Robot Framework + Selenium

自动化测试入门

--虫师

个人博客：
http://fnng.cnblogs.com

更多测试课程与公开课：
http://itest.info/

目录

介绍.....2
Robot Framework 是什么？.....2
Selenium 是什么？.....2
一、Robot Framework Selenium 环境搭建.....4
二、创建第一个自动化脚本.....5
1、创建测试项目.....6
2、创建测试套件.....6
3、创建测试用例.....7
4、导入 selenium2library 库.....7
5、编写用例.....8
6、运行测试用例.....9
三、Selenium 元素定位.....11
1、id 和 name 定位.....11
2、xpath 定位.....12
3、CSS 定位.....14
四、Robot Framework Selenium API.....16
一、浏览器驱动.....17
二、关闭浏览器.....18
三、浏览器最大化.....18
四、设置浏览器宽、高.....18
五、文本输入.....19
六、点击元素.....19
七、点击按钮.....19
八、注释.....19
九、固定时间休眠.....20
十、等待元素出现在当前页面.....20
十一、获取 title.....20
十二、获取文本信息.....21
十三、获取元素属性值.....21
十四、cookie 处理.....21
十五、声明变量.....22
十六、日志（输出）.....22
十七、获得浏览器窗口宽、高.....23
十八、验证.....23
十九、表单嵌套.....24
二十、下拉框选择.....24
二十一、If 分支语句.....24
二十二、for 循环语句.....25
五、Robot Framework 分层设计.....26
1、创建用户关键字.....28
2、添加调用关键字.....31

介绍

Robot Framework 是什么？

Robot Framework 的架构是一个通用的验收测试和验收测试驱动开发的自动化测试框架（ATDD）。它具有易于使用的表格来组织测试过程和测试数据。

New Test Case		
open browser	http://www.baidu.com	
input text	id=kw	robot framework
click button	id=su	
close browser		

它使用关键字驱动的测试方法。

对于上面的例子来说，open browser 、input text、click button 和 close browser，都是“关键字”，这些关键字由 robotframework-selenium2library 类库所提供。当然，我们也可以自定义关键字。

其检测能力可以通过测试库实现可以使用 Python 或 Java 的扩展，用户可以使用相同的语法，用于创建测试用例创建新的更高层次的现有的关键词。

Robot Framework 的操作系统和应用独立框架。核心框架是使用 Python 和运行在 Jython（JVM）和 IronPython（.NET）。

Selenium 是什么？

Selenium 是 web 自动化测试工具集，包括 IDE、Grid、RC（selenium 1.0）、WebDriver（selenium 2.0）等。

Selenium IDE 是 firefox 浏览器的一个插件。提供简单的脚本录制、编辑与回放功能。

Selenium Grid 是用来对测试脚步做分布式处理。现在已经集成到 selenium server 中了。

RC 和 WebDriver 更多应该把它看成一套规范，在这套规范里定义客户端脚步与浏览器交互的协议。以及元素定位与操作的接口。

WebDriver API 是什么？

对于刚接触 selenium 自动化测试的同学来说不太容易理解 API 是什么，它到底和编程语言之是什么关系。

Webdriver API(接口规范)说，我们要提供一下 id 的定位方法。

Ruby 的 webdriver 模块是这么实现的：

```
require "selenium-webdriver"    #导入 ruby 版的 selenium(webdriver)

find_element(:id, "xx")      #id 定位方法
```

C#的 webdriver 模块是这么实现的：

```
using OpenQA.Selenium;
using OpenQA.Selenium.Firefox;   //导入 C#版的 selenium(webdriver)

FindElement(By.Id("xx"))    //id 定位方法
```

python 的 webdriver 模块是这么实现的：

```
from   selenium import webdriver      #导入 python 版的 selenium(webdriver)

find_element_by_id("xx")    #id 定位方法
```

Java 的 webdriver 模块是这么实现的：

```
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;    //导入 java 版的 selenium(webdriver)

findElement(By.id("xx"))      //id 定位方法
```

Robot Framework +selenium

因为 Robot Framework 对于底层过于封装，所以，更看不到语言层面的方法定义。所以，提供给我们的方法如下：

1、导入 Robot Framework 版本的 selenium（webdriver）

Import	Name / Path	Arguments	Comment
Library	Selenium2Library		

2、使用 id 方法

Click element	id=xx	

需要说明的是 webdriver API 只提供了页面操作的相关规范，比如元素定位方法，浏览器操作，获取 web 页元素属性等。

一、Robot Framework Selenium 环境搭建

最近工具中用 Robot Framework 框架来做自动化，所以，花时间学习了一下。

=======所需环境==========

Python:

https://www.python.org/

RF 框架是基于 python 的，所以一定要有 python 环境。

Robot framework：

https://pypi.python.org/pypi/robotframework/2.8.5

这个不是解释了，RF 框架。虽然在做基于 UI 的自动化时，它展现出来的很像 QTP，我之前也以为它和 QTP 差不多，仔细了解你会发展它能做的事情还是很多的。就像初学 selenium 者，会误以为 selenium 就是 selenium IDE。

wxPython :

http://www.wxpython.org/download.php

Wxpython 是 python 非常有名的一个 GUI 库，因为 RIDE 是基于这个库开发的，所以这个必须安装。

Robot framework-ride

https://pypi.python.org/pypi/robotframework-ride

RIDE 就是一个图形界面的用于创建、组织、运行测试的软件。

Robot framework-selenium2library：

https://pypi.python.org/pypi/robotframework-selenium2library/1.5.0

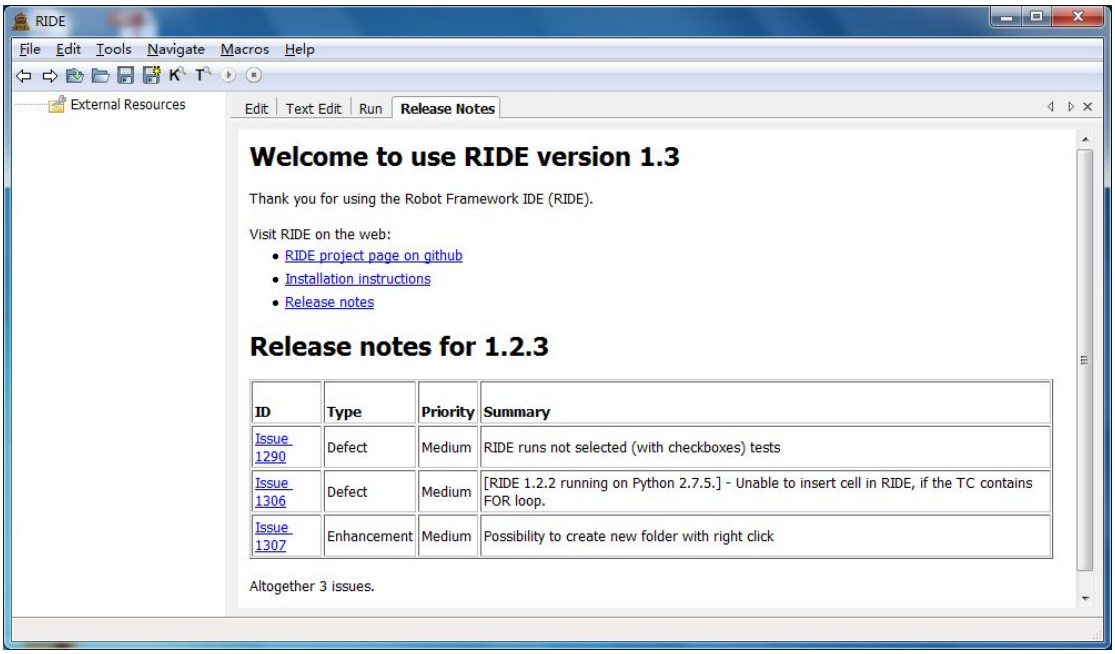
RF-seleniumlibrary 可以看做 RF 版的 selenium 库，selenium （webdriver）可以认为是一套基于 web 的规范（API），所以，RF 、appium 等测试工具都可以基于这套 API 进行页面的定位与操作。

