Robot Framework+Selenium2 操作手册

目录

1. [环境搭建 3](#_bookmark0)
   1. [安装说明 3](#_bookmark1)
   2. [安装方法 3](#_bookmark2)
2. [Robot Framework 使用方法 10](#_bookmark3)
   1. [新建工程、案例区、资源区 10](#_bookmark4)
   2. [新建用户关键词 15](#_bookmark5)
   3. [新建测试用例 17](#_bookmark6)

[2.4 运行 18](#_bookmark7)

* 1. [日志展示 20](#_bookmark8)
  2. [报告展示 20](#_bookmark9)

1. [元素如何定位 22](#_bookmark10)
   1. [Xpath 基础知识 22](#_bookmark11)
   2. [根据 Xpath 进行元素定位 26](#_bookmark12)
2. [Selenium2Library 常用关键词 28](#_bookmark13)
   1. [浏览器交互 28](#_bookmark14)
   2. [frame 30](#_bookmark15)

[4.3 输入 31](#_bookmark16)

[4.4 点击 33](#_bookmark17)

[4.5 选择 34](#_bookmark18)

[4.6 校验 36](#_bookmark19)

* + 1. [Page 37](#_bookmark20)
    2. [Location 39](#_bookmark21)
    3. [Title 39](#_bookmark22)
    4. [Text 40](#_bookmark23)
    5. [Element 41](#_bookmark24)
    6. [List 43](#_bookmark25)
    7. [Checkbox 43](#_bookmark26)
    8. [Radio Button 44](#_bookmark27)
    9. [Table 44](#_bookmark28)

[5 示例 47](#_bookmark29)

* 1. [“打开首页”关键词 47](#_bookmark30)
  2. [“用户名密码”关键词 47](#_bookmark31)
  3. [“点击登录”关键词 49](#_bookmark32)
  4. [“验证登录是否成功”关键词 50](#_bookmark33)
  5. [“登录成功后续处理”关键词 51](#_bookmark34)
  6. [“关键字的二次封装”测试用例 51](#_bookmark35)

# 环境搭建

## 安装说明

1、请根据自己计算机下载相对应的版本，下文中的参考版本仅供参考。

2、如果是 exe 的直接安装就好了。

3、对于只有源代码的测试库（只有 py 文件）的安装，请在进入 cmd 命令行后，进入测试库的目录（有 setup.py 的目录），输入 python setup.py install 进行安装。

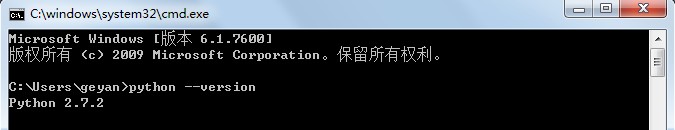
如果提示 python 不是命令，请把 python 的安装目录加到环境变量的 path 中。在 path 中增加 D:\Python27 和 D:\Python27\Scripts 两个路径。

如果安装报错，请以管理员身份运行命令提示符后，再输入 python setup.py install 进行安装。

## 安装方法

1、Python 环境（ActivePython）

地 址 ：<http://www.activestate.com/activepython/downloads>参考版本：ActivePython-2.7.2.5-win32-x86.msi

说明：直接安装就 ok，不用配置环境变量，安装时自动配置好，而且还有一 些 其 他 实用 工 具 包含 在 里 面了 。 验 证安 装 是 否成 功 ， 命令 行 打python--version，出现如下图提示。

2、wxpython（ride 要有他支持）

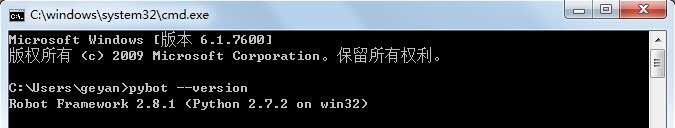
地 址 ： <http://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/>参考版本：wxPython2.8-win32-unicode-2.8.12.1-py27.exe

说明：下载对应的 unicode 的版本，一路下一步就ok 了

3、Robot Framework

地 址 ：<http://code.google.com/p/robotframework/downloads/list>参考版本：robotframework-2.8.1.tar.gz

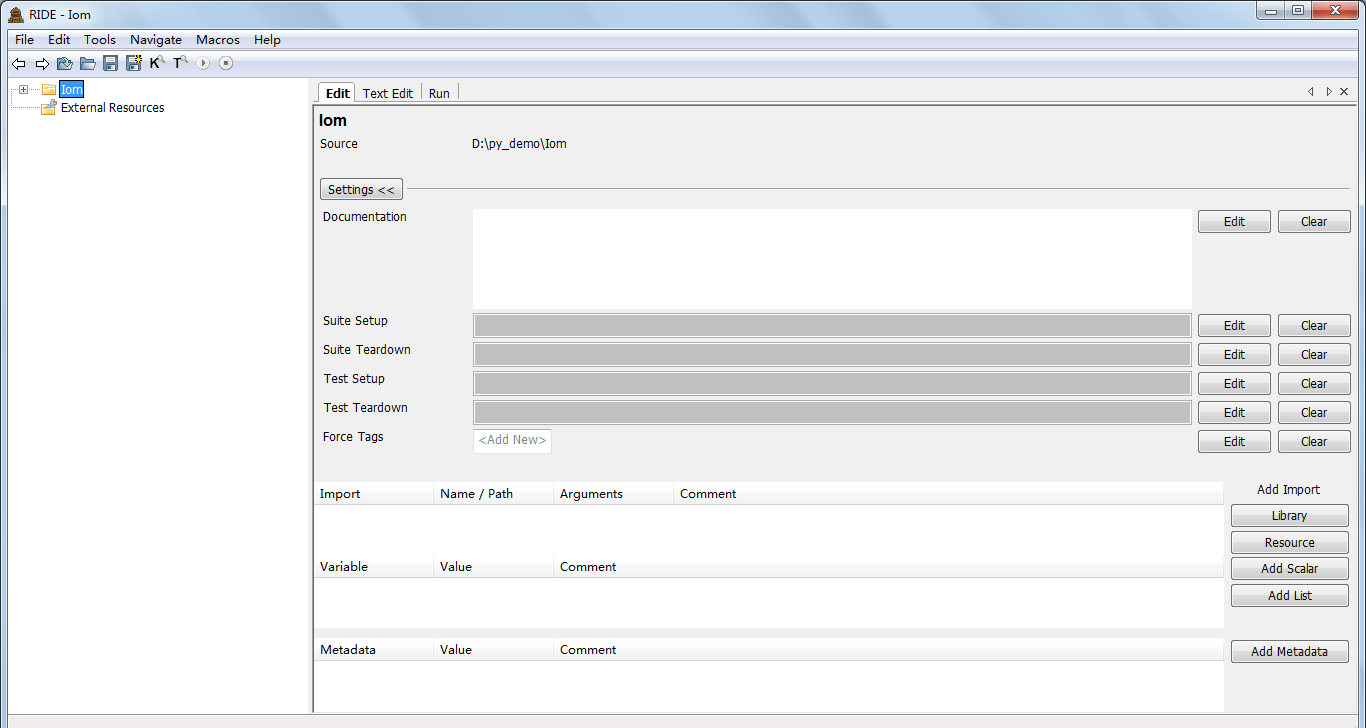
说明：验证安装是否成功，命令行打 pybot--version，出现如下图提示。



4、Robot Framework ride 地址：

<http://code.google.com/p/robotframework-ride/downloads/list>参考版本：robotframework-ride-1.3.win32.exe

说明：验证安装是否成功，打开 D:\Python27\Scripts\ride.py，或是通过桌面产生的 RIDE 快捷方式，出现下图界面。



5、decorator-3.4.0（S2L 库的依赖库）

地 址 ：<https://pypi.python.org/pypi/decorator/3.4.0>参考版本：decorator-3.4.0.tar.gz

6、docutils-0.11（S2L 库的依赖库）

地 址 ：<https://pypi.python.org/pypi/docutils/0.11>参考版本：docutils-0.11.tar.gz

7、selenium-2.35.0（是 selenium 的原生 python 库）

地 址 ：<https://pypi.python.org/pypi/selenium/2.35.0>参考版本：selenium-2.35.0.tar.gz

8、robotframework-selenium2library （selenium2 测试库，基于

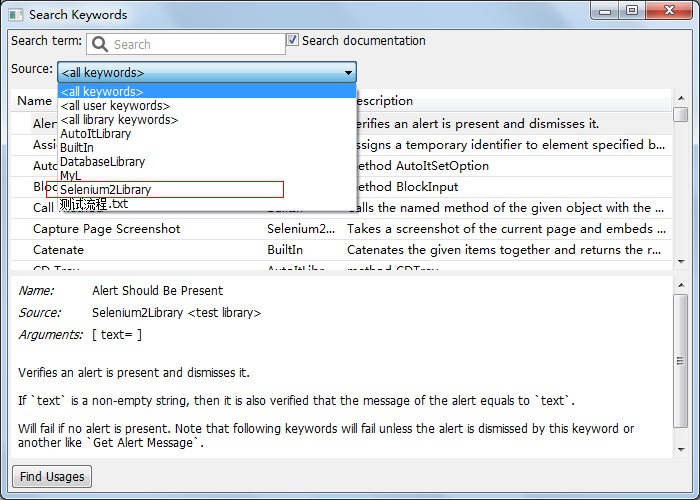
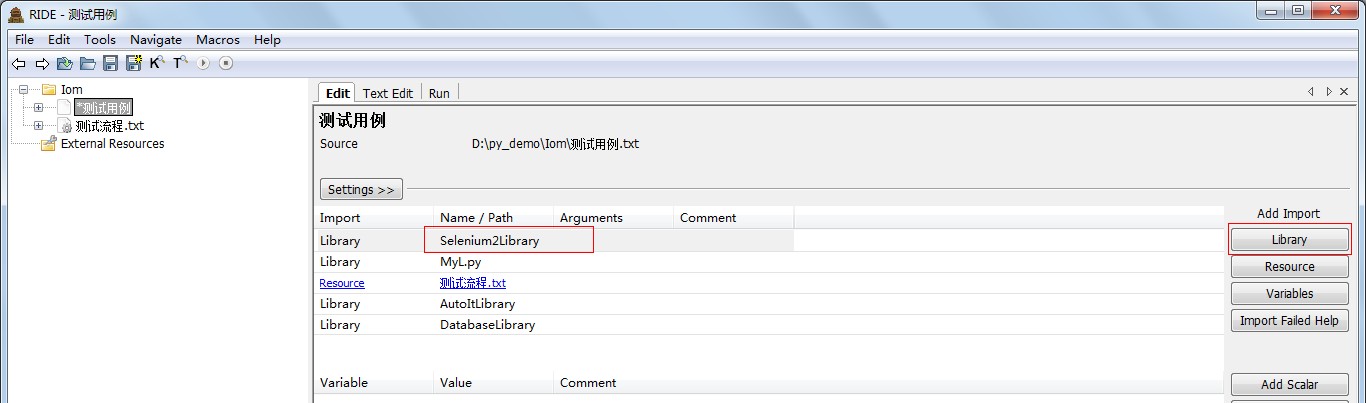
webdriver 驱动） 地址：

