

19.Vuex 安装和使用

学习要点：

1.Vuex 安装使用

本节课我们来开始了解 Vuex 安装和使用方式。

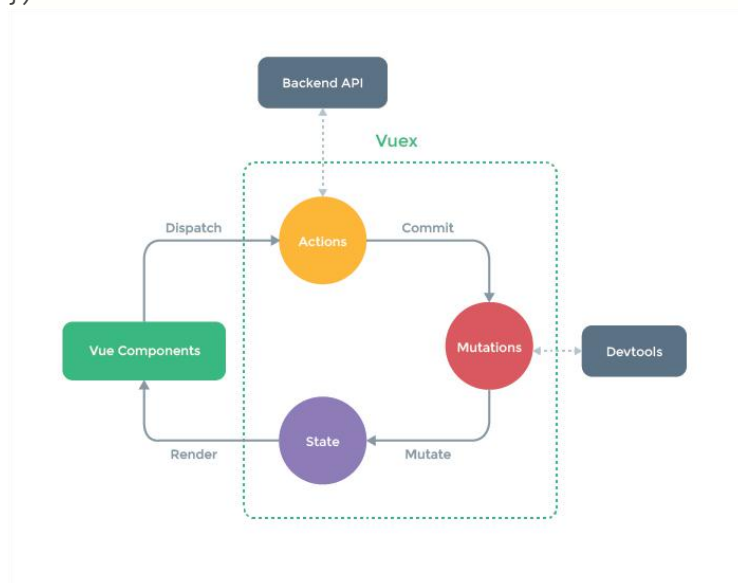
一. Vuex 安装使用

1. 上节课已经说了，安装脚手架时选择 Vuex 插件即可，其它不变；

```
Vue CLI v5.0.0-beta.1
? Please pick a preset: Manually select features
? Check the features needed for your project:
(*) Choose Vue version
(*) Babel
( ) TypeScript
( ) Progressive Web App (PWA) Support
(*) Router
(*) Vuex
( ) CSS Pre-processors
>( ) Linter / Formatter
( ) Unit Testing
( ) E2E Testing
```

2. Vuex 有两个特性：1.它是响应式的；2.不能直接改变 store 状态；
3. 安装好后，会在 src 目录下创建 store/index.js，先分析一拨；

```
export default new Vuex.Store({
  state: {}, //状态值
  mutations: {}, //修改状态
  actions: {}, //接口异步请求，服务端请求数据
  modules: {}
})
```



PS: Actions 获取数据->Mutations 修改处理->State 状态更新->Vue 组件渲染;

4. 再使用 Vuex 实现一个全局共享的计数器功能;

```
//状态值
state: {
  count : 0
},
//修改状态
mutations: {
  increment(state) {
    state.count++
  }
},

//About.vue
<h1>全局 count : {{$store.state.count}}</h1>
<button @click="increment">全局计数</button>
```

PS: \$store 是插件挂载在 Vue 实例上的，在实例内访问用 this.\$store;

```
export default {
  name : 'about',
  methods : {
    increment() {
      //修改状态执行 mutation 里的 increment 方法
      this.$store.commit('increment')
    }
  }
}
```

PS: 上面的单向循环图中可以看到 commit 字样，表示执行 mutations 用 commit 方法;

PS: 为何要使用 mutation 方法，是因为我们要明确追踪状态的变化，如果直接修改，则无;

```
this.$store.state.count++    //修改这里解释了特性第 2 点
```