

目 录

- 1.引言
 - 1.1 编写目的
 - 1.2 pyswat优势有哪些
 - 1.3 pyswat框架环境搭建
- 2.pyswat使用步骤
 - 2.1 程序概览
 - 2.2 认识配置文件
 - 2.3 开始录制
 - 2.4 回放及调试
 - 2.5 执行结果查看
- 3.pyswat高阶使用
 - 3.1 “输入” 和 “参数”
 - 3.2 “at” 指定
 - 3.3 pyswat中写python
 - 3.3.1 check()方法
 - 3.3.2 uploadFile()方法
 - 3.3.3 scroll()方法
 - 3.3.4 find_by_text()方法
 - 3.3.5 show()方法
 - 3.3.6 time_start和time_end
 - 3.3.7 check_style()方法
 - 3.4 index属性
 - 3.5 innertext属性关联参数
- 4.写在后面



1.引言

1.1 编写目的

1.2 pyswat优势有哪些

1.3 pyswat框架环境搭建



1.1 编写目的

1.1 编写目的

目前web页面UI自动化测试框架种类繁多，多数需要使用人员具备一定的编程能力。而pyswat采用的执行方式是类似QTP或Selenium IDE的录制脚本的原理，然后进行回放，而不同的是pyswat录制的不是脚本命令，而是操作执行的步骤（案例），excel对于操作步骤清晰可见，同时便于修改维护。

另外看到那些录制后通过xpath的方式定位元素的方式感到恐怖。

而采用录制抛弃手写自动化程序的意义一方面是降低自动化测试成本（人力成本，时间成本），另一方面是以往项目的界面更新修改导致脚本定位元素失效是很头疼的事情，而录制是可以最大限度的降低此种影响，对于有变动的步骤重新单独录制一遍即可。



1.2 pyswat优势有哪些

1.2 pyswat优势有哪些

pyswat框架最主要的模块在于录制和回放上。虽然pyswat也是基于selenium实现的web自动化操作，但是pyswat框架完全对selenium进行了封装，在回放时不需要大家去操心如何去找页面元素。而录制时一方面借用了selenium，另一方面利用pythonHook记录下用户的鼠标，键盘操作以达到录制案例的目的。而对于诸如select，frame，框架中处理的是很好的。另外框架自己也封装了一些方法，如：drag方法（拖拽，区别于selenium的软拖拽，完全是根据传入两个元素的位置做了物理拖拽，效果更理想）check方法（页面内容检查，传入你想检查的内容，回放时会在日志中记录查找结果统计）scroll方法（滚动屏幕，支持底部，顶部滚动，滚动到元素位置，流畅滚动等）uploadFile方法（传入文件路径即可，支持多个文件同时上传，对于检查文件不存在等异常会自动关闭选择文件）

另外pyswat框架在异常处理上，做了浏览器弹出框的失败重新回放执行步骤，这样再也不怕测试时遇到弹出框导致用例执行失败了。这里我以雪球网站为例，看看录制登录的案例的模样，如下图

stepID	Action	思考时间	坐标	属性	步骤	输入	参数
step1	Mouse	1.97	1205,90	tagName: SPAN, id, class, innertext: 登录_docker:	点击-->登录		
step2	Mouse	1.71	726,379	tagName: INPUT, id, name, value, placeholder: 请输入手机号或者邮箱, class, href, innertext, docker:	请输入手机号或者邮箱	15000000000	
step3	Mouse	2.41	650,446	tagName: INPUT, id, name, value, placeholder: 请输入登录密码, class, href, innertext, docker:	请输入登录密码		
step4	Mouse	2.58	754,565	tagName: DIV, id, class, modal__login_btn, innertext, docker:	点击-->DIV		
stepOver	Over						

在“login”表里大家可以看到StepID（步骤ID），Action（动作来源），思考时间（动作执行等待时间），坐标（鼠标点击的屏幕坐标，这个在回放时可以当做一个轻骑兵来用），属性（记录元素的属性），步骤（记录操作，只做提示查看不影响回放），输入（需要输入文本时，后续完善优化脚本需要在这里调整），参数（配合输入中的内容，可以将这个输入定义为一个参数，后文可以调取这个参数名）