<https://github.com/rtomac/robotframework-selenium2library/releases>

参考版本：robotframework-selenium2library-1.1.0.tar.gz

说明：随意新建一个工程，一个 suite，一个 case，然后试着在 suite 中添

加一个 library，写 Selenium2Library（注意大小写！），确定之后，不是红色，

说明已经成功了！按 F5，看看 Source 里是否有Selenium2Library。

9、cx\_Oracle（安装 DatabaseLibrary）

地 址 ：<http://sourceforge.net/projects/cx-oracle/files/>参考版本：cx\_Oracle-5.1.2-11g.win32-py2.7.msi

说明：目前最新 5.1.2，注意要和你本地的 oracle 客户端版本一致，如果你的 oracle 是 10g 的就去下载 10g 的文件。

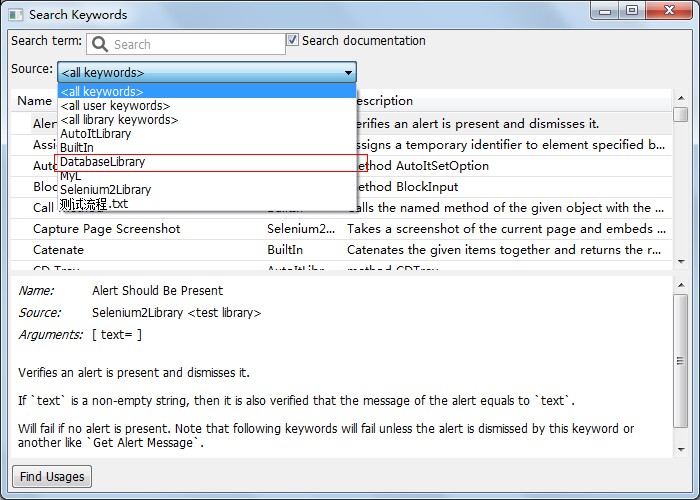
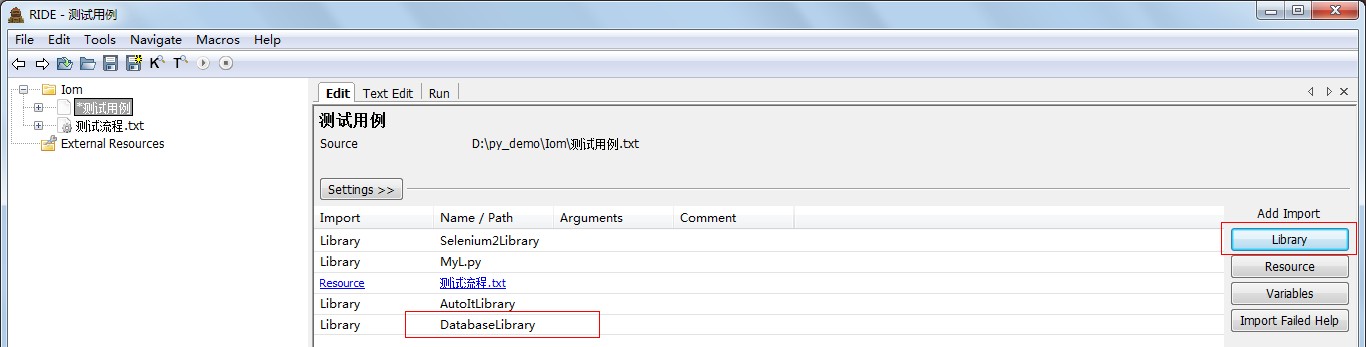
10、DatabaseLibrary（安装 DatabaseLibrary）

地址：<http://franz-see.github.io/Robotframework-Database-Library/>

参考版本：

franz-see-Robotframework-Database-Library-0.6-3-g216432f.tar.gz 说明：在 suite 中添加一个library，写 DatabaseLibrary（注意大小写！），

确定之后，不是红色， 说明已经成功了！ 按 F5 ，看看 Source 里是否有

DatabaseLibrary。

11、pywin32（安装 AutoItLibrary）

地址：<http://sourceforge.net/projects/pywin32/files/pywin32/>

参考版本：pywin32-219.win32-py2.7.exe

说明：版本较多，请注意py 版本

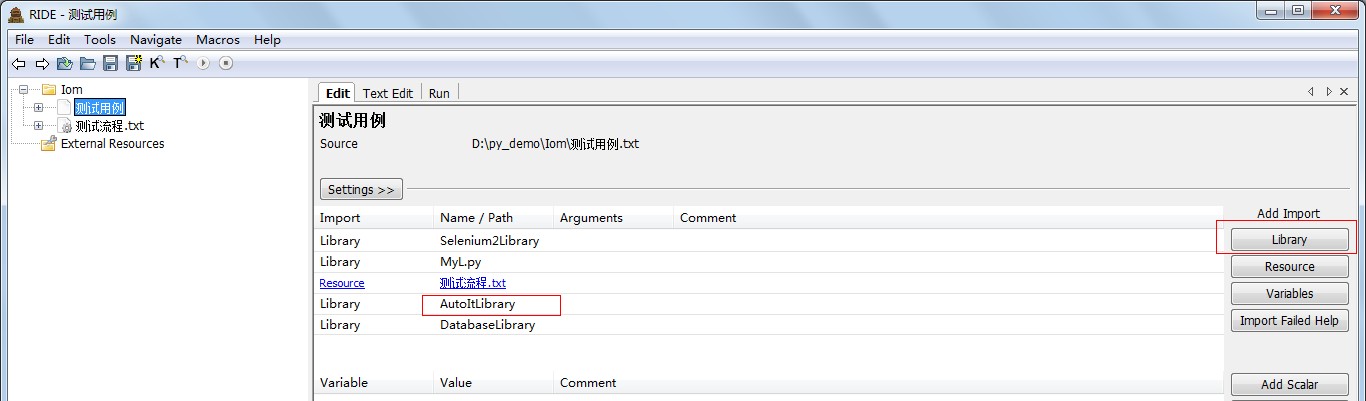
12、AutoItLibrary-1.1

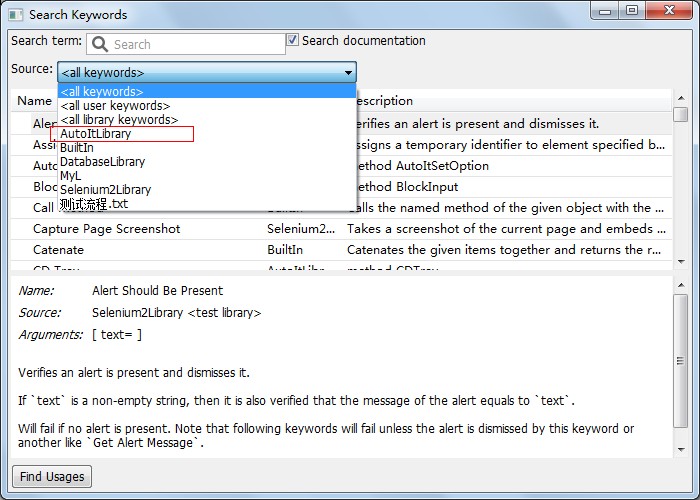
地 址 ：<http://code.google.com/p/robotframework-autoitlibrary/>参考版本：AutoItLibrary-1.1.zip

说明：必须先安装上面的 pywin32，并且 Python 的安装目录不能有空格， 如果有空格会导致注册 autoit 的 dll 时出错。64 位机器需要在再安装 AutoItV3 才能用，官网下载地址：

<http://www.autoitscript.com/site/autoit-news/autoit-v3-3-10-0-rele> ased/

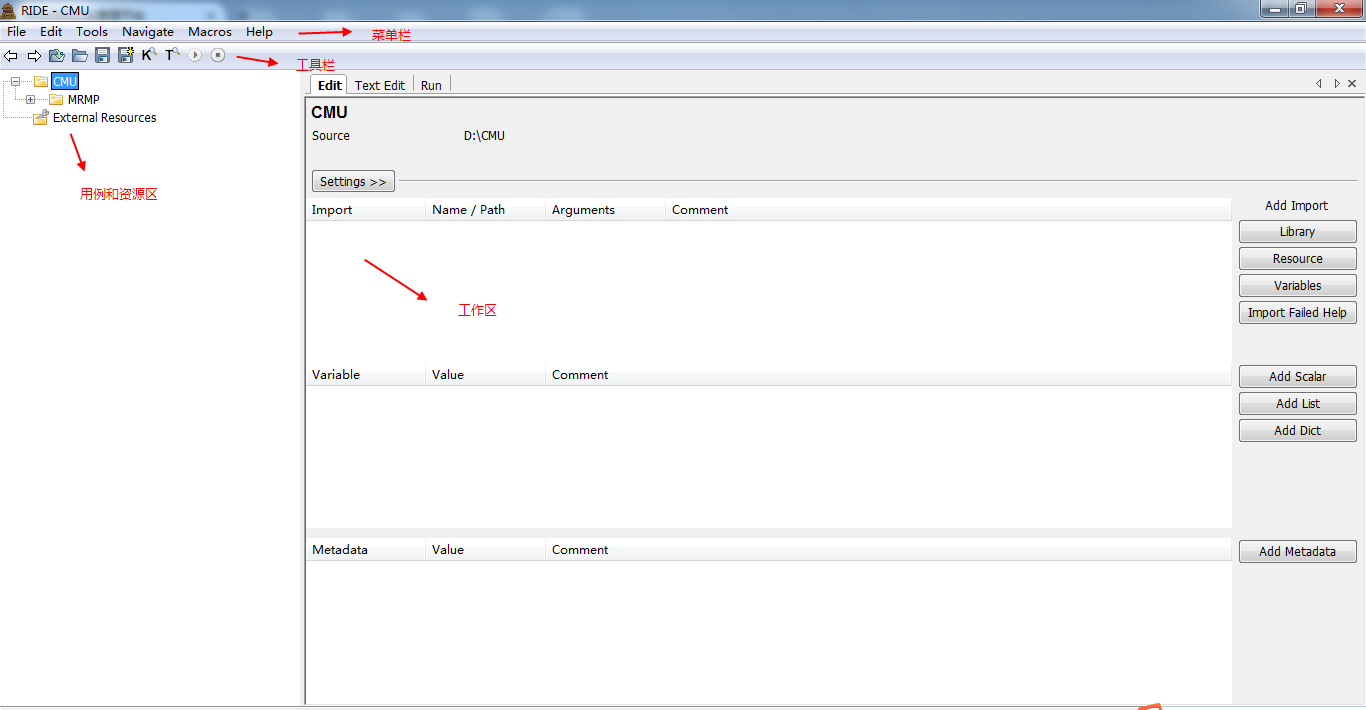
在 suite 中添加一个 library，写 AutoItLibrary（注意大小写！），确定之后，