可以通过 python 的 pip 工具包进行安装：

>pip install robotframework-selenium2library

如果初次接触上面的东西的话，觉得装的东西有点多。如果之前有了解过 python 或 selenium 的话就不会有这样的感觉。

在你安装好 RF-RIDE 之后，桌面就会生成一个 RIDE 图标。双击启动，界面如下：

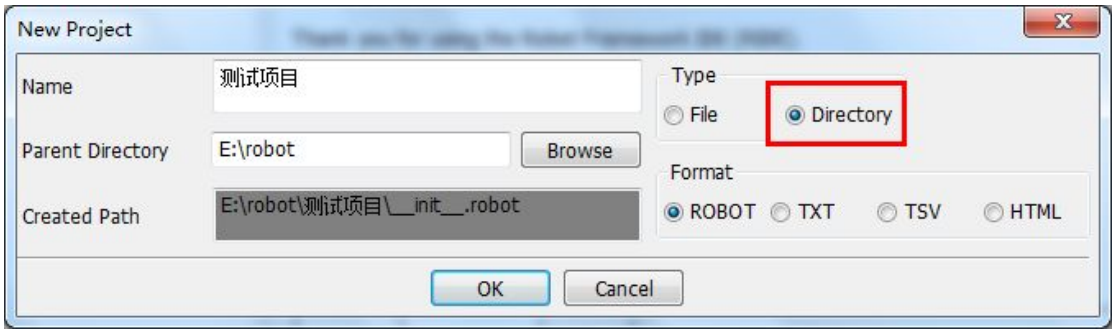


二、创建第一个自动化脚本

下面我们就一步一步的创建第一条用例，至于细节不多解释，只是对 RF 框架写用例有个感性的认识。

1、创建测试项目

选择菜单栏 file----->new Project

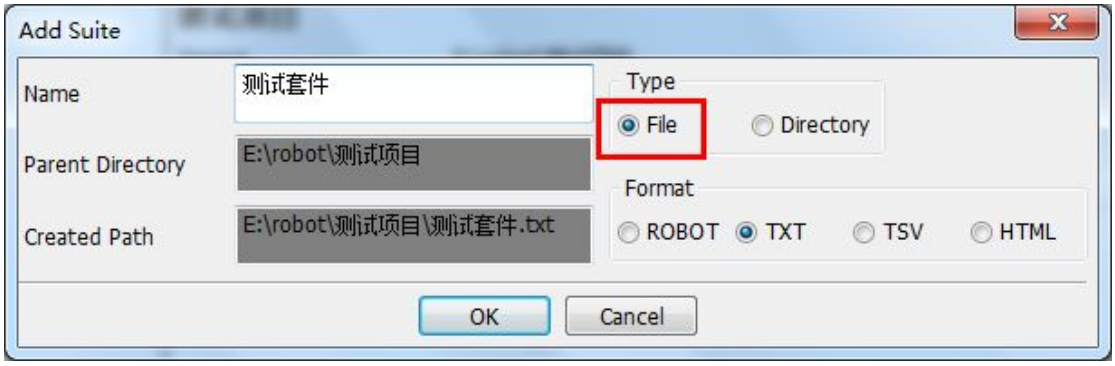


Name 输入项目名称。

Type 选择 Directory。

2、创建测试套件

右键点击“测试项目”选择 new Suite 选项

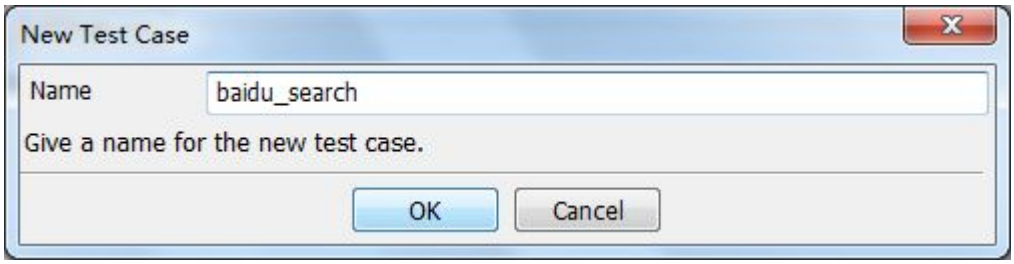


Name 输入项目名称。

Type 选择 File。

3、创建测试用例

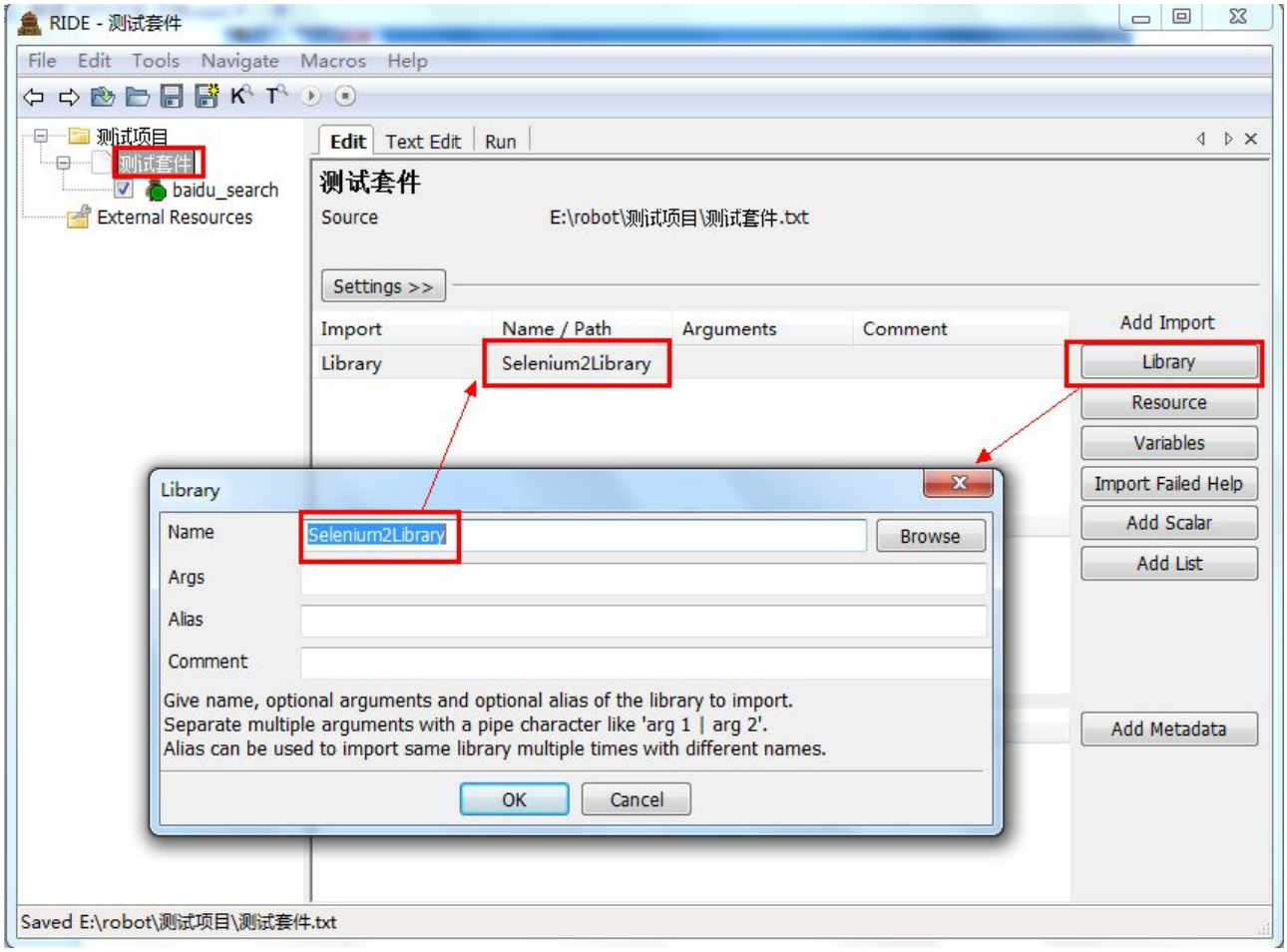
右键点击“测试项目”选择 new Test Case



用例只需要输入用例 name ， 点击 OK 即可。

4、导入 selenium2library 库

因为 RF 框架编写基于 web 的测试用例，所以，我们需要 selenium 的库支持。所以，我们在使用的过程中需要加载 selenium2library 库。