大家看到的这个excel表完全是自动生成的，只有当优化脚本时，你可能需要定义参数或是使用框架的封装功能时才需要手动调整。对于运行，详实记录日志及报错截图，当运行完后生成柱状图。（目前还未加入自动发邮件功能）

1.3 pyswat框架环境搭建

1.3 pyswat框架使用环境搭建

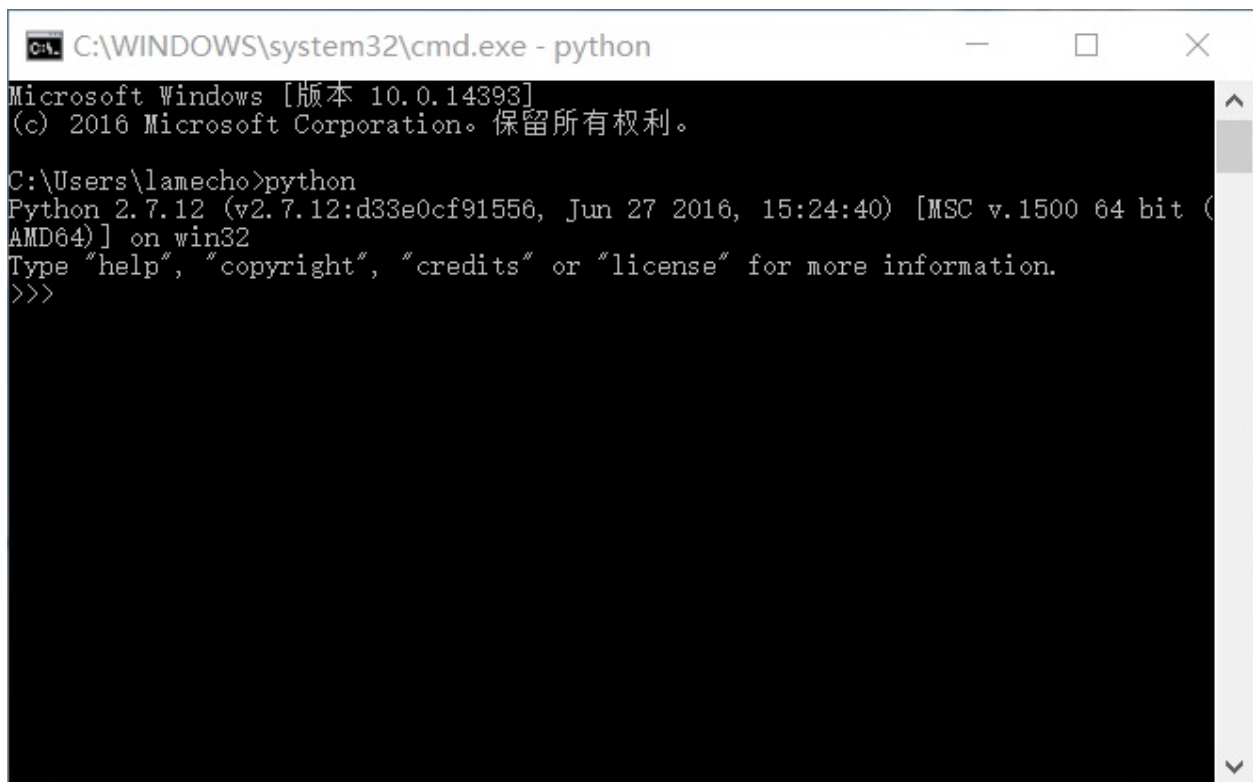
由于pyswat是使用python写的，目前只支持windows下python2.7的环境。