不是红色，说明已经成功了！按F5，看看 Source 里是否有 AutoItLibrary。



# Robot Framework 使用方法

RIDE 是 Robot Framework 的 IDE，界面如下。



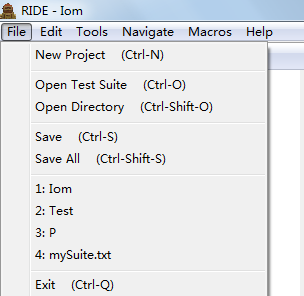
从界面左侧可以看出 IOM 项目下包括2 个区：案例区和资源区。案例区“测试用例”下包括具体的测试用例。资源区“测试流程.txt”下包括用户关键词， 每个用户关键词里写有一个页面功能，相当于一个函数，可以在关键词中互相调用，也可以被不同的测试用例进行调用，以组合成不同的测试场景。

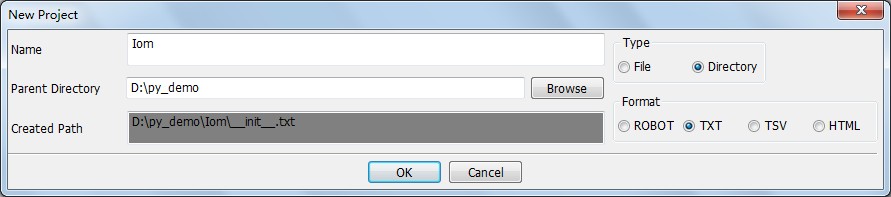
## 新建工程、案例区、资源区

1、点击 File—New Project，新建一个 Project

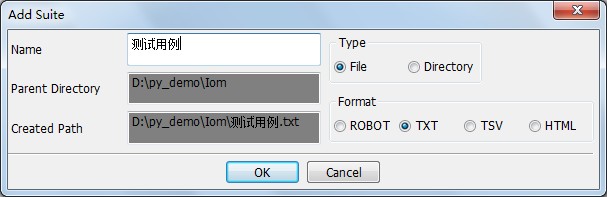
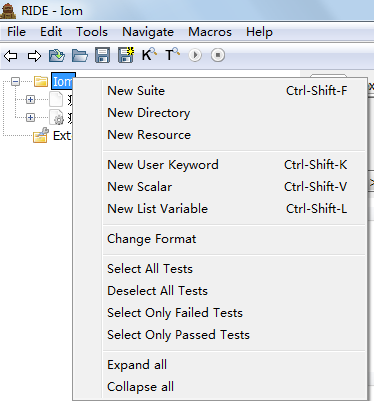
创建测试项目，测试项目可以目录或文件存储，格式可以用 ROBOT，TXT，

TSV 或 HTML 格式，建议选择目录和 TXT，这样便于管理。

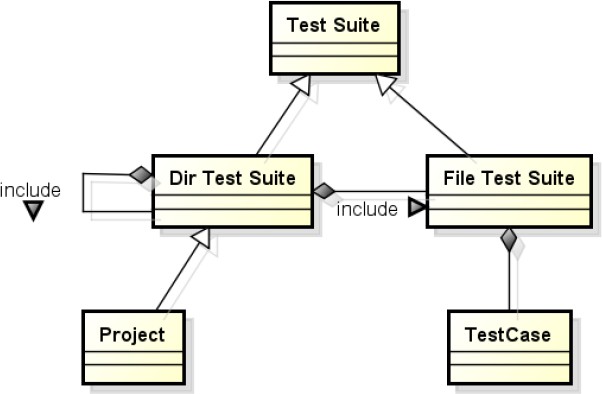


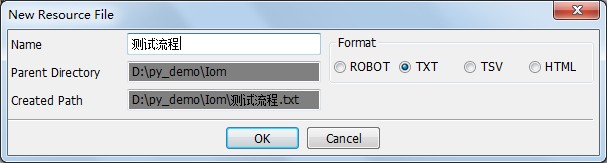
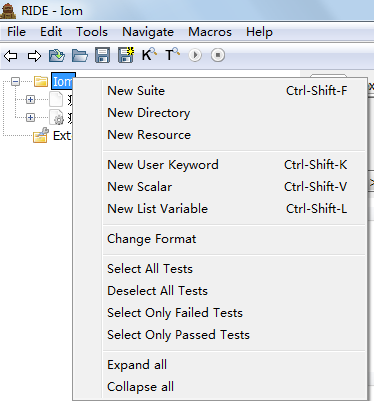


2、在新建的项目 Iom 上右键选择New Suite，新建案例集



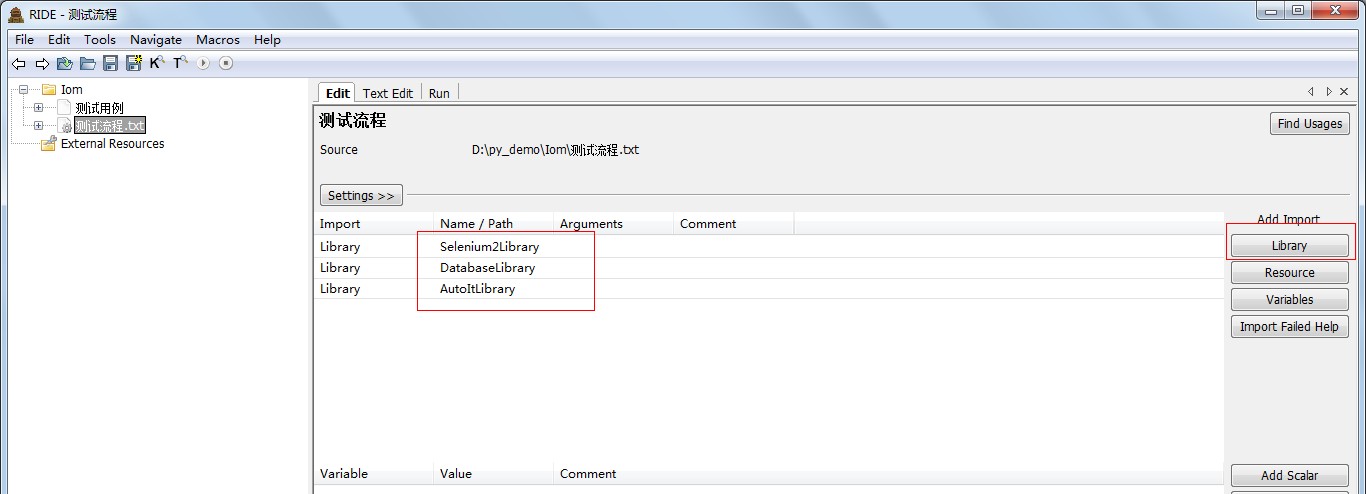
案例集可以选择是文件还是目录，层级关系如下图



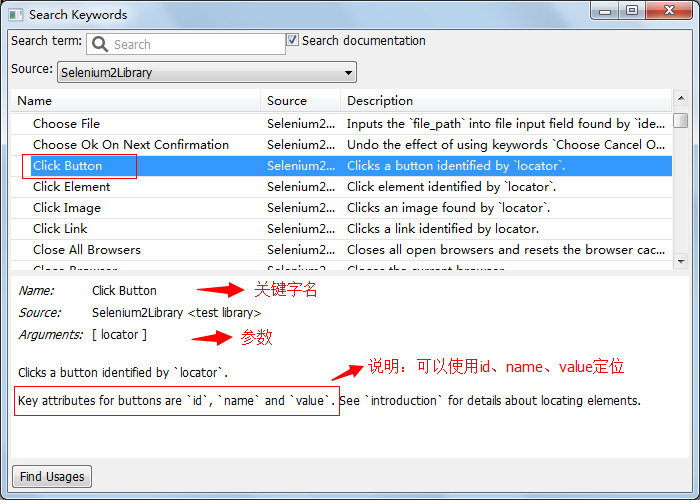
3、在 Iom 上右键选择 New Resources，新建资源。（由于测试用例需要使用资源中的关键词，所以我们先新建需要的 User Keyword）

4 、在这里需要加载测试库 Selenium2Library 、DatabaseLibrary 、AutoItLibrary ，加载后在测试流程.txt 下 User Keyword 就可以使用

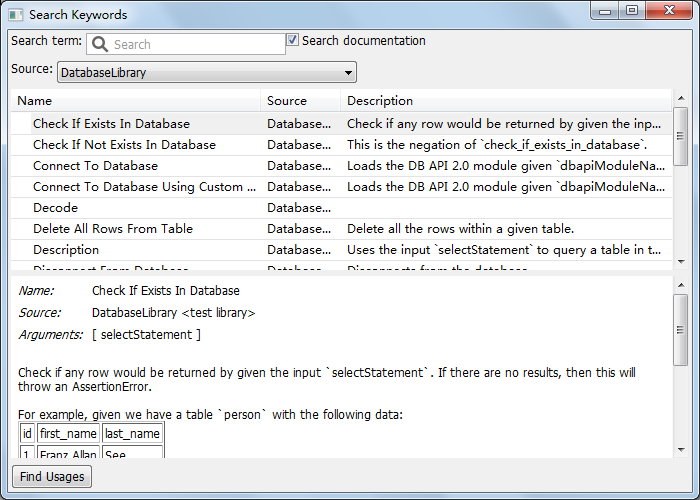
Selenium2Library、DatabaseLibrary、AutoItLibrary 中提供的关键词了，加