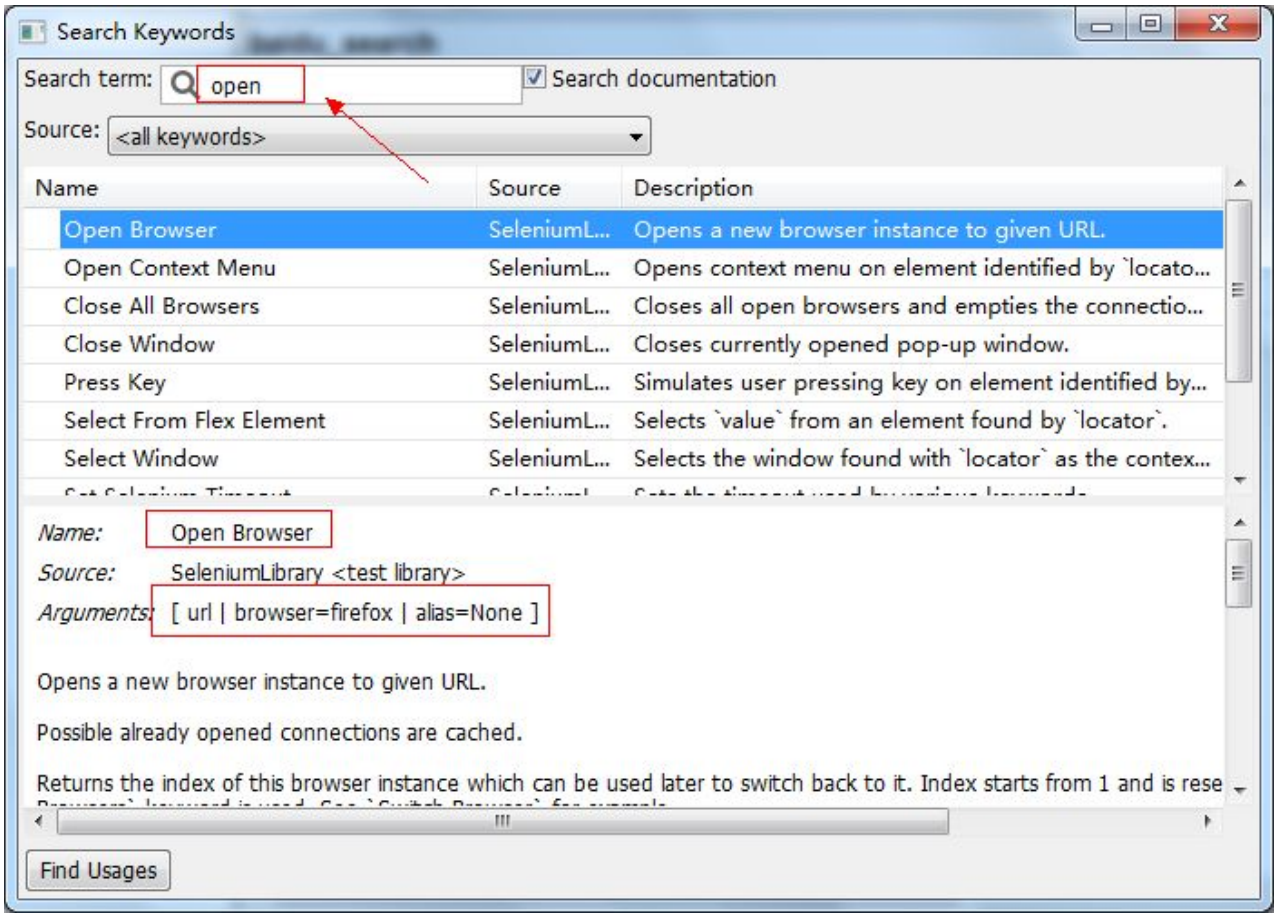


在“测试套件”的 Edit 标签页，点击“Library”按钮，弹出输入框，Name 输入：Selenium2Library，点击 OK 完。

如果导入的库显示为红色，表示导入的库不存在。如果是黑色则表示导入成功。

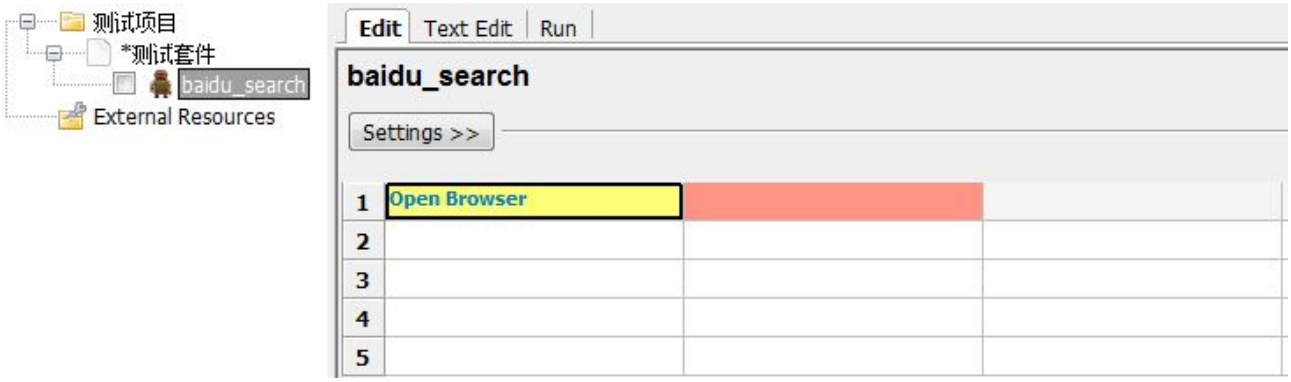
5、编写用例

下面就可以开始写我们的用例了，可是怎么写呢？我们可以通过按 F5 快捷键来查询脚本的关键字。如果你接触过 QTP 或 selenium IDE 等自动化工具的话，应该会有一些思路。



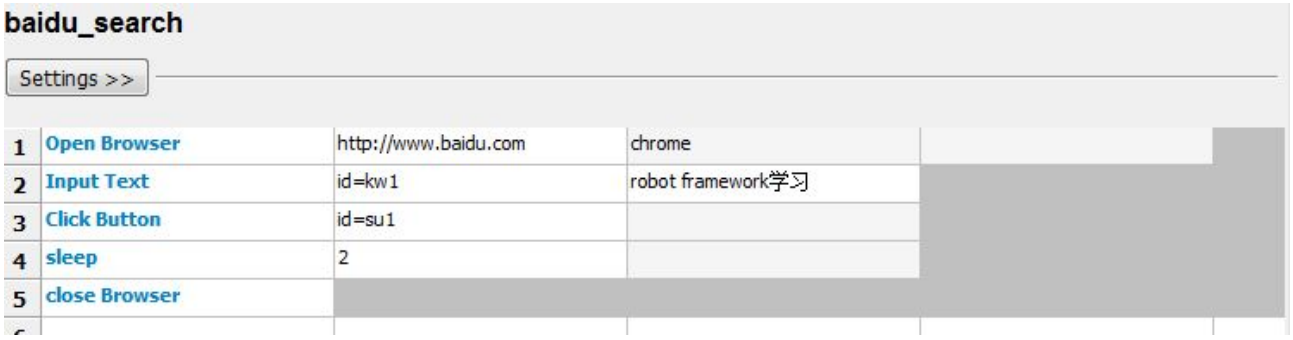
如上图，自动化脚本从打开浏览器开发，如上图，我想打开一个浏览器，想的是“open”为关键字进行搜索，结果找到了一个“Open Browser”的关键字，点击这个关键字，想显示它的用法和说明。

根据说明，我们来尝试创建这个打开浏览器的操作吧：



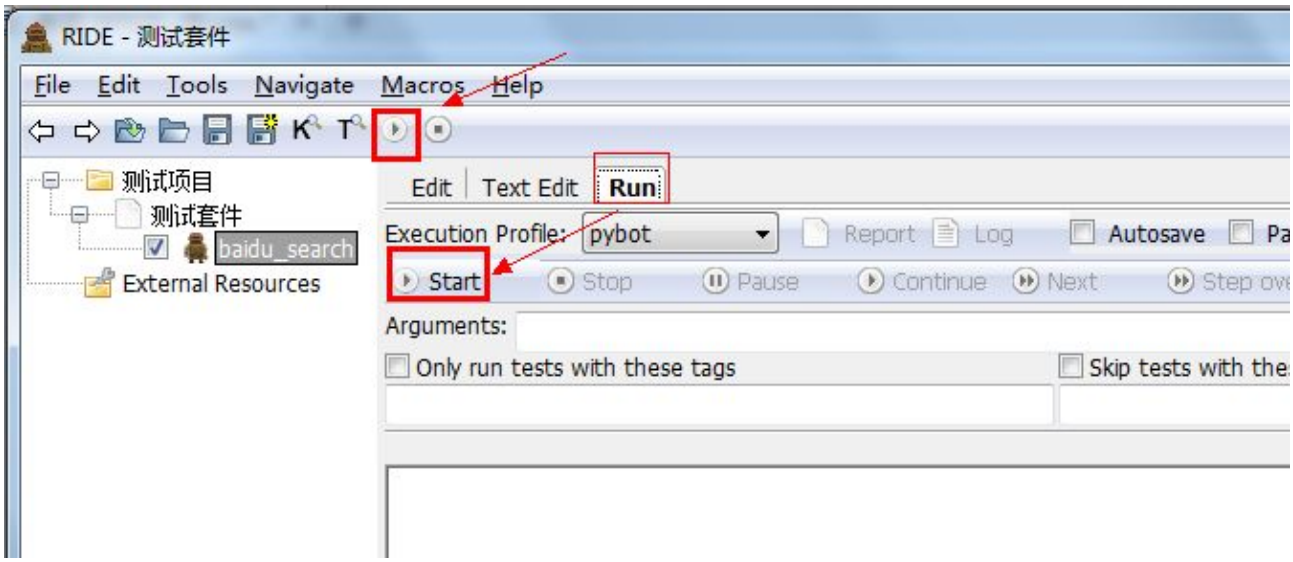
“Open Browser”变蓝了，说明它是一个合法的关键字，后面有一个方框是红色的，表示这个参数不能缺省的。通过说明信息中，我发现它需要一个 url 地址是必填的，当然还需要指定 browser （默认不填为 friefox）