1.3.1 python安装

官网下载python27windows版本，最好是64位的。

<https://www.python.org/download/releases/2.7/>

安装成功后，运行--cmd--输入 python



```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

C:\Users\lamecho>python
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:24:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

看到如下信息表示安装成功。

1.3.2 selenium库安装

运行--cmd--输入 pip install selenium

注意不要使用3.4的版本（最新版貌似用起来有问题）

当selenium安装成功后，大家先手写一个脚本，通过selenium打开浏览器，如果成功再继续后面的步骤。有关脚本及selenium相关的内容大家可以看我的《python自动化测试应用-第5篇（WEB测试）--

Selenium环境篇》。强烈建议大家首先看看这篇文章的内容。

关于selenium的环境准备可以移步我的博客查看 [python自动化测试应用-第5篇 \(WEB测试\)](#) -- [Selenium环境篇](#)。也可以百度搜索“辣么丑 selenium”，微博搜索“lamecho辣么丑”。

1.3.3 其他的库安装

使用到的库，大家可以手动下载或者通过 运行控制台输入 pip install 第三方库名

1.xlrd, xlwt两个库。用于操作excel表格

直接

```
pip install xlrd
```

```
pip install xlwt
```

2.ConfigParser。用于读写“Enviroment.ini” 配置文件

```
pip install ConfigParser
```

3.numpy库 用于生成运行结果柱状图

numpy库，建议大家去<https://pan.baidu.com/s/1dE6u7kt> 下载安装（这是我提供的网盘链接）

4.matplotlib库。用于生成运行结果柱状图

matplotlib库，可以去官网下载<http://matplotlib.org/users/installing.html#windows> 或直接pip安装，输入

```
python -m pip install -U pip setuptools
```

```
python -m pip install matplotlib
```

5.pyHook库。用于python钩子，记录鼠标键盘操作。注意安装版本要对应python版本和32位还是64位的要求

```
官网链接https://sourceforge.net/projects/pyhook/files/pyhook/1.5.1/  
由于官链的版本是win32位的,如果操作系统是win64位的,去下面的链接下载  
http://www.lfd.uci.edu/~gohlke/pythonlibs/#pyhook
```

（当然我提供的程序压缩文件里，包含了64位的安装包。）

替换之前的32位pyHook文件，进入到python27安装目录下，我的在C:\Python27\Lib\site-packages\pyHook，删除之前安装的pyHook，将刚下载的64位压缩包中的pyHook复制到安装目录下就好了。

6.pywin32。python下操作win32api。同样注意下载的版本是32位还是64位。

大家去<https://pan.baidu.com/s/1dE6u7kt> 下载安装（这是我提供的网盘链接），里边是64位操作系统的版本

pywin32安装完后，运行脚本

```
# -*-encoding:utf-8 -*-  
import win32gui  
import win32con  
win32gui.MessageBox(0, u'请准备好测试环境，点击确认继续！', u'pyswat-录制', win32con.MB_OK)
```

如果弹出提示框表示安装成功。

如果还是有问题，参考<http://jingyan.baidu.com/article/72ee561aa7f129e16138dfa1.html?qq-pf-to=pcqq.group>

里的解决方法

7.bs4库。beautifulsoup4

pip install beautifulsoup4

全部都安装完成后，可以运行pip list，查看目前安装的版本信息。如下图