载方法见下图。

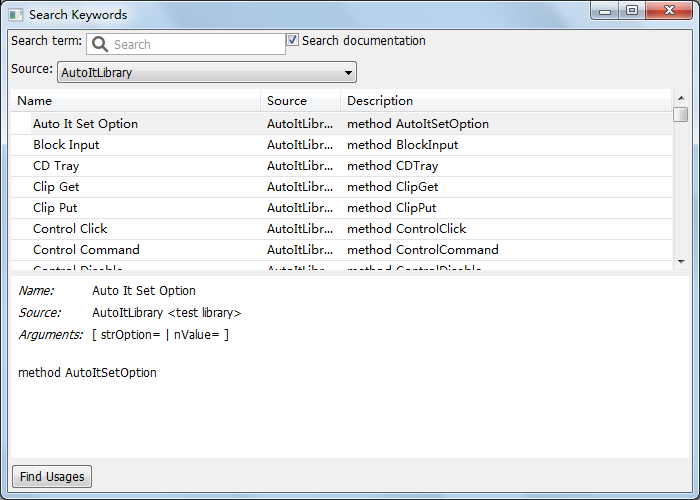
如果加载成功，按 F5，Source 选择 Selenium2Library



Source 选择 DatabaseLibrary

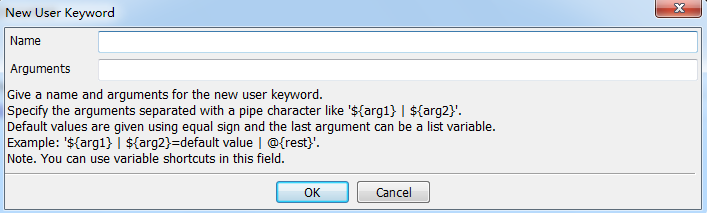


Source 选择 AutoItLibrary

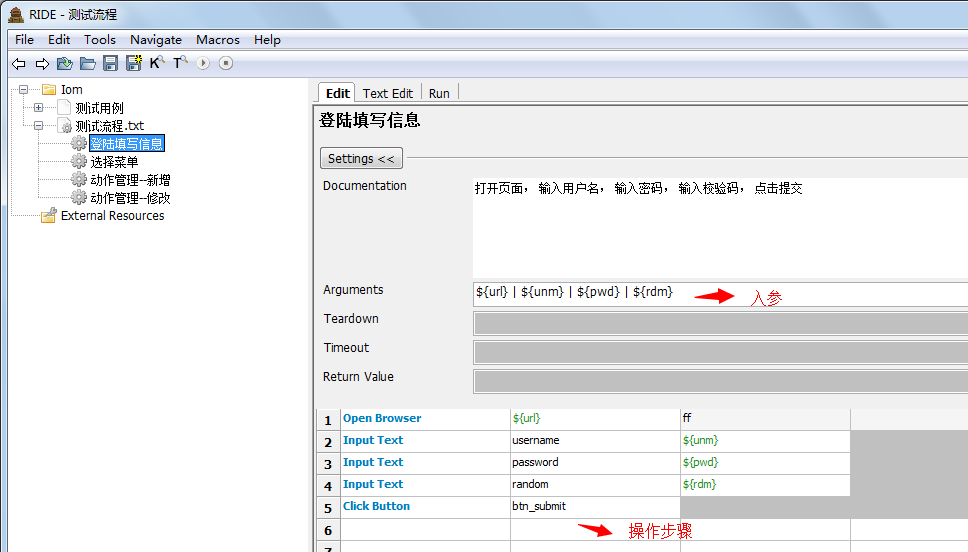


## 新建用户关键词

1、在测试流程.txt 上右键选择 New User Keyword



2、在登录填写信息中，编写操作步骤脚本



（1）Arguments: 设置传入参数

* + - 变量类型

RF 里的变量分两种，一种是 Scalar，可以理解为单值变量，一种是 List

Variable，list 型变量，或者可以理解为数组变量。Scalar 型变量用$符号开头，

List 型变量用@符号开头。

* + - 必填参数

给关键词先添加 4 个参数，参数之间使用|进行分隔。（如果想在默认值里使

用|作为一个值而不是分隔符，那么就要使用 \| 来表示|的值）

* + - 选填参数

假设我们想把其中的一个参数设成选填参数，选填参数就是给参数加上默认值，默认值使用=加上值，如果想默认为空，只写等号就行了。${rdm}=8888， 如果某一个参数设置成了选填参数，那么他后面的参数都必须是选填参数，不能 是必填参数。

* + - List 变量参数

List 变量也是可以作为参数，但是List 变量只能放在最后一位。如果放在前面，就会报错。

如果把${pwd}改成了@{pwd}这样的 List 变量，保存时就会提示 List 变量只能作为最后一个参数。

首先，List 变量本身来说，他是一个可变的，即 List 的成员数量不确定。而

List 作为参数的话，有几个成员就是相当于几个单值参数，他实际上是提供了一种参数个数可变的方式。那么既然可变，如果放在前面的话，他就没法确定传入的参数到底哪些是给 pwd 的，哪个是给 rmd 的了；同理，如果放 2 个 List 也是不行的，因为也是无法区分传入的参数到底哪些是给 pwd，哪些是给 rmd。所以最终限定是只能有 1 个 List 参数，并且必须放在最后。

1. Teardown: 设置完成时的动作

比如写上 Close All Browsers，表示在这个用户关键词执行完成之后会执行什么关键词。可以理解为关键词完成后还要调一个关键词。参数要用|分隔。

1. Timeout: 设置超时时间

如写上 1min，表示 1 分钟超时，如果这个关键词执行超过 1 分钟则认为失

败。

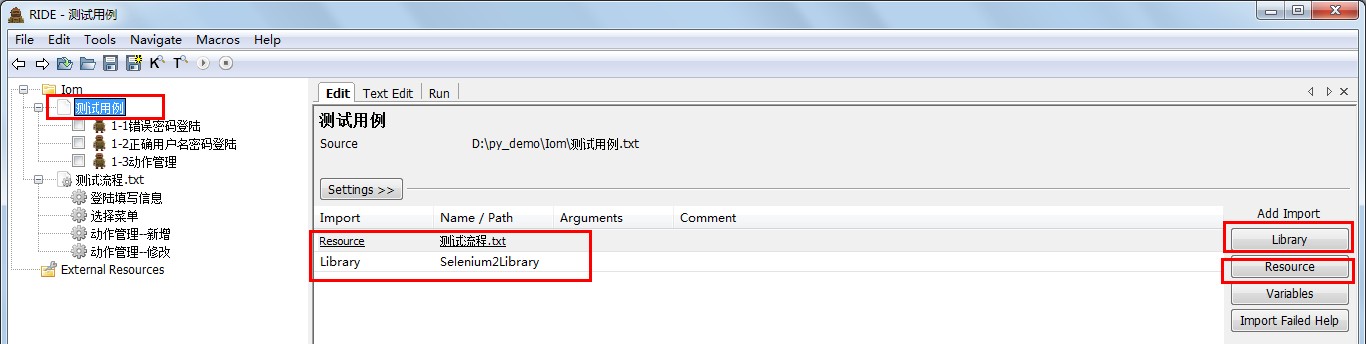
1. Return Value: 设置返回值

是支持多个变量返回的，也是用|分隔的。

（5）表格步骤中左侧蓝色的关键词是Selenium2Library 测试库的关键词， 具体用法可以使用 F5 查看。

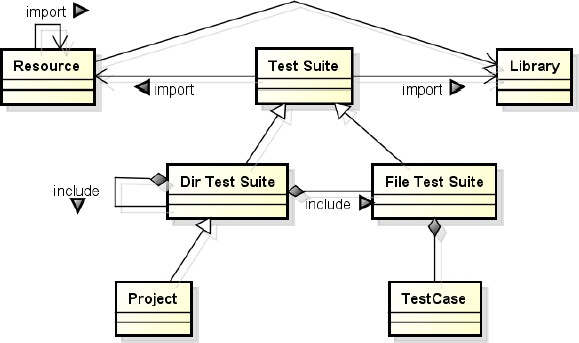
## 新建测试用例

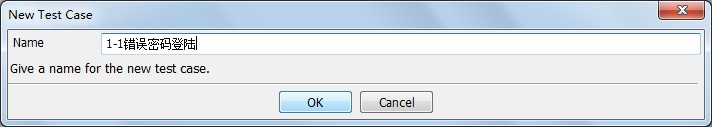
1、接下来，在测试用例里加载测试库“Selenium2Library”和资源“测试流程.txt”



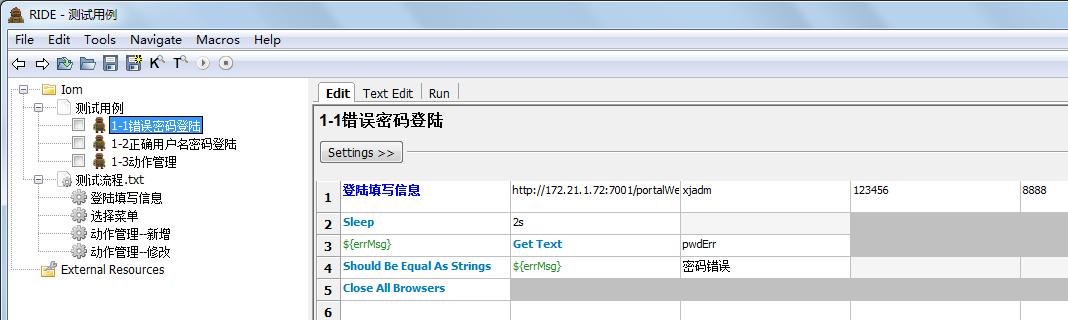
在做自动化案例的时候，用的最多的是关键词。关键词大体上可以分为测试库关键词（或叫系统关键词）和用户关键词，前者一般都是通过加载Library 得到的，后者用户关键词一般都是通过加载 Resouce 得到的。

