



链滴

Vuex -- 状态管理器

作者: [cdq](#)

原文链接: <https://ld246.com/article/1636705338179>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

- 什么是 Vuex

- Vuex 是专门为 Vue 开发的一个状态管理模式。它采用集中式存储管理应用的所有组件的状态并以相应的规则保证状态以一种可预测的方式发生变化。

- 上面是官方的说法，我自己的理解就是：Vuex 是一个状态存储器，可以将自己 Vue 开发中的数据存入，然后在整个 Vue 项目任何地方都能够获取到该数据。

- Vuex 中的五大概念

- State 池（数据源）
 - Getters 查（自定义状态）
 - Mutation 改（真正修改状态的动作）
 - Action 触发修改行为（可异步）
 - Module 可同时拥有多个数据源（池）

- 导出 store

每一个 Vuex 应用的核心就是 store(仓库)。store 基本上就是一个容器，包含着应用中大部分状态 (state)。在 Vuex 中状态是响应式的，当 store 中的状态发生改变时，使用改状态的组件的值也会相的更新。改变状态的办法是显示的提交 (commit) Mutation (变化)，这样能使我们更好的追踪每个状态的变化。

```
import Vuex from 'Vuex'

const store = new Vuex.store({

  state:{
    age: 18
  },//该store存储状态的地方
  mutations:{
    addAge(state){
      store.age++
    }
  },//修改数据的方法
  actions:{
    addAge({commit}){
      commit('addAge')
    }
  }
},//提交mutation
})
```

export default store

1. 获取数据源的数据

第一种方法：

```
beforeCreate(){
  console.log(this.$store.state.age)
}
```

2. 第二种方法：

定义 store 下的 getters

```
getters: {
  getAge(state){
    return state.age + '岁'
  }
}
```

使用如下:

```
computed: {
  age(){
    return this.$store.getters.getAge
  }
}
```

- 改变数据源的数据

先定义 mutations 中的方法和 actions 中的方法:

```
mutations:{
  subAge(state){
    state.age--
  }
},
actions:{
  subAge({commit}){
    commit('subAge')
  }
}
```

再从组件中调用:

```
methods:{
  ...mapActions([
    'subAge'//将 'this.subAge()' 映射为 'this.$store.dispatch('subAge')'
  ]),
  sub(){
    this.subAge()
  }
}
```

上面是使用了 actions 调用的 mutations 中的 subAge() 方法, 好处是可以进行异步操作。

下面的是不适用 actions, 直接调用 mutations:

```
methods:{
  ...mapMutations: ([
    'subAge'//将 'this.subAge()' 映射成 'this.$store.commit('subAge')'
  ]),
  sub(){
    this.subAge()
  }
}
```