

## 25.Module 模块化

学习要点：

### 1.Module 设置

本节课我们来开始了解 Vuex 中 Module 的设置。

#### 一. Module 设置

1. 由于 `state` 使用的单一状态树，所有状态都集中在一个对象中，大了就很重；
2. Vuex 提供了 Module 模块化设置，可以将状态分割成一个个小的模块；
3. 注意：为了解决各种场景问题，手册上知识点比较繁杂，这里我们只入个门；
4. 在 `store` 目录下创建一个新的 js 文件：`module/list.js`，代码如下：

```
export default {
  state : {
    name : 'no name'
  },

  getters : {
    getName(state) {
      return '【' + state.name + '】'
    }
  },

  mutations : {
    setName(state, value) {
      state.name = value
    }
  },

  actions : {
    setName({commit}) {
      setTimeout(() => {
        commit('setName', 'Mr.Wang')
      }, 1000)
    }
  }
}
```

5. 在 `store/index.js` 中的 Module 引入这个模块；

```
import list from "@/store/module/list"
//模块
modules: {
```

```
    list
  }
```

PS: 为了测试模块状态和主状态冲突，故意设置相同的状态名称和其它；

6. 为了隔离主状态和模块状态名称冲突，需要设置一个命名空间；

```
export default {
  namespaced : true,
}
```

7. 调用方式，可以通过查看 `console.log(this.$store)` 来查看；

```
<template>
  <div class="list">
    <h3>{{ $store.state.list.name }}</h3>
    <h3>{{ getName }}</h3>
    <h3>{{ $store.state.name }}</h3>
    <button @click="setName">获取+</button>
  </div>
</template>

<script>
  import { mapActions } from 'vuex'

  export default {
    name: "List",

    computed : {
      getName() {
        return this.$store.getters['list/getName']
      }
    },

    methods : {
      // setName() {
      //   this.$store.dispatch('list/setName')
      // }
      ...mapActions({
        setName : 'list/setName'
      })
    }
  }
</script>
```