关系如下图所示：



2、右键选择 New Test Case，在 Test Suite 中添加用例

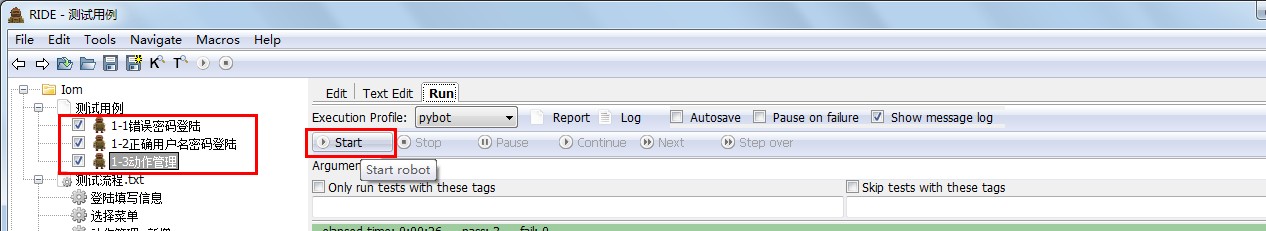
3、在用例中添加步骤



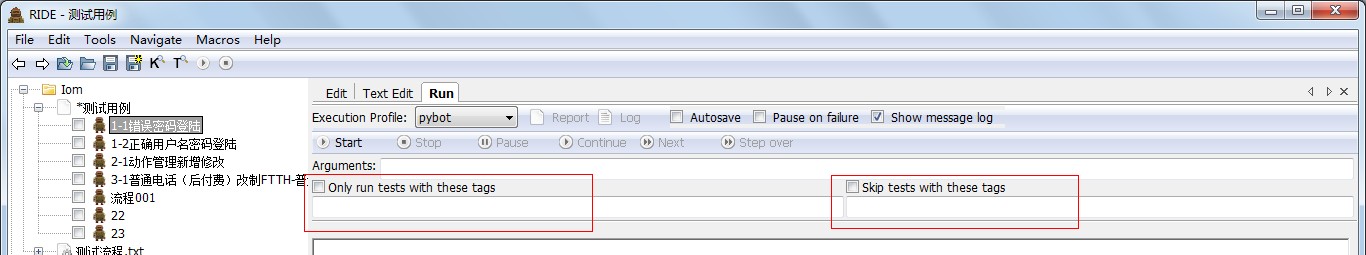
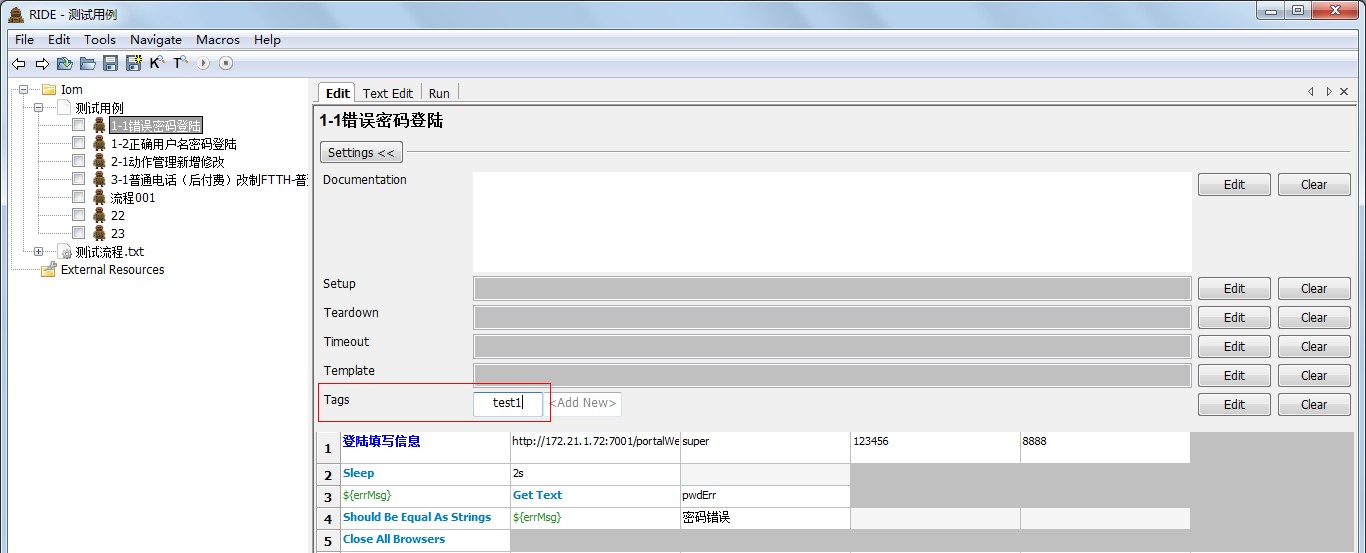
其中，“登录填写信息”是调用了刚才写的用户关键词，后面的 4 个值，为此关键词需要的入参，这样就保障了数据和流程分离。后面的步骤是获取密码错误的提示信息，用于校验。

