1. 测试要求
   1. 新增代码UT覆盖率必须超过30%， 否则PO不予验收；
   2. 全部代码的覆盖率要逐步提升到50%以上；
   3. 覆盖率统计以sonar9001为准，地址：
   4. <http://sonar.ispeco.com:9001/dashboard?id=com.supermap%3Awebapps-map-dashboard>
2. 测试环境介绍
   1. 项目使用jest 进行测试框架，同时使用
      1. Vue Test Utils（vue官方推出的测试辅助工具库）https://vue-test-utils.vuejs.org/zh/
      2. jest-mock-axios用于测试异步等功能https://www.npmjs.com/package/jest-mock-axios
      3. sinon库，用于替代函数，<https://www.npmjs.com/package/sinon>
   2. 测试案例目录为： ./test/unit/
3. 测试案例编写规范说明
   1. 测试文件名统一命名为： xxx.spec.js （xxx === 被测试文件名）；
   2. 测试文件路径与src路径下保持一致，方便查找；
   3. 测试中，因需要尽可能模拟开发环境，所以写了一个通用的工具文件 TestUtil.js，其中只做了vue初始化依赖，引入资源文件， 以及全局变量的定义等工作，写测试文件时，可以直接引入该文件，即可完成环境的初始化工作；
   4. 测试文件文件应该至少包含一个 describe 函数，用以包裹具体测试案例；
   5. 每一个测试案例写为一个it 函数，结构如下：
4. it('getSceneList-翻页搜索', ()=>{  
    wrapper.vm.getSceneList(2);  
    expect(wrapper.vm.**param**.**currentPage**).toBe(2);  
   })

第一个函数是该测试案例的说明，结构为： 被测试函数名-中文释义，方便快捷定位被测试函数；

f) 每一个测试案例中至少有一个断言语句，且保证断言的有效性；

1. 测试技巧
   1. 函数mock，如果一个函数A的结果并没有改变data，而是调用了一个其他函数，那么可以通过函数mock来验证该函数A是否正确执行；

具体代码参考 chart.spec.js 中对于selectChart 的测试；

b） 测试方法中的setTimeut(), 需要加入done参数,作为结束该测试的标志位，

例如it(“xxxxx”,(done)=>{

setTime(()=>{expect(“XXX”).to(“XXXX”); done();},200);

});