

第二十三课 vuex 基本介绍与用法

学习目录

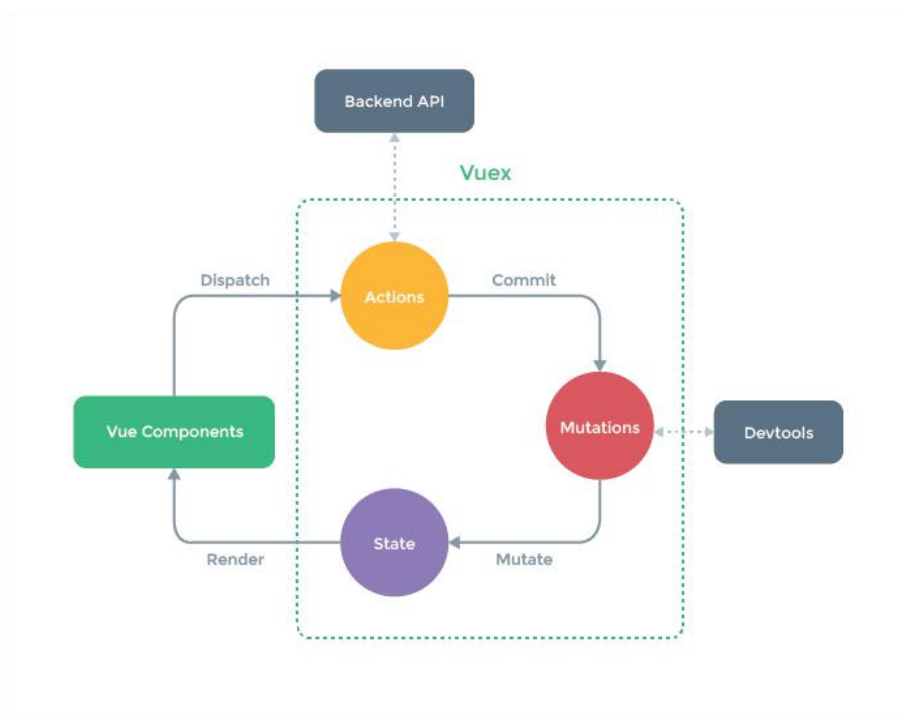
- vuex 基本介绍
- vuex 用法

一. vuex 基本介绍

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。

这是官方的定义,从这里我们可以分析出 vuex 的主要作用第一是集中式状态数据存储,第二是用可预测的方式修改状态数据。

简单来理解的话,等于是把我们定义在父组件中需要共享的数据存放在一个全局任何组件都能访问的地方,每个组件用符合相应规则的方式应用或者修改这些需要共享的数据。



二. vuex 用法

先要安装 vuex，在安装脚手架的时候我们其实安装好了。安装好以后，src 目录下会构建 store 文件夹，在这个文件夹下面构建 index.js，我们可以在这个文件中编写我们的 vuex 代码。

引入 vue 和 vuex，同时让 vue 使用 vuex，use 是 vue 使用插件的方法。

```
import Vue from 'vue'
```

```
import Vuex from 'vuex'
```

```
Vue.use(Vuex)
```

实例化 vuex 的实例对象，注意是实例化 Vuex 下面的 Store 方法，实例方法接收一个对象作为参数，我们在 state 中来定义我们的状态数据，在 mutations 中来定义我们修改 state 的方法。

```
const store = new Vuex.Store({  
  
  state: {  
  
    num: 0  
  
  },  
  
  mutations: {  
  
    add(state) {  
  
      state.num++  
  
    }  
  
  }  
  
})
```

Vuex 提供了从根组件向所有子组件，以 store 配置选项的方式“注入”定义好的 store 的机制，也就是在 main.js 中配置 store 选项对象吗，配置好以后我们可以在任意组件中访问这个定义好的 num 数据。

```
<div>{{ $store.state.num }}</div>

created(){

  console.log(this.$store.state.num);

}
```

假如我们要修改这个 num，不能直接修改赋值，必须要通过 mutations 用可预测的方式来修改 num，任意组件都可以调用 mutations 来完成对 state 中数据的修改，其他使用了 state 中数据的组件会响应式的得到修改之后的 state 数据。

```
<button @click="addFn">按钮 1</button>

methods: {

  addFn() {

    this.$store.commit('add');

  }

}
```

为什么我们要提交 mutation 修改 state，而非直接改变 store.state.num，是因为我们想要更明确地追踪到状态的变化。因此变化的管理是可预测的，这样我们的数据会按照我们想要的变化在任意组件中显示。

谢谢观看！如果觉得课程还不错的话，记得给个好评！

我是星星课堂老师：周小周