

## 知识点

### 状态管理store

使用store:

注册vuex和各个js文件。

状态管理核心状态管理有5个核心，分别是state、getter、mutation、action以及module。

**state:** 单一状态树，定义我们所需要管理的数组、对象、字符串等等

**getter:** getter有点类似vue.js的计算属性，当我们需要从store的state中派生出一些状态，那么我们就需要使用getter，getter会接收state作为第一个参数，而且getter的返回值会根据它的依赖被缓存起来，只有getter中的依赖值（state中的某个需要派生状态的值）发生改变的时候才会被重新计算。

**mutation:** 更改store中state状态的唯一方法就是提交mutation，就很类似事件。每个mutation都有一个字符串类型的事件类型和一个回调函数，我们需要改变state的值就要在回调函数中改变。我们要执行这个回调函数，那么我们需要执行一个相应的调用方法：

store.commit。

**action:** action可以提交mutation，在action中可以执行store.commit，而且action中可以有任何的异步操作。在页面中如果我们要调用这个action，则需要执行

store.dispatch、module module其实只是解决了当state中很复杂臃肿的时候，module可以将store分割成模块，每个模块中拥有自己的state、mutation、action和getter。

### 请求拦截

// 请求拦截

```
service.interceptors.request.use(  
  config => {  
    // if(store.getters.token) {  
    //   config.headers['token'] = getToken()  
    // }  
    config.headers['withCredentials'] = true  
    return config  
  },  
  error => {  
    console.log(error)  
    Promise.reject(error)  
  }  
)
```

```
service.interceptors.response.use(  
  response => response.data  
)
```