import Vue from 'vue'

import Vuex from 'vuex'

import { getGoods } from '../api/api'

Vue.use(Vuex)

/\* 创建仓库 \*/

const store = new Vuex.Store({

//状态

state: {

name: '张三',

age: 30,

num: 0,

list: [],

stus: [{id: '1', age: 20},{id: '2', age: 16},{id: '3', age: 11},{id: '4', age: 33}]

},

//同步通知：改变状态的唯一方式

mutations: {

//参数1： state本身， 参数2：本次提交的通知对象

changeName(state, obj){

//每个通知函数中，都会收到state，这个state就是状态对象本身

state.name = obj.newName

},

changNum(state, obj){

state.num += obj.num

},

initlist(state, obj){

state.list = obj.list

}

},

//异步通知：如果本次数据改变有异步操作，则需要写到action中

actions: {

async initlist(context){

//1. 先执行完所有异步操作！！！

let data = await getGoods()

//2. 拿到数据后，commit提交mutation让mutation去改变state

context.commit({

type: 'initlist',

list: data.data

})

}

},

getters: {

//获取年满18岁的学生

getStus(state){

return state.stus.filter(obj => obj.age >= 18)

}

}

})

//发出mutation通知

//方式一：直接传入要触发的函数名

// store.commit('changeName')

//方式二：使用对象的方式触发

// store.commit({

// type: 'changeName', //type就是要触发的函数名

// newName: '班长'

// })

// store.commit({

// type: 'changNum',

// num: -1

// })

//发出action通知

store.dispatch('initlist')

//获取getters的数据，访问方式同computed，不需要写()

console.log(store.getters.getStus)

// //取状态

// console.log(store.state.name)

export default store