更多关键的使用，请参考相关 API 文档。这里不过多介绍。按照上面的方法。创建百度搜索用例如下：

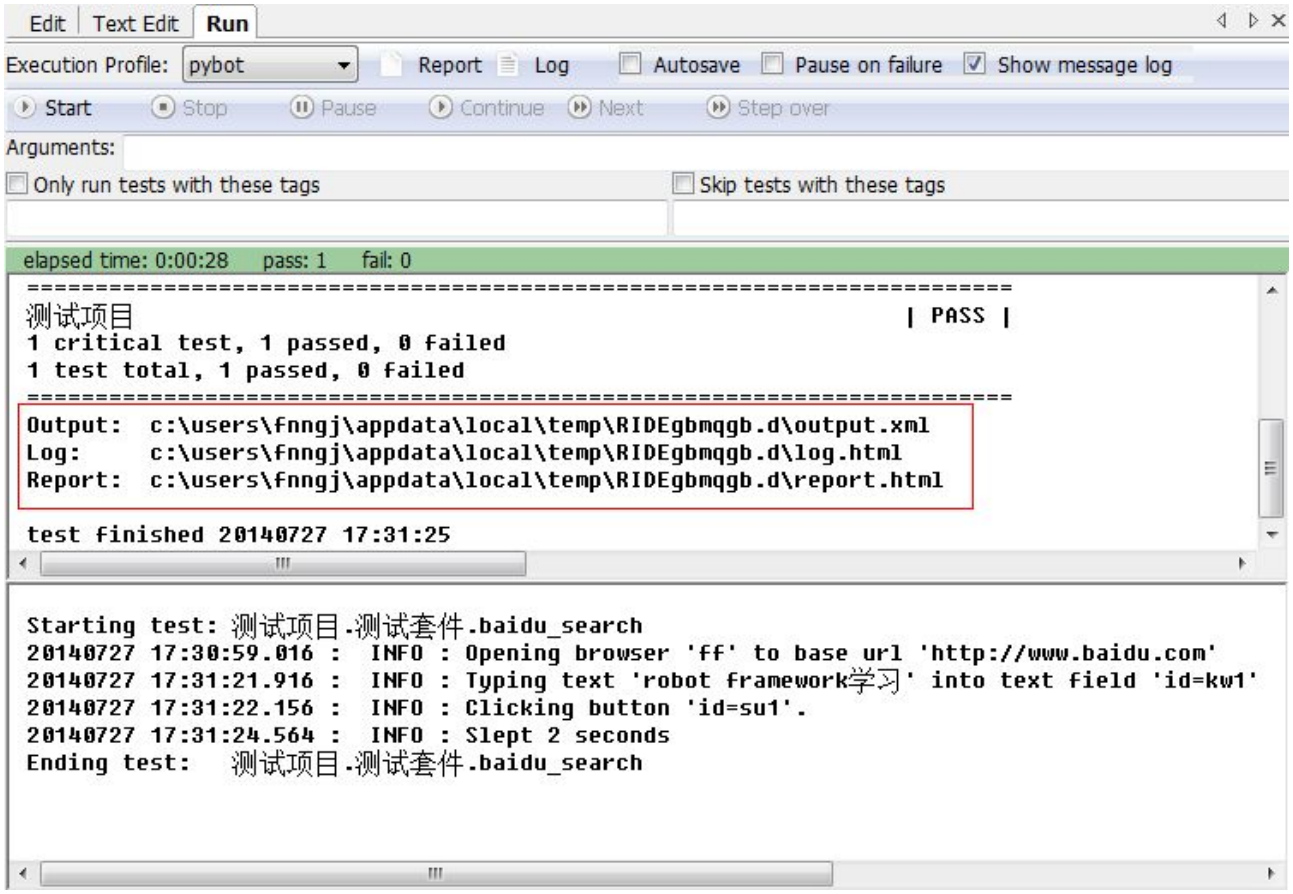


6、运行测试用例

勾选当前需要运行的测试用例，点击工具栏运行按钮，如果只运行单个用例的话，也可以切换到用例的 Run 标签页，点击“start”按钮。



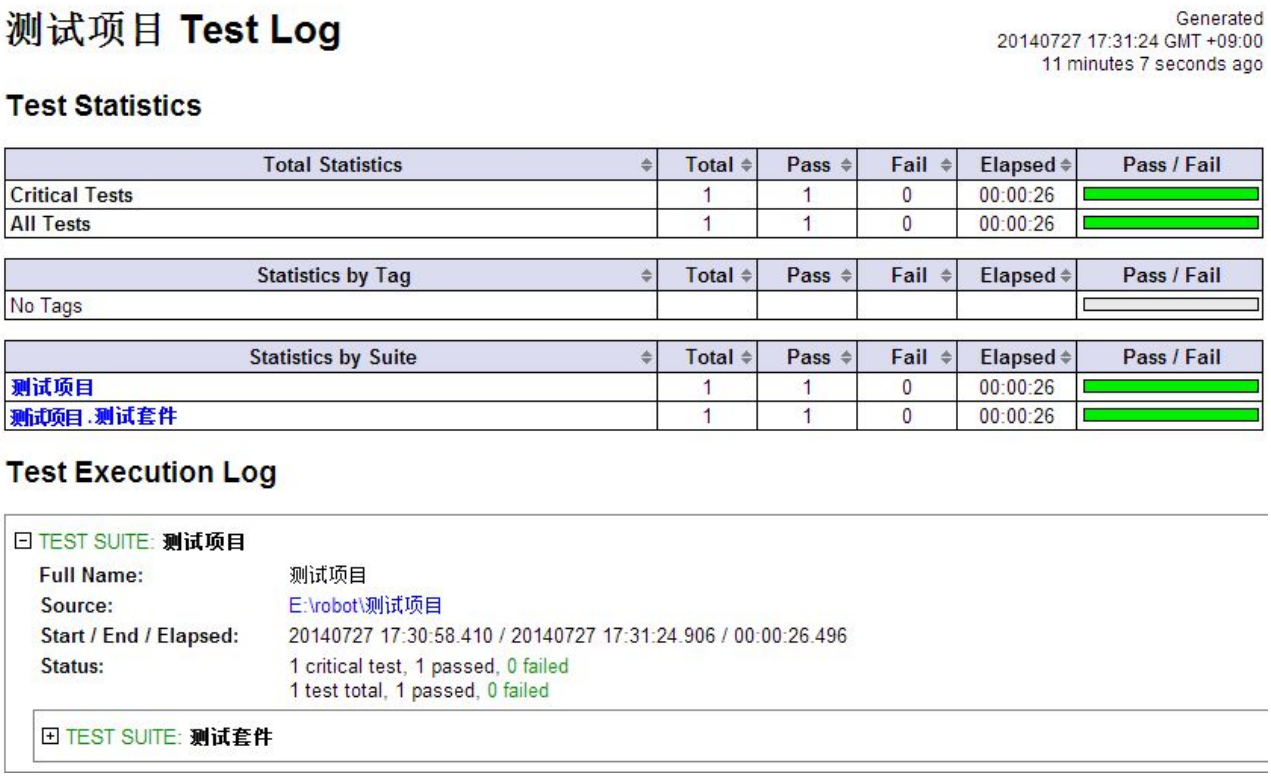
运行结果：



运行信息显示会生成三个文件：Output.xml、Log.html、Report.html

我们重点查看 Log.html 和 Report.html ，Log.html 更关注脚本的执行过程的记录，Report.html 更关注脚本的执行结果的展示。

赶快打开你的测试报告看看效果吧！



测试项目 Test Log

Generated
20140727 17:31:24 GMT +09:00
11 minutes 7 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:26	
All Tests	1	1	0	00:00:26	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
测试项目	1	1	0	00:00:26	
测试项目.测试套件	1	1	0	00:00:26	