用户关键词相当于作为公用部分提取出来，以供在测试用例中不同场景中不同参数组合调用。

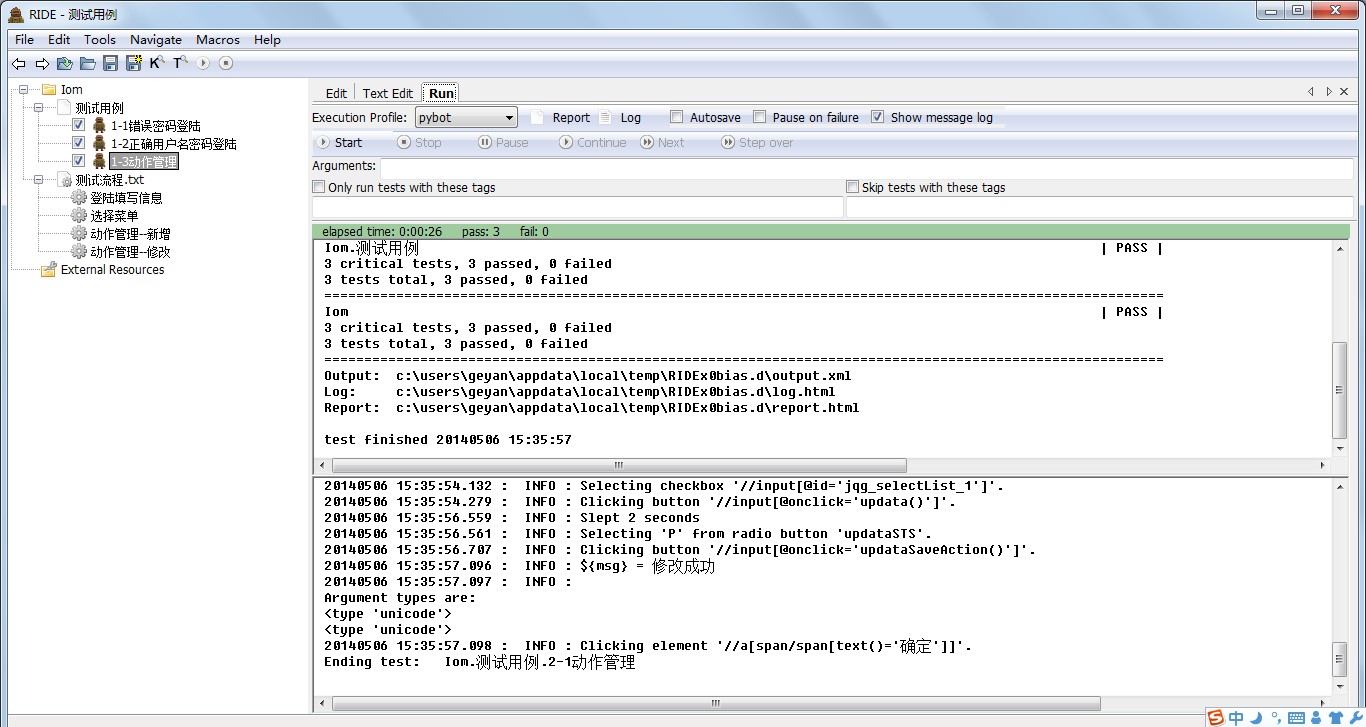
## 运行

勾选需要运行的脚本，点击 Start。

测试用例可以标记多个 Tag，可以按 Tag 来执行或跳过用例

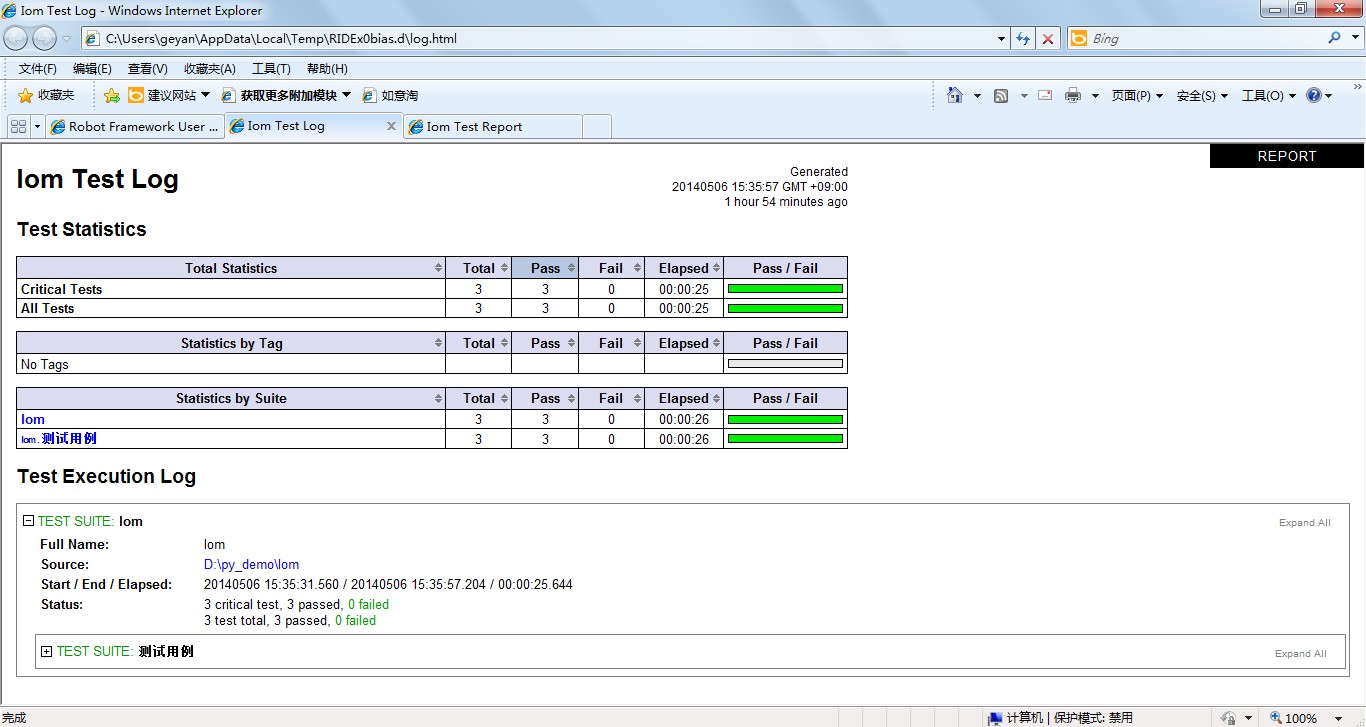


运行结果展示

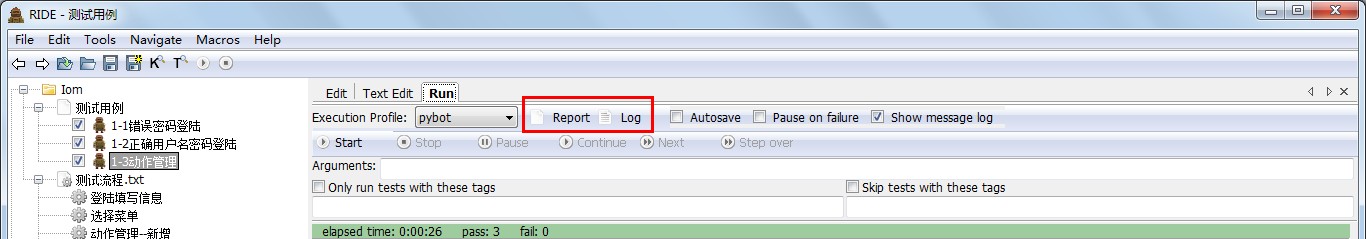


## 日志展示

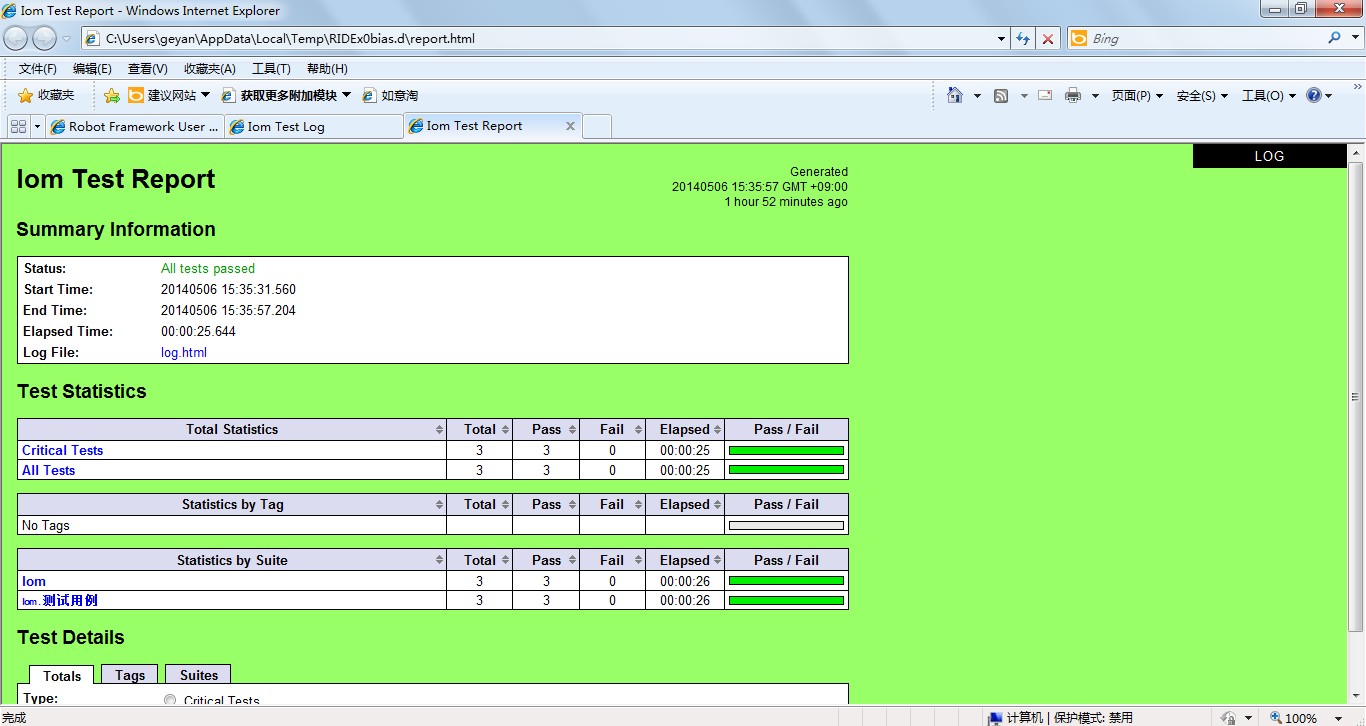
选择 Log 可以查看日志



## 报告展示

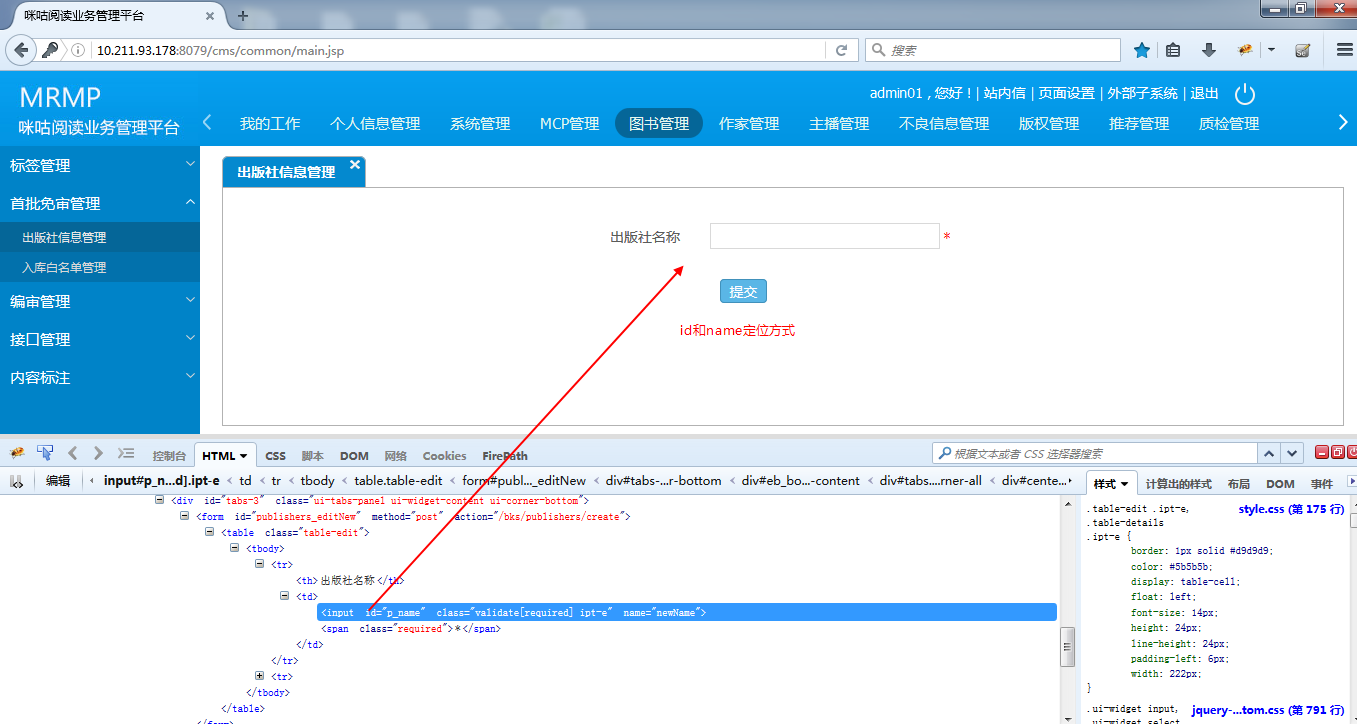


选择 Report 可以查看报告



# 元素如何定位

元素的定位，可以先通过 firefox 的 firebug，或者 IE 的 F12 进行查看。在需要定位的元素上右键选择“使用 firebug 查看元素”。

然后在 Selenium2 语句中，通过指定 id=username、name=username， 或者 xpath=//input[@id=’username’]进行定位。

## Xpath 基础知识

Xpath 是在 XML 文档中查找信息的一种语言。使用路径表达式来选取 XML

文档中的节点或节点集。由于 HTML 和 XML 结构类似，所以 Xpath 也用于查找 HTML 文档中的节点。

（1）/AAA，选择根元素 AAA

<AAA>

<BBB/>

<CCC/>

<DDD>

<BBB/>

</DDD>

<CCC/>

</AAA>

（2）/AAA/CCC，选择 AAA 的子元素 CCC

<AAA>

<BBB/>

<CCC/>

<DDD>

<BBB/>

</DDD>

<CCC/>

</AAA>

（3）/AAA/DDD/BBB，选择 AAA 的子元素DDD 的子元素BBB

<AAA>

<BBB/>

<CCC/>

<DDD>

<BBB/>

</DDD>

<CCC/>

</AAA>

（4）//BBB，选择所有 BBB 元素

<AAA>

<BBB/>

<CCC/>

<DDD>

<BBB/>

</DDD>

<CCC/>

</AAA>

（5）/AAA/BBB/\*，选择 AAA 的子元素 BBB 下的所有子元素

<AAA>

<BBB>

<CCC/>

<DDD/>

</BBB>

</AAA>

（6）/AAA/BBB[1]，选择 AAA 下的第一个子元素 BBB

<AAA>

<BBB/>

<BBB/>

<BBB/>

</AAA>

（7）//BBB[@id]，选择所有有 id 属性的 BBB 元素

<AAA>

<BBB id=“b1”/>

<BBB id=“b2”/>

<BBB name=“bbb”/>

</AAA>

（8）//BBB[@id=‘b1’] ，选择所有有 id 属性，并且值为b1 的 BBB 元

素

<AAA>

<BBB id=“b1”/>

<BBB id=“b2”/>

<BBB name=“bbb”/>

</AAA>

## 根据 Xpath 进行元素定位

1、依靠自己属性，文本定位

//input[@name='abc']

//input[@type='radio' and @value='1']---ie 不支持

//input[@type='radio'][text()='abc']---ie 不支持

//td[text()='abc']

//div[contails(@class, 'abc')]

//input[starts-with(@id,'abc ')]

//span[not(contains(text(),'abc'))]

//input[not(@id='abc')]

//div[@id='abc']/span[1]

//span[substring(@name,3,5)= 'abc']

<span class="spanclass-789>test</span>

//span[substring-before(@class,'-')='spanclass']

<span class="780-spanclass2>test</span>

//span[substring-after(@class,'-')='spanclass2']

2、依靠父节点定位

//div[@class='abc']/div

3、依靠子节点定位

//div[div[@id='abc']]

4、混合型

//div[div[@name='abc']]/img

<div>

<input id="abc">

<input>

</div>

//input[@id='abc']/following-sibling::input

<div>

<span>test</span>

<input id="abc">

</div>

//input[@id='abc']/preceding-sibling::span

# Selenium2Library 常用关键词

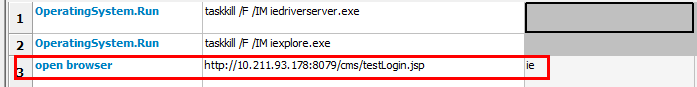
Selenium Library 将 Selenium2 与 Robot Framework 连接起来。

## 浏览器交互

1、打开页面：Open Browser

参 数 ： [ url | browser=firefox | alias=None | remote\_url=False | desired\_capabilities=None | ff\_profile\_dir=None ]

返回值：the index of this browser instance

说明：参数中有“=”的参数表示缺省时的默认值。截图中背景色白色表示必填，灰色为选填。

也可以将返回的 index 存在变量 a 中，用于指定 index 值进行页面跳转。



2、跳转页面：Switch Browser 参数：[ index\_or\_alias ]

说明：可以根据 index 或 alias 定位需要跳转的页面



3、获取窗口 name：Get Window Names

参数：[ ]

返回值：Returns and logs names of all windows known to the browser.



