



vuex 使用详解

一、什么是vuex

vuex是一个专门为vue.js设计的集中式状态管理架构。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。

二、vuex使用场景

vuex主要是是做数据交互，父子组件传值可以很容易办到，但是兄弟组件间传值（兄弟组件下又有父子组件），或者大型spa单页面框架项目，使用vuex可以简化数据交互。

场景说明：

- 由于vuex中的state存放的数据在页面刷新时会丢失，vuex只能用于单个页面中不同组件（例如兄弟组件）的数据流通

三、Vuex核心概念

- 1、store：类似容器，包含应用的大部分状态
 - 一个页面只能有一个容器
 - 状态存储是响应式的
 - 不能直接改变store中的状态，唯一途径显示地提交mutations
 - 在actions里面，也不能直接更改state里面的状态值，必须先定义一个mutations，然后在actions里面commit这个mutations，从而来更改state的状态值；
 - 如果要再次请求异步，那么就是dispatch一个actions
- 2、State：包含所有应用级别状态的对象
- 3、Getters：在组件内部获取store中状态的函数，类似组件的计算属性computed
- 4、Mutations：唯一修改状态的事件回调函数，默认是同步的，如果要异步就使用Actions
- 5、Actions：包含异步操作、提交mutations改变状态
- 6、Modules：将store分割成不同的模块

四、引入Vuex

1、安装vuex

```
npm install vuex --save
```

2、新建一个store文件夹（这个不是必须的），并在文件夹下新建index.js文件，文件中引入我们的vue和vuex

```

import Vue from 'vue'
import Vuex from 'vuex'

import app from './modules/app'
import user from './modules/user'
import permission from './modules/permission'
import getters from './getters'

Vue.use(Vuex)

export default new Vuex.Store({
  modules: {
    app,
    user,
    permission
  },
  state: {
  },
  mutations: {
  },
  actions: {
  },
  getters
})

```

3、在main.js 中引入新建的store

```

import store from './store/'

new Vue({
  el: '#app',
  router,
  store, //使用store
  template: '<App/>',
  components: { App }
})

```

五、页面使用

1. 读取store里的值:

this.\$store.state.字段名
如果有 module 的话, 假设叫 user , 那么取值又要变了, 加上 module 名

this.\$store.state.user.字段名

2. 发起操作请求:

this.\$store.dispatch('action中的方法名' , '参数') ;

参数你可以随便传json

3. getters的用法

this.\$store.getters.filterDoned
filterDoned 是在 todo 里写的一个 getters 方法, 就这么调用噢