: {

...mapMutations([

'increment', // 将 `this.increment()` 映射为 `this.$store.commit('increment')`

// `mapMutations` 也支持载荷：

'incrementBy' // 将 `this.incrementBy(amount)` 映射为 `this.$store.commit('incrementBy', amount)` ]),

...mapMutations({

add: 'increment' // 将 `this.add()` 映射为 `this.$store.commit('increment')` })

}

}

[IMG_257](https://www.cnblogs.com/wupeng88/p/javascript:void(0);)

如果想要获取对应的状态你就可以直接使用this.$store.state获取，当然，也可以利用vuex提供的mapState辅助函数将state映射到计算属性中去，如.

|  |  |
| --- | --- |
| 1 | const store = new Vuex.Store({ state: { count: 1 }, mutations: { increment (state) { // 变更状态 state.count++ } } }) |

Action去comitt一个mutation。它要指定去commit哪一个mutation，然后指定结束之后要做什么什么事情就要给出一个函数，所以说mutation的构成有两点名称和函数。

[IMG_258](https://www.cnblogs.com/wupeng88/p/javascript:void(0);)

const store = new Vuex.Store({

state: {

count: 0

},

mutations: {

increment (state) {

state.count++

}

},

actions: {

increment (context) {

context.commit('increment')

}

}

})

[IMG_259](https://www.cnblogs.com/wupeng88/p/javascript:void(0);)

action 异步操作

[IMG_260](https://www.cnblogs.com/wupeng88/p/javascript:void(0);)

actions: {

incrementAsync ({ commit }) {

setTimeout(() => {

commit('increment')

}, 1000)

}

}