```
MarkupSafe (0.23)
matplotlib (2.0.0)
mistune (0.7.3)
mozdevice (0.50)
mozfile (1.2)
mozinfo (0.9)
mozlog (3.4)
moznetwork (0.27)
mozprocess (0.25)
mozprofile (0.28)
mozrunner (6.13)
mozversion (1.4)
nbconvert (4.2.0)
nbformat (4.1.0)
notebook (4.2.3)
numpy (1.11.3+mkl)
olefile (0.44)
pandas (0.19.2)
pathlib2 (2.1.0)
pickleshare (0.7.4)
Pillow (4.1.0)
pip (9.0.1)
prompt-toolkit (1.0.7)
psutil (5.2.1)
Pygments (2.1.3)
pyHook (1.5.1)
Pylsy (2.71)
pyparsing (2.2.0)
python-dateutil (2.6.0)
pytz (2017.2)
pywin32 (220)
pyzmq (15.4.0)
qtconsole (4.2.1)
requests (2.13.0)
scipy (0.19.0)
selenium (3.0.2)
sendkeys (0.3)
setuptools (20.10.1)
simplegeneric (0.8.1)
singledispatch (3.4.0.3)
six (1.10.0)
tornado (4.4.2)
traitlets (4.3.1)
tushare (0.7.4)
wcwidth (0.1.7)
widgetsnbextension (1.2.6)
win-unicode-console (0.5)
xlrd (1.0.0)
xlwt (1.2.0)
yapf (0.16.1)
```

2.pyswat使用步骤

- 2.1 程序概览
- 2.2 认识配置文件
- 2.3 开始录制
- 2.4 回放及调试
- 2.5 执行结果查看

2.1 程序概览

2.1.1 pyswat框架文件结构

modual	2017/5/3 12:49	文件夹	
pic	2017/5/3 11:49	文件夹	
2excel.py	2017/4/24 18:08	Python File	9 KB
Enviroment.ini	2017/5/3 13:23	配置设置	1 KB
res.png	2017/5/3 12:49	PNG 图像	569 KB
run.log	2017/5/3 12:49	文本文档	62 KB
temp_case.txt	2017/5/3 12:20	文本文档	2 KB
temp_case.xls	2017/5/3 12:37	Microsoft Excel ...	10 KB
webs_record.py	2017/5/2 16:36	Python File	9 KB
webs_replay.py	2017/5/3 12:51	Python File	6 KB
雪球网演示案例.xls	2017/5/3 10:16	Microsoft Excel ...	39 KB

首先获取到程序文件后，拷贝至任意目录。红框所示文件是必须文件。下面我们来一一认识一下：

1. “modual” 文件夹中包含程序运行依赖的模块。

2. “pic” 文件夹保存程序运行时报错截图
3. “2excel.py” 是生成excel案例的python程序。运行时确保 “temp_case.txt” 存在，当然 “temp_case.txt” 名称是可以在配置文件中任意指定的。“temp_case.txt” 可删除。
4. “res.png” 是运行结果的柱状图。可删除。
5. “run.log” 是运行时的日志文件，便于查看运行情况。可删除。
6. “temp_case.txt” 文件是运行 “webs_record.py” 生成的录制原始文档。需要用 “2excel.py” 转换为excel格式。可删除。
7. “temp_case.xls” 文件是转换后的案例文件，可删除。
8. “webs_record.py” 为录制程序。
9. “webs_replay.py” 为回放程序。
10. “雪球网演示案例.xls” 为回放执行案例。

2.2 认识配置文件

2.2.1 “ Enviroment.ini ”

“Enviroment.ini”是配置pyswat框架运行环境的作用，测试人员在使用pyswat时需要提前配置好运行的各项参数，如：录制网址，录制生成文件名称，录制浏览器选择，回放网址，回放案例，回放浏览器选择，参数记录等内容

2.2.2 “ Enviroment.ini ”详细说明

