Module

2019年11月4日

13:40

模块内部的局部状态:

1.state

state 为嵌套状态

在组件中的应用

...mapState({

      countA: state => state.a.count,//A模块

      count: state => state.count//根节点

    })

2.mutations

mutations: {  
 increment (state) {  
 // 这里的 `state` 对象是模块的局部状态  
 state.count++  
 }  
 },

3.getters

getters: {  
 doubleCount (state) {  
 return state.count \* 2  
 }  
 }

//根节点状态可以作为第三个参数(rootState)传递进来

getters: {  
 sumWithRootCount (state, getters, rootState) {  
 return state.count + rootState.count  
 }  
 }

4.actions

//state(context.state)暴露局部状态,rootState(context.rootState)暴露根节点状态

actions: {  
 incrementIfOddOnRootSum ({ state, commit, rootState }) {  
 if ((state.count + rootState.count) % 2 === 1) {  
 commit('increment')  
 }  
 }  
 }

5.命名空间

默认情况下，模块内部的 action、mutation 和 getter 是注册在全局命名空间的.为了提高模块封装度和复用性，可以通过添加 namespaced: true 的方式使其成为带命名空间的模块。当模块被注册后，它的所有 getter、action 及 mutation 都会自动根据模块注册的路径调整命名。

**命名空间使用后的使用**

1. **state**

命名空间的使用对state没有影响

1. **直接引用:**

this.$store.state.a.count

1. **绑定函数:**

  ...mapState({

      countA: state => state.a.count,

      count: state => state.count

    })

1. **mutations**
   1. **直接引用:**

this.$store.commit('a/addAge')

1. **绑定函数:**

    <button @click="this['a/increament']">-</button>

    <button @click="increament">+</button>

    ...mapMutations(["increament"]),

    ...mapMutations(["a/increament"]),

1. **getters**
   1. **模块的书写**

// 在这个模块的 getter 中，`getters` 被局部化了  
 // 你可以使用 getter 的第四个参数来调用 `rootGetters`  
 someGetter (state, getters, rootState, rootGetters)

1. **直接引用**

<h3>{{this.$store.getters['a/sum']}}</h3>

1. **绑定函数**

<h3>{{this['a/sum']}}</h3>

 ...mapGetters(["a/sum"])

1. **actions**
   1. **模块的书写:**

// 在这个模块中， dispatch 和 commit 也被局部化了  
 // 他们可以接受 `root` 属性以访问根 dispatch 或 commit

someAction ({dispatch, commit, getters, rootGetters }){

dispatch('someOtherAction', null, { root: true })

dispatch('someOtherAction')

commit('someMutation') // -> 'foo/someMutation'  
commit('someMutation', null, { root: true }) // -> 'someMutation'

}

1. **全局action注册:**

actions: {

addAge: {

root: true,

handler: (context) => {

setTimeout(() => {

context.commit('addAge')

}, 3000);

}

}

}

1. **直接引用**

this.$store.dispatch('a/addAge')

1. **绑定函数**

<button @click="this['a/increamentIfOddOnRootSum']">+</button>

methods: {  
 ...mapActions(['a/increamentIfOddOnRootSum'])

])

1. **或者:**

 <button @click="increamentIfOddOnRootSum()">-</button>

 ...mapActions('a',[

        "increamentIfOddOnRootSum"

        ]),

!!!!注意: state和getters的绑定函数要写在computed属性中,

actions 和 mutation 的绑定函数要写在methods属性中