Vue单元测试

**测试环境搭建及配置**

1. 使用脚手架（vue-cli）初始化vue项目

npm install -g vue-cli

vue init webpack vue-unit-test-project

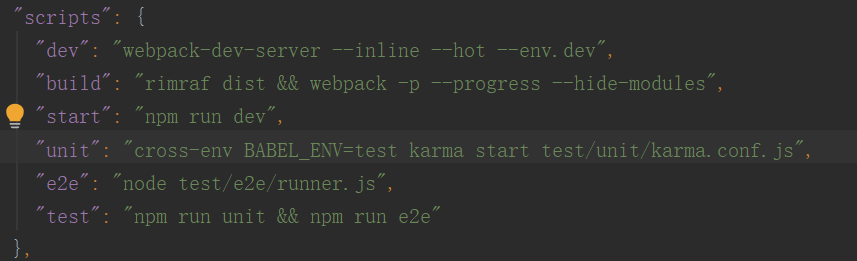
1. 由于所用Element项目模板没有congfig、test以及build文件夹，把新建项目中相应的文件夹复制到我们的项目中。
2. 安装karma+mocha模块

npm i -D karma karma-webpack phantomjs-prebuilt karma-phantomjs-launcher karma-phantomjs-shim karma-chrome-launcher karma-sourcemap-loader mocha karma-mocha sinon chai sinon-chai karma-sinon-chai karma-spec-reporter karma-coverage

1. 安装Vue-test-utils

Npm install –save-dev vue-test-utils

1. 修改package.json



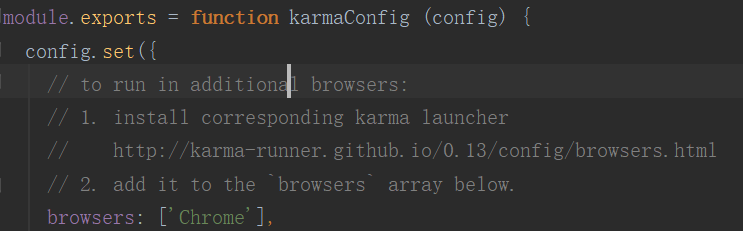
1. 为了解决一些容易出现的错误，进行一下配置：
2. 修改test目录下的index.js文件



1. 安装Karma-chrome-lauch

npm install karma-chrome-laucher –save-dev

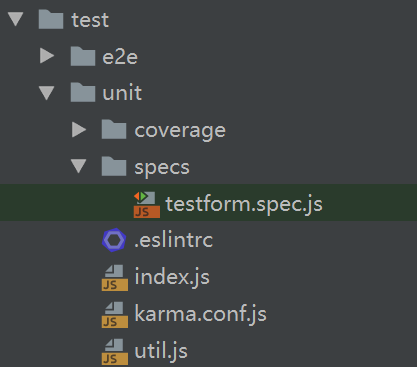
然后修改test/unit/karma.conf.js文件



可减少由默认浏览器PhantomJS引起的错误

**测试部分**

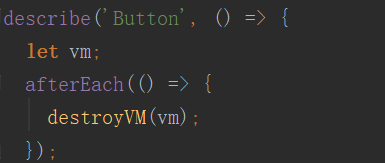
1. 测试脚本都要放在 test/unit/specs/ 目录下，脚本命名方式为 [组件名].spec.js。 例如这里是testform.vue组件的测试脚本。



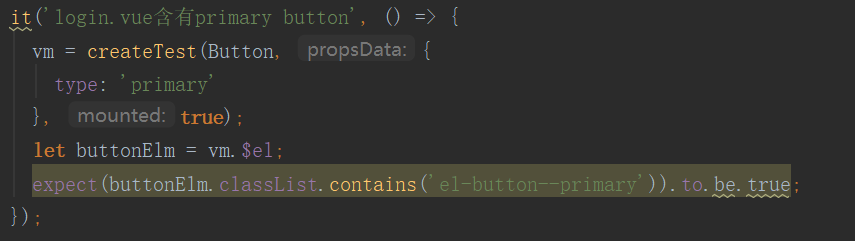
2. 使用Element的[单元测试工具脚本Util.js](https://github.com/ElemeFE/element/blob/dev/test/unit/util.js)，它封装了Vue单元测试中常用的方法。Utils.js文件放在test/unit目录下。

3. 对登陆页面的button进行单元测试。Element提供了很多测试脚本可以学习。

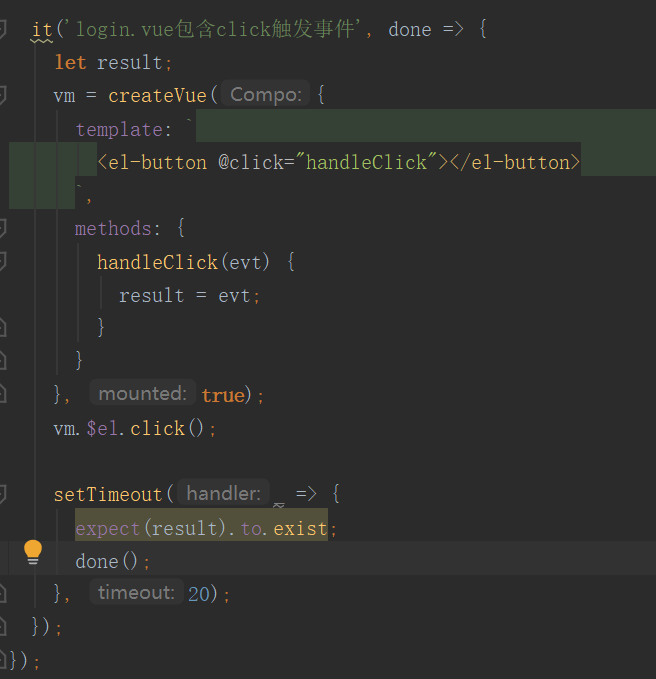
对于Vue的单元测试，需要将组件实例化为一个Vue实例，



查询通过contains方法获取，查找的结果为true或false



每一个describe可以包含多个it



测试结果：