打开“Enviroment.ini”如上图所示，配置文件包含5个section (record, 2excel, replay, parameters, parameters_seq)。其中后两项parameters, parameters_seq是自动生成的参数值，当案例中配置有参数后，在案例回放后会自动写入更新。

1. [record]

控制“webs_record.py”录制程序的相关参数。

- 1.1“file = temp_case.txt”
录制完成后生成的原始记录的文件名称
- 1.2“url = https://xueqiu.com”
录制网址url链接
- 1.3“browser = Chrome”
录制使用浏览器，目前支持Chrome, Firefox, IE。需要selenium运行环境中存在对应各个浏览器的driver
- 1.4“ex-record = 1”
录制增强模式，默认打开。0是关闭（建议打开）

2. [2excel]

控制“2excel.py”程序转换案例程序的相关参数。

- 2.1“sheet = temp_case”
生成案例的sheet名称，可自定义
- 2.2“file = temp_case.xls”
生成案例的excel文档名称，可自定义
- 2.3“smart_mode = 1”
转化模式，目前默认为1，自动转换键盘输入内容。目前pyswat框架运行只识别该模式下的案例。

3. [replay]

控制“webs_replay.py”回放程序的相关参数。

- 3.1“file = C:\Users\lamecho\Desktop\pyswat\雪球网演示案例.xls”
运行案例的完整路径
- 3.2“case = login,send_message(20)”
运行案例的case名称及运行次数。如本参数是运行“雪球网演示案例”中 login,send_message两个case，其中send_message运行20次。
- 3.3“url = https://xueqiu.com”
回放时打开的web网址
- 3.4“browser = Chrome”
录制使用浏览器，目前支持Chrome, Firefox, IE。需要selenium运行环境中存在对应各个浏览器的driver
- 3.5“think_time = 0”
思考时间配置，0表示不使用录制的思考时间；1表示回放时执行思考时间。
- 3.6“wait_time = 3”

3.7 "retry = 0" 各个case切换时的等待时间，默认3秒
当案例回放失败后是否执行重试，0不打开重试功能，1执行重试。目前演示版本未放开此功能。

4. [parameters]

保存案例中指定random方式生成的参数名称及其内容。

4.1 "p1 = message-0020"

当存在多个参数时，会依次保存。可修改参数内容，运行时确保案例中调用的参数存在于配置文件中。

5. [parameters_seq]

保存案例中指定的sequence方式生成的参数及其内容。

5.1 "p1 = message-0011, message-0020, message-0017"

当存在多个参数时，会依次保存。可修改参数内容，运行时确保案例中调用的参数存在于配置文件中。注意分隔符号采用英文逗号。

2.3 开始录制

2.3.1 录制第一个案例

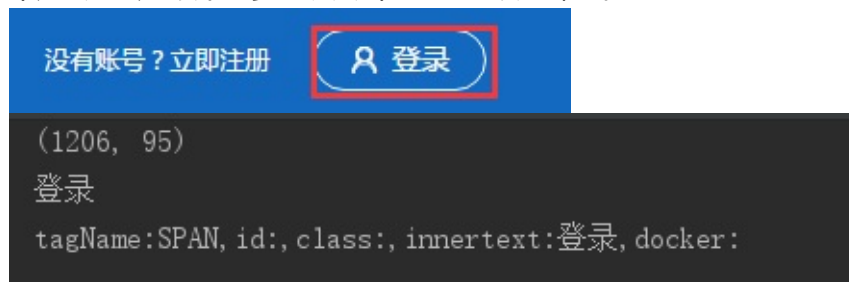
- 使用webs_record.py开始录制。

当我们准备好配置文件“Enviroment.ini”后，双击“webs_record.py”运行录制程序，等待浏览器打开被录制网址。本教程以雪球网站登录为操作case。如下图，正常打开雪球后浏览器窗口最大化，并弹出提示框提示是否准备好测试环境，点击“确定”开始录制。

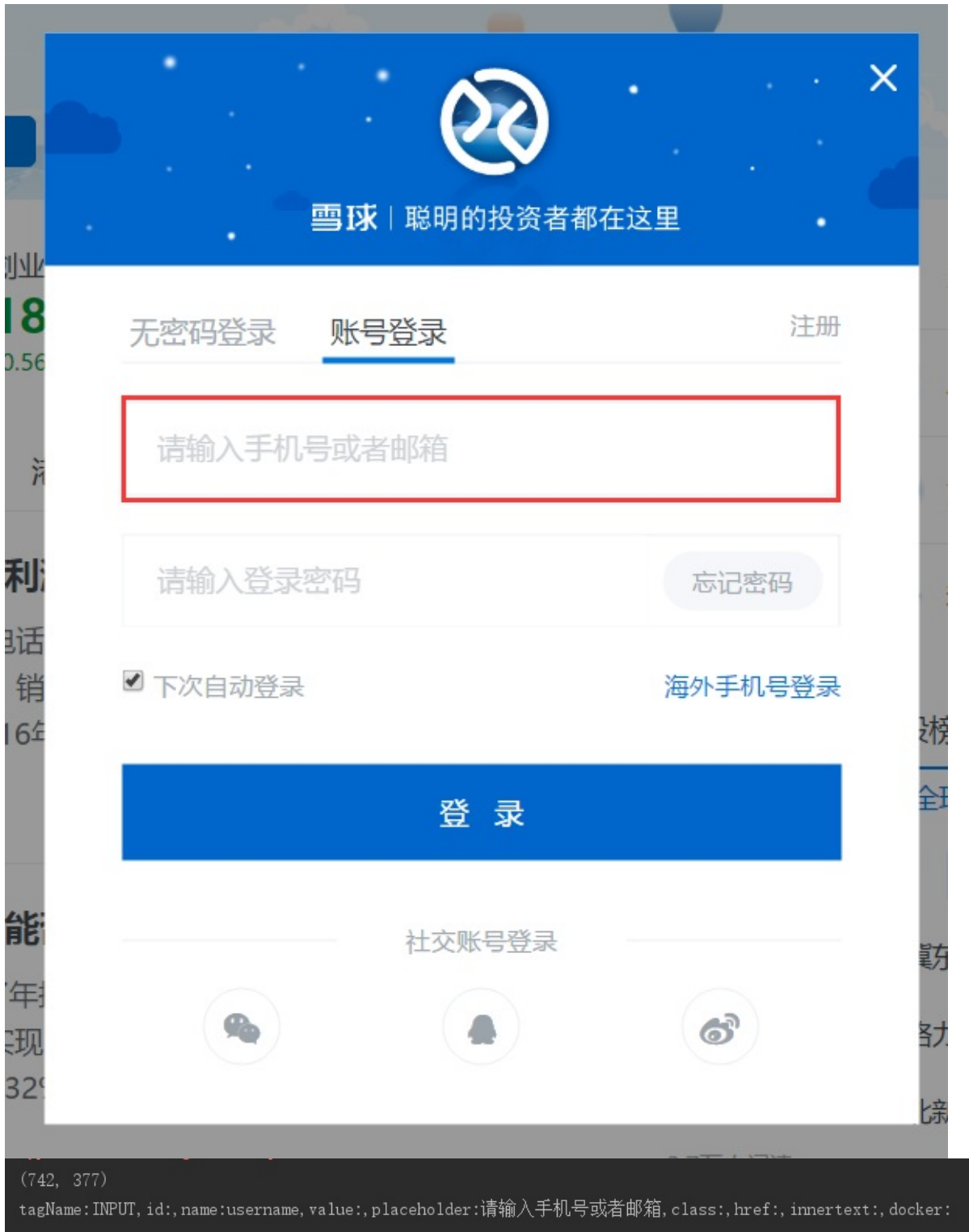


2.3.2 录制操作

pyswat的录制采用右击定位网页元素的方式，先获取定位元素的各项属性，及点击坐标位置等内容。右键选中的网页元素会被画一个红框表示定位元素正常，并且记录到当前元素的各项属性内容，这里雪球网站的“登录”按钮通过右键是无法正常获取的（红框会画在整个网站的body体上，此时代表正常获取失效），当录制程序遇到这种元素时会开启增强模式录制元素内容，我们在录制操作上不用去理会直接下一步左键点击“登录”按钮即可。



可以看到程序记录下了点击的坐标级各项属性值（注意这里增强模式录制的属性内容会比正常模式的属性内容少）。接下来当左键点击“登录”后会弹出登陆框，我们先右键继续获取输入用户名的输入框元素，这时可以看到输入框会闪一下红框，代表获取元素成功并记录下相关属性信息。



接下来输入用户名，继续获取密码，登录等元素后操作登录按钮完成本次登录的case录制。

总体来说pyswat的录制采用先右键获取元素的方式到达录制案例的目的。当我们录制结束后，点击键盘的“end”键结束录制。此时结束录制会生成一个临时的txt文档“temp_case.txt”。大家可以打开查看具体的录制结果。注意当结束录制后，“temp_case.txt”在行末加入‘over’标识。


