

一. 状态管理 Vuex

1. Vuex使用

Vuex是一个专为 Vue.js 应用程序开发的状态管理模式。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

(1)state:单一状态树,每个应用将仅仅包含一个 store 实例。

(2)getters:可以从store 中的 state 中派生出一些状态，getters的返回值会根据它的依赖被缓存起来，且只有当它的依赖值发生了改变才会被重新计算。

(3)mutations: 更改 Vuex 的 store 中的状态的唯一方法是提交 mutation。

(4)actions:

*Action 提交的是 mutation，而不是直接变更状态。

*Action 可以包含任意异步操作。

*this.\$store.dispatch("type","payload")

(5)

```
1  const store = new Vuex.Store({  
2    state: {  
3      count: 0  
4    },  
5    mutations: {  
6      increment (state ,payload) {  
7  
8      }  
9    },  
10   actions: {  
11     increment (context, payload) {  
12       context.commit('increment')  
13     }  
14   }  
15 })
```

2. 注意:

(1)应用层级的状态应该集中到单个 store 对象中。

(2)提交 mutation 是更改状态的唯一方法，并且这个过程是同步的。

(3)异步逻辑都应该封装到 action 里面。

3. vue chrome devtools

二. vuex在项目中的使用

三. 热门组件库

1. 使用第三方插件

<https://github.com/vuejs/awesome-vue#components--libraries>

集合了来自社区贡献的数以千计的插件和库。

2. 使用第三方UI框架

饿了么UED团队推出的vue 前端框架：

(1)PC框架：(element UI , iview)

<http://element.eleme.io/#/>

<https://github.com/ElemeFE/element>

(2)移动端框架：(mint UI)

<https://mint-ui.github.io/docs/#/>

<https://github.com/ElemeFE/mint-ui>