### 前期准备:

○ 开发工具: IntelliJ○ 工程: Maven工程

## Maven 环境搭建:

可参考: <a href="http://note.youdao.com/noteshare?">http://note.youdao.com/noteshare?</a>

id=4abe2a58d17c39e243d5dd0ddad53114

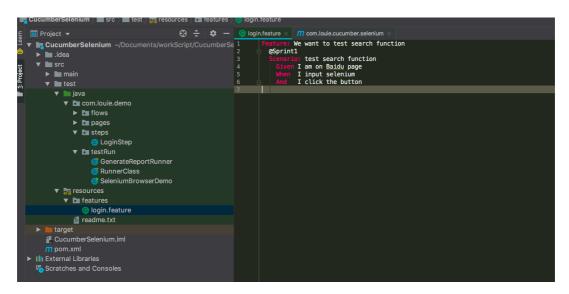
## 需求:

测试百度搜索功能是否正常。需要提供自动化测试结果。而不是手工测试结果。

#### 脚本编写:

1. pom.xml 加载相关依赖包

2. 根据需求编写测试点 ,即Feature文件 (不是可执行脚本)



3.根据Feature 文件 进行反射成Java可执行脚本. 即 step 文件

```
| CucumberSelenium | Marco | Martin | Marco | Martin | Ma
```

注\*:初学者可以将逻辑和页面元素写在映射文件中。后面不建议写在映射文件中,需要学会分层模式。映射文件只是负责映射Feature文件,Flow 文件负责编写逻辑,Page文件负责页面元素。

4.运行文件,即 RunnerClass

```
E_ CucumberSelenium ⟩ E src ⟩ E test ⟩ E java ⟩ E com ⟩ E louie ⟩ E demo ⟩ E testRun ⟩ © RunnerClass
                                                                 ⊕ 🛨 🌣 — © login.feature × © LoginStep.java × © RunnerClass.java × m com.louie.cucumber.selenium kScript/CucumberSe 1 package com.louie.demo.testRun; 2 3 cimport ... 6 7 c@RunWith(Cucumber.class)
     ▶ 🖿 .idea
▼ 🖿 src
                                                                                                                  th(Cucumber.class)
berOptions(
//漢行生成的报告json
plugin ={"pretty",
"html:target/cucumber-report/",
"json:target/cucumber-report/cucumber.json"
         ▶ main
▼ test
                 ▼ 🖿 com.louie.demo
► 🗖 flows
                     ▼ 🖿 steps
                                                                                                                    features={"src/test/resources/features"},
                                                                                                                    glue ={"com.louie.demo.steps"},
                      ▼ 🖿 testRun
                          GenerateReportRunner
RunnerClass
SeleniumBrowserDemo
                                                                                                                   //运行指定的场景
tags = {"@Sprint1"}
)
class RunnerClass {
             ▼ resources
▼ la features
⊗ login.feature
                    readme.txt
         target

CucumberSelenium.iml
  m pom.xml
► III External Libraries
      Scratches and Consoles
```

分层模式: 重写上面代码:

1. 编写测试需求,即 Feature文件

2.编写Feature映射文件, 即step 文件

```
| Project | Proj
```

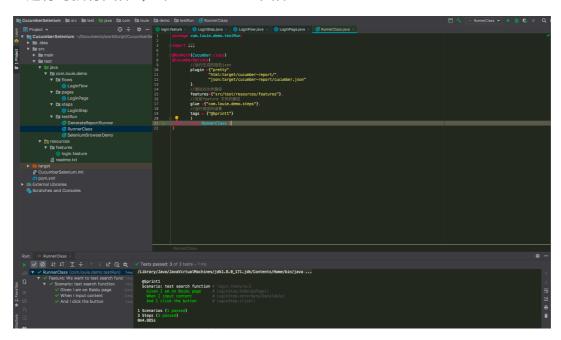
3. 编写自动化测试逻辑操作, 即 flow 文件

```
| Project | Service | Serv
```

4. 定位页面元素,与selenium 底层操作,即page文件

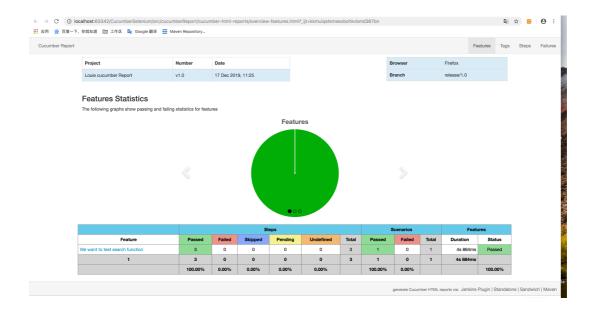
```
| Particle | Particle
```

# 5.运行可执行文件 ,即Runnerclass 文件



#### 6.运行可视化报告

```
| Second content | Seco
```



7. 代码下载: (项目名: CucumberSelenium) https://github.com/gtlouie/Auto-Test-Share