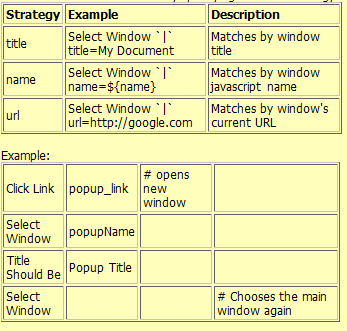
4、跳转窗口：Select Window 参数：[ locator=None ]

说 明 ：By default, when a locator value is provided, it is matched

against the title of the window and the javascript name of the window. If multiple windows with same identifier are found, the first one is selected.

Special locator `main` (default) can be used to select the main

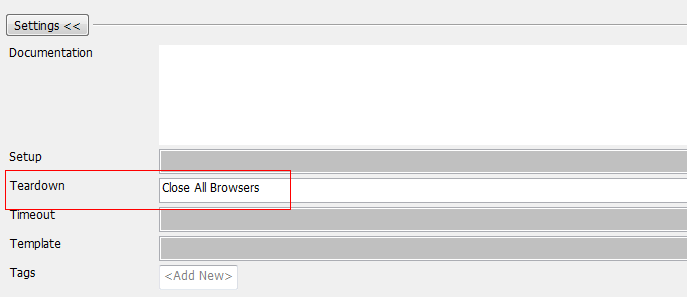
window.

It is also possible to specify the approach Selenium2Library should take to find a window by specifying a locator strategy:



5、关闭页面：Close All Browsers 参数：[ ]

说明：关闭所有打开的页面，此操作后再执行“Open Browser ”关键词

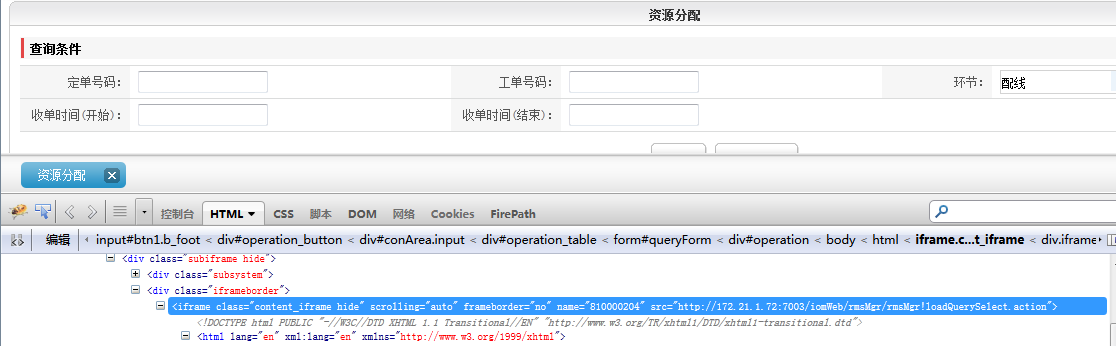
返回的 index 将被从 1 开始重新计数 。此关键词可以用于 test 或 suite 中teardown，确保关闭所有 browser。

## frame

有时需要先进入一个 frame 或 iframe 再定位一个元素

1、进入一个 frame 或 iframe：Select Frame 参数：[ locator ]

说明：可以通过 frame 的 id、name 或xpath 定位。如果需要定位的 frame 在当前 frame 内，可以直接使用 Select Frame 定位。





2、回到最上层 frame 或 iframe：Unselect Frame 参数：[ ]

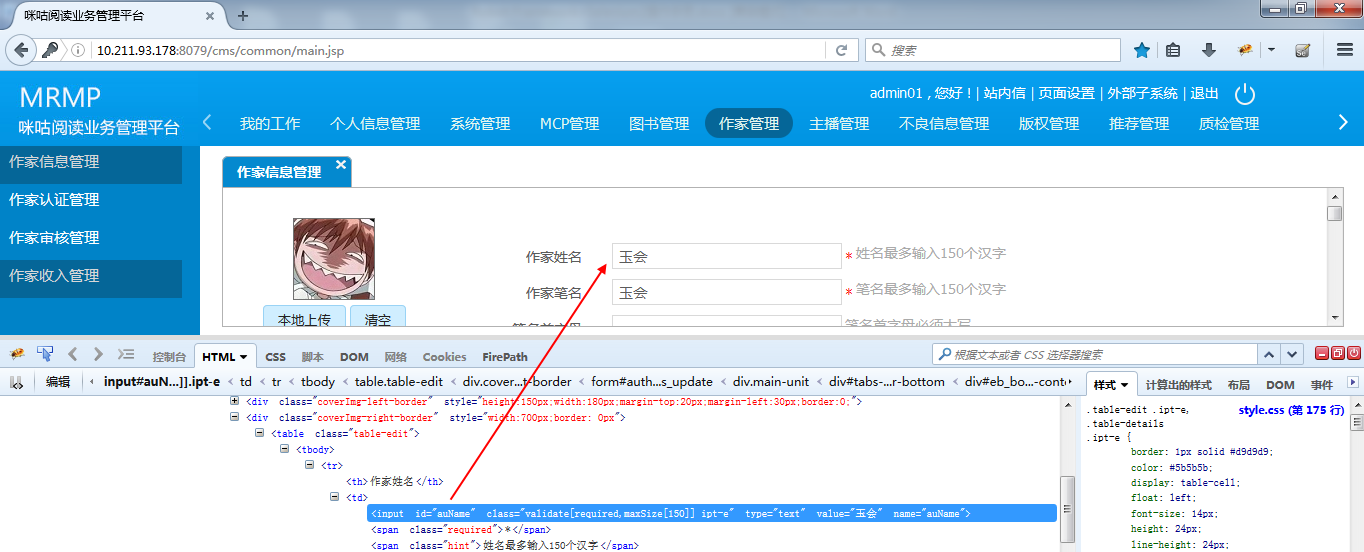
说明：如果需要定位的 frame 在当前 frame 外，或是平级关系，可以直接使用 Unselect Frame 先设置最外层的 frame 为当前 frame，然后在使用 Select

Frame 进行定位。

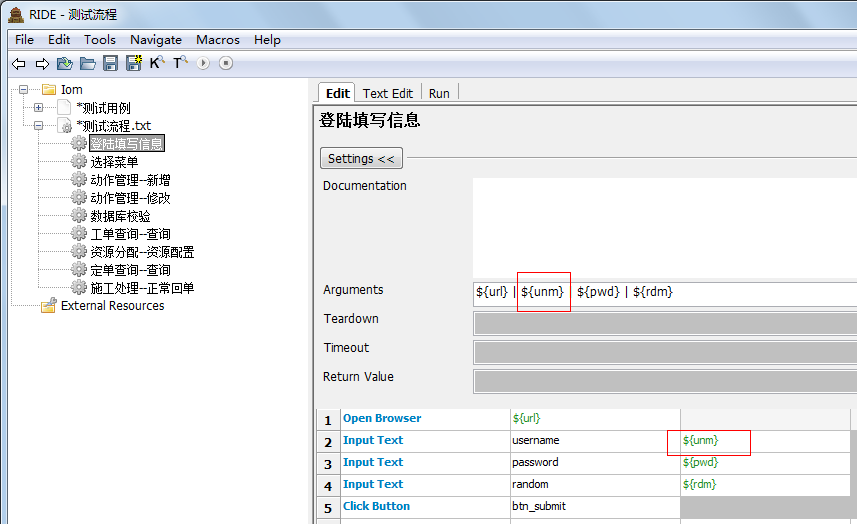
## 输入

1、文本框输入值：Input Text 参数：[ locator | text ]

说明：username 为“用户名”元素对应的 input 标签中的 id 属性值。第三个单元格需要为需要输入的值，可以直接指定值“玉”。





也可以设置为参数${unm}，在实际调用时再赋值，设置为参数后，需要设置该关键词的入参。



2、输入密码：Input Password

参数：[ locator | text ]

说明：和 Input Text 的区别是，不会在log 中显示输入的密码

3、按键：Press Key

参数：[ locator | key ]

说明：在指定位置元素模拟用户按键，“key”值可以是单个字符，也可以是以“\”开头的 ASCII code。

4、获取文本值：Get Text 参数：[ locator ]

返回值：目标元素的 text value

说明：可以用于和预期结果对比



## 点击

1、点击按钮：Click Button 参数:[ locator ]

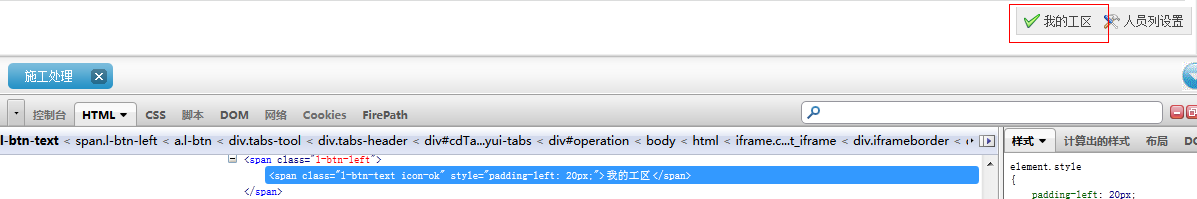
说明：通过 id，name，xpath 指定需要点击的按钮位置





2、点击元素：Click Element 参数：[ locator ]

说明：通过 id，name，xpath 定位



3、点击链接：Click Link 参数：[ locator ]

说明：通过 id，name，href，link text 定位



4、点击图片：Click Image 参数：[ locator ]

说明：通过 id，src，alt 定位



## 选择

1、Select All From List

参数：[ locator ]

说明：Selects all values from multi-select list identified by `id`.

2、Select From List

参数：[ locator | \*items ]

说明：Selects `\*items` from list identified by `locator`

If more than one value is given for a single-selection list, the last value will be selected. If the target list is a multi -selection list, and

`\*items` is an empty list, all values of the list will be selected.

\*items try to select by value then by label.

It's faster to use 'by index/value/label' functions.

Select list keywords work on both lists and combo boxes.

3 、 Unselect From List 参数：[ locator | \*items ]

说明：Unselects given values from select list identified by locator.

As a special case, giving empty list as `\*items` will remove all selections.

\*items try to unselect by value AND by label.

It's faster to use 'by index/value/label' functions.

Select list keywords work on both lists and combo boxes.

4、Select Checkbox

参数：[ locator ]

说明：Selects checkbox identified by `locator`. Does nothing if checkbox is already selected.

5、Unselect Checkbox 参数：[ locator ]

说明：Removes selection of checkbox identified by `locator`. Does nothing if the checkbox is not checked.