```
temp_case.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Mouse|1493785174.33|1138,379|tagName:INPUT,id:login_username,name:username,value:,placeholder:请输入手机号,class:form-control input-lg,href:,innertext:,docker:
Mouse|1493785176.02|1093,436|tagName:INPUT,id:login_password,name:password,value:,placeholder:密码,class:form-control input-lg,href:,innertext:,docker:
Mouse|1493785177.28|1065,504|tagName:BUTTON,id:btn-login,name:,value:,placeholder:,class:btn btn-primary btn-lg btn-block btn-login,href:,innertext:立即登录,docker:
Mouse|1493785179.94|1289,552|tagName:A,id:forgetpass,name:,value:,placeholder:,class:gray1,href:http://shop.md168.cn/index/login/findstep1,innertext:忘记密码?,docker:
Mouse|1493785182.98|662,342|tagName:INPUT,id:mobile,name:mobile,value:,placeholder:请输入手机号码,class:form-control input-lg,href:,innertext:,docker:
Mouse|1493785184.29|630,405|tagName:INPUT,id:yzm_code,name:verify,value:,placeholder:验证码,class:form-control i-ymz input-lg,href:,innertext:,docker:
Mouse|1493785196.41|660,492|tagName:BUTTON,id:,class:btn btn-link btn-qiehuan,innertext:换一张,docker:
over
```

当我们录制完一个case后，接下来就是将“temp_case.txt”转化为excel案例case。

2.3.3 转换案例 2excel

- 使用2excel.py转换“temp_case.txt”，生成“temp_case.xls”案例文档。这里需要注意，转换前检查“temp_case.txt”里只存在一个‘over’标识。尽量在录制新的案例时，先删除原有的“temp_case.txt”文件。转换成功后会覆盖或生成新的“temp_case.xls”，此时可以将生成的案例拷贝之自己的项目案例总表中（项目案例总表需要自己创建，维护）

2.3.4 总结

pyswat录制采用先右键获取元素的方式，所以一定要在操作上使用好右键，才能达到准确识别点击元素。结束录制点击键盘的‘end’按键。

2.4 回放及调试

2.4.1 回放案例

当我们成功录制好案例后，在配置文件中定义好回放的相关配置如：file（回放案例完整路径）case（案例中想要回放的case，多个case以英文分割）url（回放网址）等。在准备好回放配置后，双击运行“webs_replay.py”我们的回放程序，就会开始回放了。如下图程序打印日志。