Test Execution Log

TEST SUITE: 测试项目

Full Name: 测试项目
Source: E:\robot\测试项目
Start / End / Elapsed: 20140727 17:30:58.410 / 20140727 17:31:24.906 / 00:00:26.496
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

TEST SUITE: 测试套件

三、Selenium 元素定位

1、id 和 name 定位

假如把一个元素看作一个人的话，id 和 name 可以看作一个人的身份证号和姓名。当然，这些属性值是否唯一要看前端工程师如何设计了。

百度搜索框和搜索按钮

```
.....
<input id="kw1" class="s_ipt" type="text" maxlength="100" name="wd" autocomplete="off">
.....
<input id="su1" class="bg s_btn" type="submit" onmouseout="this.className='bg s_btn'"
onmousedown="this.className='bg s_btn s_btn_h'" value="百度一下">
.....
```

根据上面的例子，百度输入框可以取 id 或 name 进行定位。（前提是 id 和 name 的值在当页面上唯一）

id = kw1

name = wd

在 Robot framework 中就是这样写的：

Input Text	id=kw1	robot framework 学习
input text	name=wd	robot framework 学习

Input text 用于输入框的关键字，“robot framework 学习”是要给输入框输入的内容。

百度按钮只 id 数据可以利用：

id=su1

Click Button	id=su1	

Click Button 是按钮点击的关键字。

2、xpath 定位

假如，一个人没身份证号没名字怎么找呢？想想你是怎么找朋友吃饭的，他手机不通，电话不回呢？直接上他家去呗，那你一定有他家住址，xx 市 xx 区 xx 路 xx 号。Xpath 就可以通过这种层级关系找到元素。

来看看百度输入框在整个页面上的位置吧：

```
<html>
    <head>
    <body link="#0000cc">
    <div id="wrapper" style="display: block;">
        <div id="debug"
style="display:block;position:absolute;top:30px;right:30px;border:1px solid;padding:5px
10px;"></div>
        <div id="u">
        <div id="head">
        <div id="content" style="display: block;">
            <div id="u1" style="display: block;">
            <div id="m">
                <p id="lg">
                <p id="nv">
                <div id="fm">
```

```
                <form id="form1" class="fm" action="/s" name="f1">
                    <span class="bg s_ipt_wr">
                        <input id="kw1" class="s_ipt" type="text" maxlength="100" name="wd"
autocomplete="off">
```

1、Xpath 的绝对路径：

Xpath = /html/body/div[1]/div[4]/div[2]/div/form/span[1]/input

我们可以从最外层开始找，html 下面的 body 下面的 div 下面的第 4 个 div 下面的....input 标签。通过一级一级的锁定就找到了想要的元素。

2、Xpath 的相对路径：

绝对路径的用法往往是在我们迫不得已的时候才用的。大多时候用相对路径更简便。

2.1、元素本身：

Xpath 同样可以利用元素自身的属性：

Xpath = //*[@id='kw1']

//表示某个层级下，*表示某个标签名。@id=kw1 表示这个元素有个 id 等于 kw1 。

当然，一般也可以制定标签名：

Xpath = //input[@id='kw1']

元素本身，可以利用的属性就不只局限为于 id 和 name ， 如：

Xpath = //input[@type='text']

Xpath = //input[@autocomplete='off']

但要保证这些元素可以唯一的识别一个元素。

2.2、找上级：

当我们要找的一个人是个刚出生的婴儿，还没起名子也没有入户口（身份证号），但是你会永远跟在你父亲的身边，你的父亲是有唯一的名字和身份证号的，这样我们可以先找到你父亲，自然就找到你的。

元素的上级属性为：

```
<form id="form1" class="fm" action="/s" name="f1">
    <span class="bg s_ipt_wr">
```

```
<input id="kw1" class="s_ipt" type="text" maxlength="100" name="wd" autocomplete="off">
```

找爸爸：

xpath = //span[@class='bg s_ipt_w']/input

如果爸爸没有唯一的属性，可以找爷爷：

xpath = //form[@id='form1']/span/input

这样一级一级找上去，直到 html ，那么就是一个绝对路径了。

2.3、布尔值写法：

如果一个人的姓名不是唯一的，身份证号也不是唯一的，但是同时叫张三 并且 身份证号为 123 的人却可以唯一的确定一个人。那么可以这样写：

Xpath = //input[@id='kw1' and @name='wd']

可以 and ， 当然也可以 or :

Xpath = //input[@id='kw1' or @name='wd']

但 or 的实际意义不太。我们一般不需要说，找的人名字或者叫张三，或者身份证号是 123 也可以。

Robot framework 中的写法：

Input Text	xpath = //*[@id='kw1']	robot framework 学习
input text	xpath = //span[@class='bg s_ipt_w']/input	robot framework 学习
input text	xpath = //input[@id='kw1' and @name='wd']	robot framework 学习

3、CSS 定位

Css 的定位更灵活，因为他它用到的更多的匹配符和规格。

http://www.w3school.com.cn/cssref/css_selectors.asp

选择器	例子	例子描述
.class	.intro	选择 class="intro" 的所有元素。
#id	#firstname	选择 id="firstname" 的所有元素。
*	*	选择所有元素。
element	p	选择所有 <p> 元素。
element,element	div,p	选择所有 元素和所有 <p> 元素。
element element	div p	选择 元素内部的所有 <p> 元素。
element>element	div>p	选择父元素为 元素的所有 <p> 元素。
element+element	div+p	选择紧接在 元素之后的所有 <p> 元素。
[attribute]	[target]	选择带有 target 属性所有元素。
[attribute=value]	[target=_blank]	选择 target="_blank" 的所有元素。
[attribute~=value]	[title~=flower]	选择 title 属性包含单词 "flower" 的所有元素。
[attribute|=value]	[lang|=en]	选择 lang 属性值以 "en" 开头的所有元素。

同样以百度输入框的代码，我们来看看 CSS 如何定位。

```
<form id="form1" class="fm" action="/s" name="f1">
    <span class="bg s_ipt_wr">
        <input id="kw1" class="s_ipt" type="text" maxlength="100" name="wd"
autocomplete="off">
```

id 定位：css=#kw1

class 定位：css=.s_ipt

其它属性：css=[name=wd]

css=[type=text]

css=[autocomplete=off]

父子定位：css=span > input

css=form > span > input

根据标签名定位：css=input

Robot framework 中的写法：

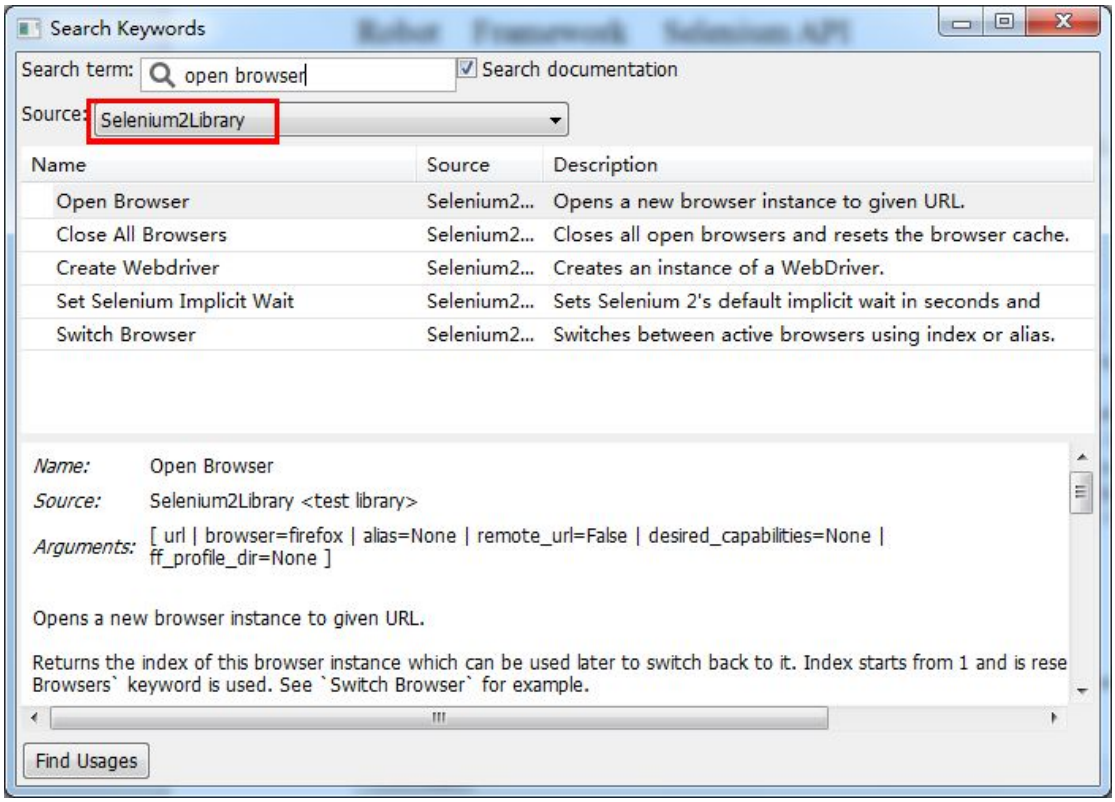
Input Text	css=#kw1	robot framework 学习
input text	css=.s_ipt	robot framework 学习
input text	css=[name=wd]	robot framework 学习