6、Select Radio Button

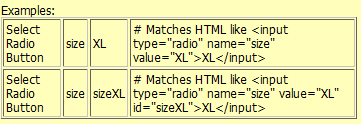
参数：[ group\_name | value ]

说 明 ： Sets selection of radio button group identified by

`group\_name` to `value`.

The radio button to be selected is located by two arguments:

`group\_name` is used as the name of the radio input

`value` is used for the value attribute or for the id attribute

## 校验

验证页面是否包含相应结果

* + 1. Page

1、Page Should Contain Button

（Page Should Not Contain Button）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies button identified by `locator` is found from current page.

This keyword searches for buttons created with either `input` or

`button` tag.

2、Page Should Contain Checkbox

（Page Should Not Contain Checkbox）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies checkbox identified by `locator` is found from current page.

3、Page Should Contain Element

（Page Should Not Contain Element）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies element identified by `locator` is found on the current page.

`message` can be used to override default error message.

4、Page Should Contain Image

（Page Should Not Contain Image）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies image identified by `locator` is found from current page.

5、Page Should Contain Link

（Page Should Not Contain Link）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies link identified by `locator` is found from current page.

6、Page Should Contain List

（Page Should Not Contain List）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies select list identified by `locator` is found from current page.

7、Page Should Contain Radio Button

（Page Should Not Contain Radio Button） 参数：[ locator | message= | loglevel=INFO ]

说明：Verifies radio button identified by `locator` is found from current page.

8、Page Should Contain Textfield

（Page Should Not Contain Textfield）

参数：[ locator | message= | loglevel=INFO ]

说明：Verifies text field identified by `locator` is found from current page.

* + 1. Location

1、Location Should Be 参数：[ url ]

说明：Verifies that current URL is exactly `url`.



2、Location Should Contain 参数：[ expected ]

说明：Verifies that current URL contains `expected`.



* + 1. Title

1、Title Should Be 参数：[ title ]

说明：Verifies that current page title equals `title`





* + 1. Text

1、Page Should Contain

（Page Should Not Contain） 参数：[ text | loglevel=INFO ]

说明：Verifies that current page contains `text`.

说 明 ：If this keyword fails, it automatically logs the page source using the log level specified with the optional `loglevel` argument. Giving

`NONE` as level disables logging.

2 、 Current Frame Contains 参数：[ text | loglevel=INFO ]

说明：Verifies that current frame contains `text`.

3、Textfield Value Should Be

参数：[ locator | expected | message= ]

说明：Verifies the value in text field identified by `locator` is exactly

`expected`.

`message` can be used to override default error message.

4、Textfield Should Contain

参数：[ locator | expected | message= ]

说明：Verifies text field identified by `locator` contains text `expected`.

`message` can be used to override default error message.

* + 1. Element

1、Element Should Be Disabled 参数：[ locator ]

说明：Verifies that element identified with `locator` is disabled.

2、Element Should Be Enabled 参数：[ locator ]

说明：Verifies that element identified with `locator` is enabled.

3、Element Should Be Visible 参数：[ locator | message= ]

说明：Verifies that the element identified by `locator` is visible.

Herein, visible means that the element is logically visible, not

optically visible in the current browser viewport. For example, an element that carries display:none is not logically visible, so using this keyword on that element would fail.

`message` can be used to override the default error message.

4、Element Should Contain

参数：[ locator | expected | message= ]

说明：Verifies element identified by `locator` contains text `expected`.

If you wish to assert an exact (not a substring) match on the text of the element, use `Element Text Should Be`.

`message` can be used to override the default error message.

5、Element Should Not Be Visible 参数：[ locator | message= ]

说明：Verifies that the element identified by `locator` is NOT visible.

This is the opposite of `Element Should Be Visible`.

`message` can be used to override the default error message.

6、Element Text Should Be

参数：[ locator | expected | message= ]

说明：Verifies element identified by `locator` exactly contains text

`expected`.

In contrast to `Element Should Contain`, this keyword does not try a substring match but an exact match on the element identified by

`locator`.

`message` can be used to override the default error message.

* + 1. List

1、List Selection Should Be 参数：[ locator | \*items ]

说明：Verifies the selection of select list identified by `locator` is exactly `\*items`.

If you want to test that no option is selected, simply give no `items`.

Select list keywords work on both lists and combo boxes.

2 、 List Should Have No Selections 参数：[ locator ]

说明：Verifies select list identified by `locator` has no selections. Select list keywords work on both lists and combo boxes.

* + 1. Checkbox

1 、 Checkbox Should Be Selected 参数：[ locator ]

说明：Verifies checkbox identified by `locator` is selected/checked.

2、Checkbox Should Not Be Selected 参数：[ locator ]

说 明 ： Verifies checkbox identified by `locator` is not

selected/checked.

* + 1. Radio Button

1 、 Radio Button Should Be Set To 参数：[ group\_name | value ]

说明：Verifies radio button group identified by `group\_name` has its selection set to `value`.

2、Radio Button Should Not Be Selected 参数：[ group\_name ]

说明：Verifies radio button group identified by `group\_name` has no

selection.

* + 1. Table

1、Table Cell Should Contain

参 数 ：[ table\_locator | row | column | expected | loglevel=INFO ] 说明：Verifies that a certain cell in a table contains `expected`.

Row and column number start from 1. This keyword passes if the

specified cell contains the given content. If you want to test that the cell

content matches exactly, or that it e.g. starts with some text, use `Get

Table Cell` keyword in combination with built-in keywords such as

`Should Be Equal` or `Should Start With`.

2、Table Column Should Contain

参数：[ table\_locator | col | expected | loglevel=INFO ]

说明：Verifies that a specific column contains `expected`.

The first leftmost column is column number 1. If the table contains cells that span multiple columns, those merged cells count as a single column. For example both tests below work, if in one row columns A and B are merged with colspan="2", and the logical third column contains "C".

Example:



3、Table Footer Should Contain

参数：[ table\_locator | expected | loglevel=INFO ]

说明：Verifies that the table footer contains `expected`.

With table footer can be described as any <td>-element that is child of a <tfoot>-element.

4、Table Header Should Contain

参数：[ table\_locator | expected | loglevel=INFO ]

说 明 ：Verifies that the table header, i.e. any <th>...</th> element, contains `expected`.

5、Table Row Should Contain

参数：[ table\_locator | row | expected | loglevel=INFO ]

说明：Verifies that a specific table row contains `expected`.

The uppermost row is row number 1. For tables that are structured with thead, tbody and tfoot, only the tbody section is searched. Please use `Table Header Should Contain` or `Table Footer Should Contain` for tests against the header or footer content.

If the table contains cells that span multiple rows, a match only occurs for the uppermost row of those merged cells.

6、Table Should Contain

参数：[ table\_locator | expected | loglevel=INFO ]

说明：Verifies that `expected` can be found somewhere in the table.

# 示例

## “打开首页”关键词

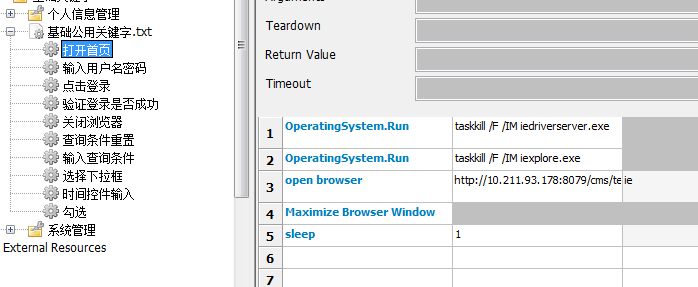
步骤：

（1）杀掉iedriverserver.exe进程；

（2）杀掉iexplore.exe进程；

（3）打开要访问的url，指定ie浏览器访问；

（4）最大化浏览器；

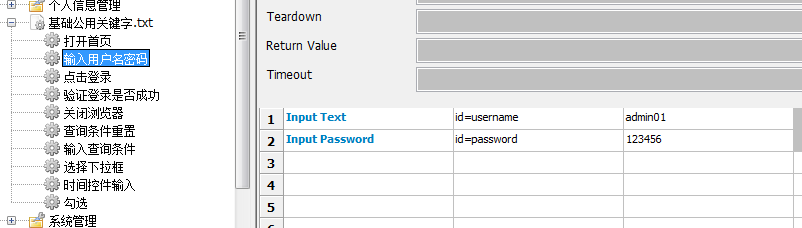
（5）等待1秒；

## “用户名密码”关键词

步骤：

（1）对所指定的输入框输入用户名；

（2）对所指定的输入框输入密码。

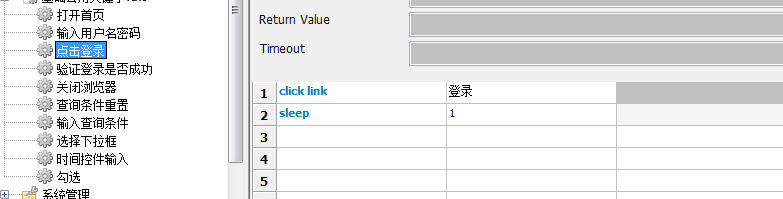


## “点击登录”关键词

步骤：

（1）使用“选择菜单”关键词，入参为集成定单、统计查询、工单查询；

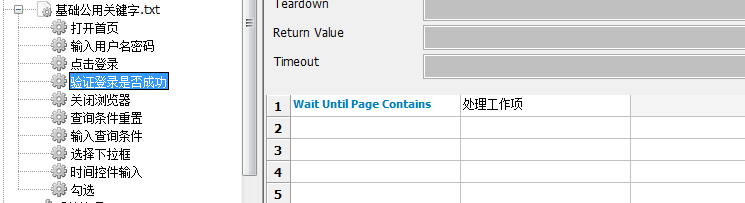
（2）等待 2S；



## “验证登录是否成功”关键词

步骤：

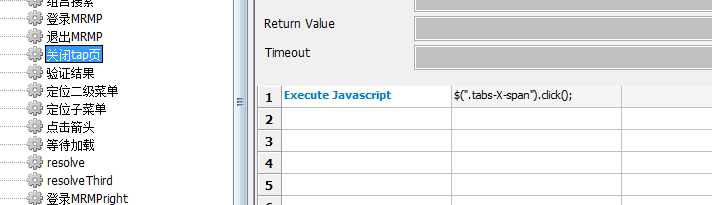
（1）等待页面出现特定文字或者字母，以代表是登录成功；



## “登录成功后续处理”关键词

步骤：

（1）登录成功后关闭默认tap页面，以防止对查找其他元素造成干扰；



## “关键字的二次封装”测试用例

步骤：

（1）新建另一个用户关键字：登录MRMP；

（2）按照步骤将1-5中所编写的关键字进行组合；

（3）这样就将打开首页，输入用户名密码，登录等一系列操作封装成一个关键字

