**18.State 状态设置**

学习要点：

1.State 设置

本节课我们来开始了解 Vuex 中 state 状态设置的方法。**一．State 设置**

# 首先，抛出问题：如果我们要设置多个状态在赋值取值就会比较繁琐；

1. 概念：Vuex 是单一状态树，一个对象包含全部状态，每个应用仅一个 store 实例；
2. 现在，给 state 增加三个共享状态，并设置一个需要修改的数据；

//状态值：store/index.js state: {

count : **0**,

name : 'Mr.Lee', age : **100**, gender : '男'

},

//修改状态mutations: {

setAge(state, value) { state.age = value

}

},

//如果不想直接用繁琐的插值，可以先用计算属性：Person.vue

computed : {

name() {

return this.$store.state.name

},

gender() {

return this.$store.state.gender

},

age : {

get() {

return this.$store.state.age

},

set(value) {

this.$store.commit('setAge', value)

}

}

}

<div class="person">

<ul>

<li>姓名：{{name}}</li>

<li>性别：{{gender}}</li>

<li>年龄：{{age}}</li>

</ul>

{{test}}

<input type="text" v-model="age">

</div>

# 上面是比较常规的解决方案，而共享状态变多之后，重复代码量会增多；

1. 在 Vuex 中提供了一个 mapState 辅组函数，我们可以使用它来简化操作；
2. 我们在需要使用这个辅组函数的地方导入即可；

//导出非默认的一个辅组函数，需要大括号，且名称固定

import { ***mapState*** } from 'vuex'

//只是显示的话，且状态名和模版插值名一致，用数组computed : ***mapState***(['name', 'gender', 'age']),

//v-model 因为丢失计算属性set,get，改成事件

<input type="text" @input="setAge" :value="age"> methods : {

setAge(e) {

this.$store.commit('setAge', e.target.value)

}

}

1. 如果状态名和模版插值名不一样，或者在插值显示的值还需要进一步处理的；

computed : mapState({

//箭头函数 state 是参数，状态对象，可改

//第一个 name 是模版显示的插值

//第二个 name 是状态对象的属性

name : state => state.name,

//简写版，同上 'gender'等于 state => state.gender gender : 'gender',

//更多逻辑操作的方法age(state) {

//return state.age + this.count return state.age

}

}),

PS：上面的方法又写死了，导致普通的计算属性无法使用，需要对象展开运算符；

computed : {

test() {

return 'test'

},

...mapState({

...

})

},