同样一个元素，根基 CSS 的不同规则，可能有几十上百种写法。CSS 更灵活强大，但是相比 path 的学习成本为更高。但是 css 和 xpath 两种定位方式是一定要学会一种，不然你的自动化工作更无法开展。

四、Robot Framework Selenium API

说明：

此文档只是将最常用的 UI 操作列出。更多方法请查找 selenium 关键字库。



一、浏览器驱动

通过不同的浏览器执行脚本

Open Browser	Htpp://www.xxx.com	chrome

浏览器对应的关键字：

firefox	FireFox
ff	
internetexplorer	Internet Explorer
ie	
googlechrome	Google Chrome
gc	
chrome	
opera	Opera
phantomjs	PhantomJS
htmlunit	HTMLUnit
htmlunitwithjs	HTMLUnit with Javascipt support
android	Android

iphone	Iphone
safari	Safari

备注：

要想通过不同的浏览打开 URL 地址，一定要安装浏览器相对应的驱动。如 chrome 的驱动：chromedriver.exe 等。

浏览器默认为空时启动 FireFox。

二、关闭浏览器

关闭浏览器

Close Browser		

关闭当前的浏览器。

关闭所有浏览器

Close All Browsers		

关闭所有打开的浏览器和浏览器缓存重置。

三、浏览器最大化

Maximize Browser Window		

使当前打开的浏览器全屏。

四、设置浏览器宽、高

Get Window Size	800	600

以像素为单位，第一个参数 800 表示宽度，第二个参数 600 表示高度。

五、文本输入

Input Text	Xpath=//* [@]	输入信息

Xpath=//* [@] ：表示元素定位，定位文本输入框。

六、点击元素

Click Element	Xpath=//* [@]	

Xpath=//* [@] ：表示元素定位，定位点击的元素。

七、点击按钮

Click Button	Xpath=//* [@]	

Xpath=//* [@] ：表示元素定位，定位点击的按钮。

八、注释

注释 1：

Comment	注释说明	

注释 2：

# 注释说明		

除了使用 Comment 关键字进行注释外，Robot framework 框架是基于 python 语言开发的，所以提供了 python 语言的注释“#”方式。

九、固定时间休眠

Sleep	42	
Sleep	1.5	
Sleep	2 minutes 10 seconds	

Sleep 表示执行到当前行固定休眠多长时间，以“秒”为单位。

42 表示 42 秒；

1.5 表示 1.5 秒；

2 minutes 10 seconds 表示 2 分 10 秒。

十、等待元素出现在当前页面

Wait Until Page Contains Element	Xpath=//* [@]	42	error

Xpath=//* [@] ：表示元素定位，这里定位出现的元素

42 ： 表示最长等待时间。

Error ： 表示错误提示，自定义错误提示，如：“元素不能正常显示”

十一、获取 title

Get Title		

获得当前浏览器窗口的 title 信息。

这里只获取 title 是没有意义的，我们通常会将获取的 title 传递给一个变量，然后与预期结果进行比较。从而判断当前脚本执行成功。

十二、获取文本信息

Get Text	Xpath=//* [@]	

Xpath=//* [@] ： 定位文本信息的元素。

十三、获取元素属性值

Get Element Attribute	id=kw@name	

id=kw@name ： id=kw 表示定位的元素。@nam 获取这个元素的 name 属性值。

十四、cookie 处理

获取 cookie

get cookies		

获得当前浏览器的所有 cookie 。

获得 cookie 值

get cookie value	Key_name	

Key_name ： key_name 表示一对 cookie 中 key 的 name 。

删除 cookie

delete cookie	Key_name	

删除 key 为 name 的 cookie 信息。

删除所有 cookies

delete all cookies		

删除当前浏览器的所有 cookie。

添加 cookie

add cookie	Key_name	Value_name

添加一对 cooke （key：value）

十五、声明变量

${a}	Set Variable	hello

定义变量 a 为 hello。

${a}	${b}=	Set Variable	hello	world

定义变量 a 为 hello ， b 为 world 。

十六、日志（输出）

${a}	Set Variable	Hello World
log	${a}	

在测试报告中输出 a 变量的值 hello word。

十七、获得浏览器窗口宽、高

${width}	${height}	get window size
log	${width}	
log	${height}	

获得浏览浏览器窗口宽、高，通过 log 将宽高，打印到报告中。

十八、验证

open browser	http://www.baidu.com	chrome
${title}	Get Title	
should contain	${title}	百度一下，你就知道

Open Browser 通过 chrome 打开百度首页。

Get Title 获得浏览器窗口的 titile ， 并赋值给变量${title}

Should Contain 比较${title}是否等于“百度一下，你就知道”。

如果 item1 不包含 item2 一次或多次，那么失败。

十九、表单嵌套

Select Frame	Xpath=//* [@]	
Unselect Frame		

Select Frame 进入表单，Xpath=//* [@] 表示定位要进入的表单。

Unselect Frame 退出表单。

二十、下拉框选择

Unselect From List By Value	Xpath=//* [@]	vlaue

Xpath=//* [@] 定位下拉框；

Vlaue 选择下拉框里的属性值。

二十一、If 分支语句

${a}	Set variable	2		

${b}	Set variable	5		
run keyword if	${a}>=1	log	a 大于 1	
...	ELSE IF	${b}<=5	log	b 小于等于 5
...	ELSE	log	上面两个条件都不满足	

首先定义两个变量 a ， b 分别为 2 和 5 。

If 判断 a 大于等于 1 ，满足条件 log 输出 “a 大于 1 ”；

不满足上面的条件，接着 else if 判断 b 小于等于 5 ，满足条件 log 输出 “b 小于等于 5”；

上面两个条件都不满足，else log 输出“上面两个条件都不满足”。

备注：注意 sele if 和 else 前面的三个点点点（...）

二十二、for 循环语句

循环 1

:FOR	${i}	in range	10
	log	${i}	

查看结果：

循环变量 i 从 0 到 9 循环 10 次。

循环 2

@{a}	create list	aaa	bbb
:FOR	${i}	in	@{a}
	log	${i}	

@{a} 定义为一个字符串列表。
通过 in 可遍历非整型（in range）

说明：

Log 、if 分支，for 循环并非 selenium 关键字库的提供的方法，是由 BuiltIn 包提供。

五、Robot Framework 分层设计

谈到 Robot Framework 分层的思想，就不得不提“关键字驱动”。

关键字驱动： 通过调用的关键字不同，从而引起测试结果的不同。

在上一节的 selenium API 中所介绍的方法其实就是关键字，如“open browser” 就是一个关键字。从底层去看它就是一个通过编程去现实的一个方法。

```
def   open_browser(url，browser):
      #通过 browser 找到相应的浏览器驱动，调用浏览器，借助 python 的 httplib、urllib 模块将 url 传递给浏览器。从而实现 open brwoser  的目的。
```

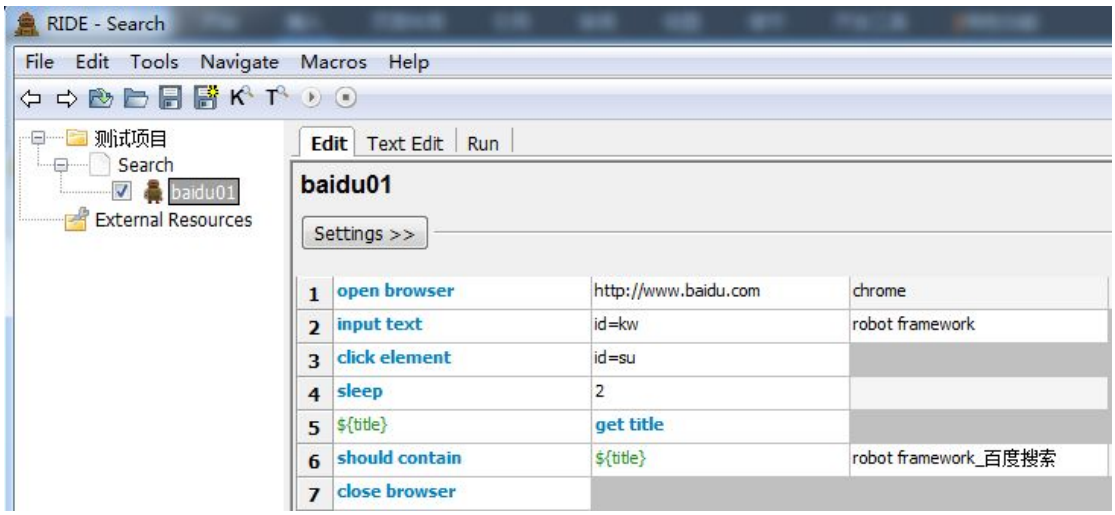
通过上面的伪代码表述的“关键字”的底层其实还是程序定义的方法。

回到分层的思想上，在程序设计的讲究设计模式，设计模式其实就是根据需求使用抽象与封装，其实就是分层思想。把一个实现过程分成不同多层。提高的灵活性，从而达到可扩展性和可维护性。

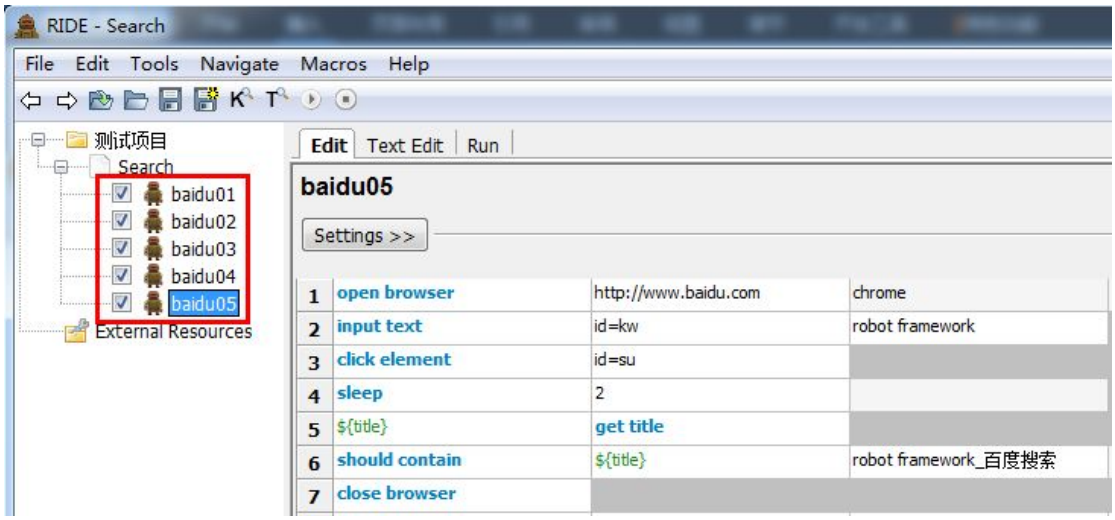
再回到自动化的话题上，我们可以把操作步骤封装一个一个的方法（关键字），通过调用关键字来实现测试用例。

http://www.cnblogs.com/fnng/p/3871712.html

参考本系列的第一节创建一条百度搜索的测试用例。



我现在要写五条百度搜索的用例：



可以在 Search 测试套件下创建 5 条测试用例。其实对于每一条测试用例来说，只是搜索的内容不同，脚本步骤是完全一样的。这样做无疑增加的脚本的冗余，而且不便于维护。假如，百度输入框的定位方式变了，我不得不打开每一条用例进行修改。

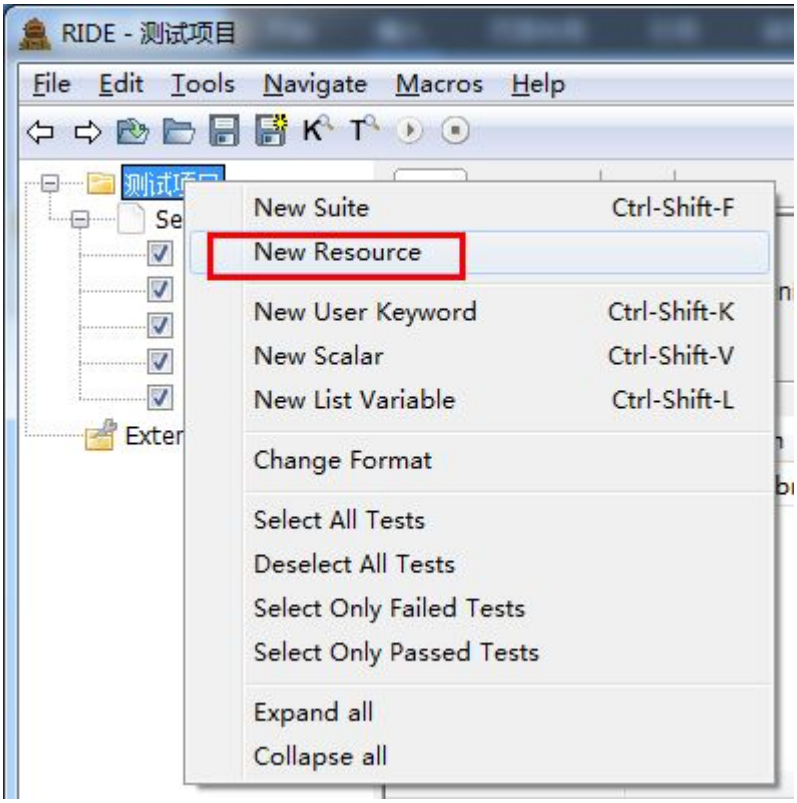
我们可以过创建关键字的方式，从而实现分层的思想来解决这个问题。

1、创建用户关键字

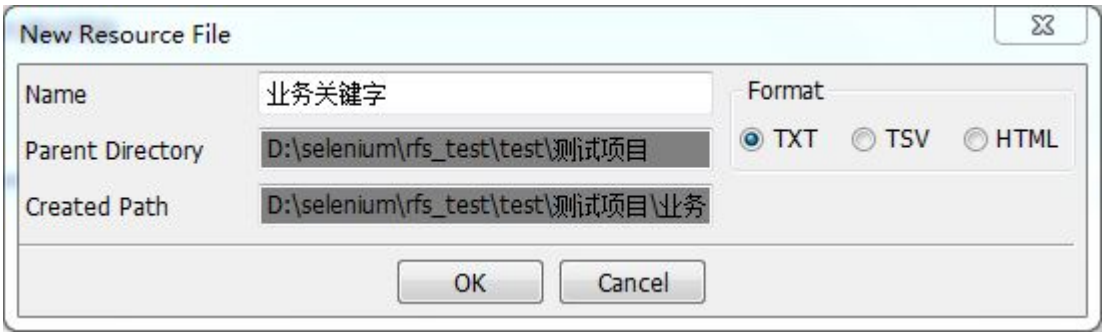
Robot Framework 创建关键字步骤：

1、创建资源

右键“测试项目”选择“new resource”创建资源。

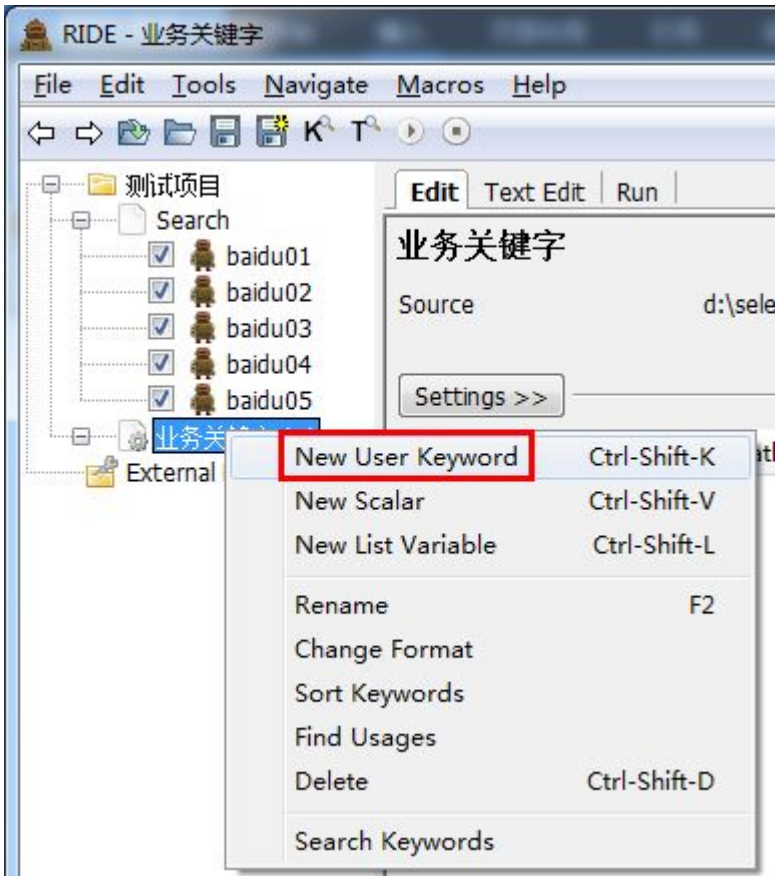


输入资源名称：

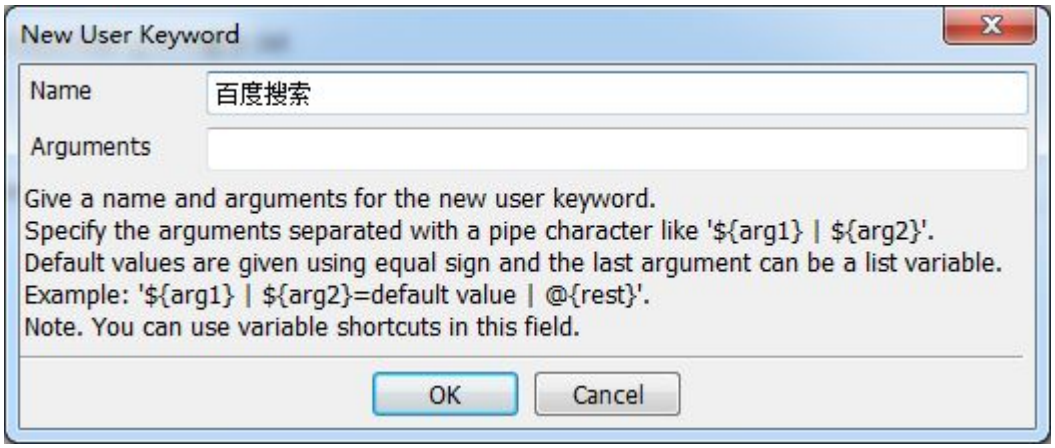


2、创建关键字

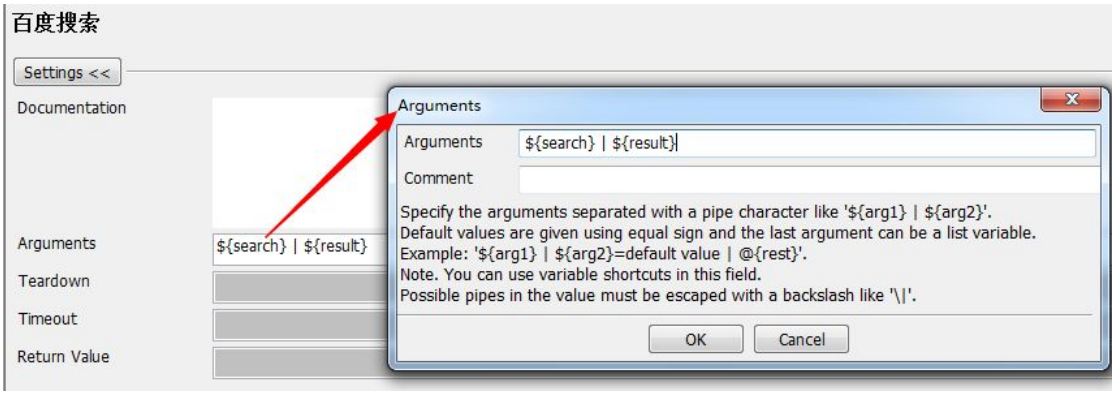
右键“业务关键字”选择“new User Keyword” 来创建用户关键字。



输入关键字的名称：



3、编辑关键字



分析：

对于一个测试用例来说，用户关心的是输入什么内容，得到什么结果。

所以，对于“百度搜索”关键字来说，需要创建两个接口变量${search}和${result} 两个变量，用于接收输入内容和预期结果。

点击 Arguments 输入框，定义变量，多个变量从用“|”隔开。

在百度用户中使用参数化变量。

1	**open browser**	http://www.baidu.com	chrome
2	**input text**	id=kw	${search}
3	**click element**	id=su	
4	**sleep**	2	
5	${title}	**get title**	
6	**should contain**	${title}	${result}
7	**close browser**		
8			

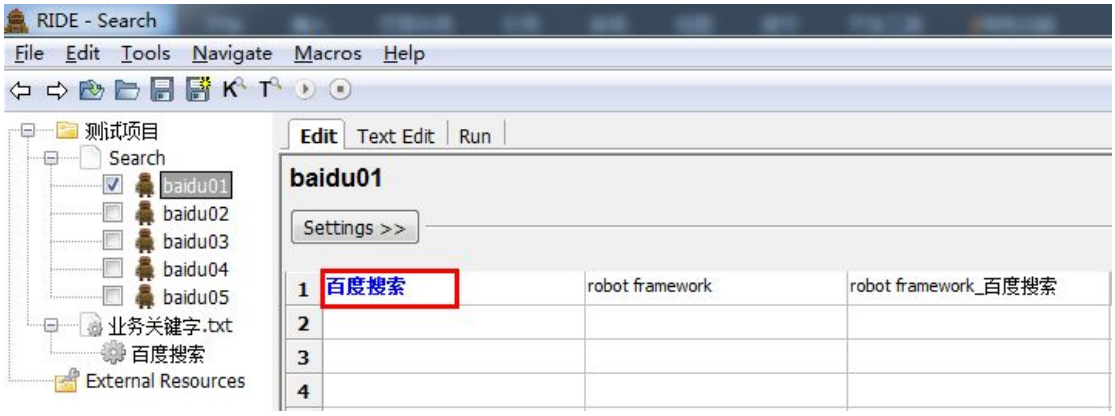
2、添加调用关键字

4、添加创建的资源

切换到测试套件（Search）页面，添加资源（业务关键字.txt）

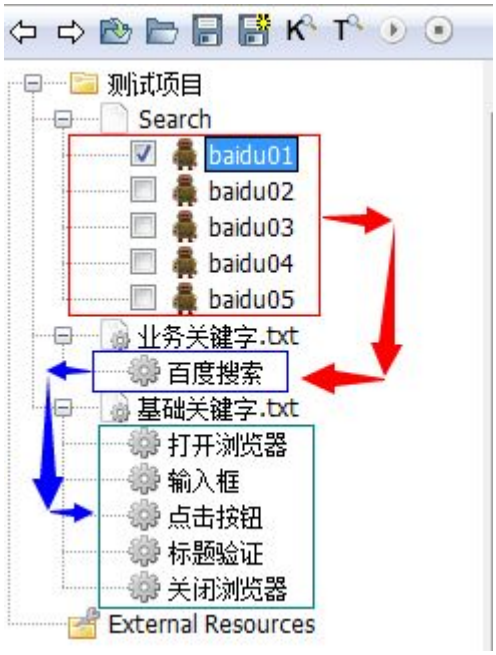
5、调用关键字

现在就可以在测试用例中使用创建的关键字了（百度搜索）。



对于每一条用例来说，调用“百度搜索”关键字，输入搜索内容，输入预期结果即可。不同关心用例是如何执行的。如果百度输入框的定位发生了变化，只用去修改“百度搜索”关键字即可，不用对每一条用例做任何修改。大大提高的用例的维护性和扩展性。

继续分层的设计：



到此，Robot Framework +selenium 自动化测试粗犷的讲完了。当然还有更多 API 的使用，和细枝末节的设置没有介绍。但我们已经可以拿它来开展自动化工作了。