```
开始执行 [ C:\Users\lamecho\Desktop\pyswat_xueqiu_show\雪球网演示案例.xls ]
-- Case: [login] 开始运行
---- step1 开始
---- step1 结束
---- step2 开始
---- step2 结束
---- ; 开始
---- ; 结束
---- step4 开始
---- step4 结束
-- Case: [login] 运行结束 (Pass) 共执行案例数: 5 失败案例数: 0
-- Case: [send_message] 开始运行
---- step5 开始
----- send_message step5 Fail Method Cannot Find Element
---- step5 结束
---- step6 开始
----- send_message step6 Fail Method Cannot Find Element
---- step6 结束
---- step7 开始
----- 查找结果共: 0条
----- 查找结果页面无此元素 随便说说
---- step7 结束
---- step8 开始
----- send_message step8 Fail Method Cannot Find Element
---- step8 结束
---- step9 开始
----- send_message step9 Fail Method Cannot Find Element
---- step9 结束
---- step10 开始
----- 查找结果共: 0条
----- 查找结果页面无此元素 随便说说
---- step10 结束
-- Case: [send_message] 运行结束 (Pass) 共执行案例数: 7 失败案例数: 4
-- Case: [pay message] 开始运行
```

当回放完成后，在根目录下也会生成“run.log”日志文件，方便对结果的检查。

```

2017-05-05 17:19:21 :开始执行 { C:\Users\lamecho\Desktop\pyswat_xueqiu_show\雪球网演示案例.xls }
2017-05-05 17:19:21 :-- Case: [login] 开始运行
2017-05-05 17:19:21 :---- step1 开始
2017-05-05 17:19:21 :---- step1 结束
2017-05-05 17:19:21 :---- step2 开始
2017-05-05 17:19:22 :---- step2 结束
2017-05-05 17:19:22 :---- ; 开始
2017-05-05 17:19:23 :---- ; 结束
2017-05-05 17:19:23 :---- step4 开始
2017-05-05 17:19:23 :---- step4 结束
2017-05-05 17:19:23 :-- Case: [login] 运行结束 (Pass) 共执行案例数: 5 失败案例数: 0
2017-05-05 17:19:26 :-- Case: [send_message] 开始运行
2017-05-05 17:19:26 :---- step5 开始
2017-05-05 17:19:28 :----- send_message step5 Fail Method Cannot Find Element
2017-05-05 17:19:28 :---- step5 结束
2017-05-05 17:19:28 :---- step6 开始
2017-05-05 17:19:30 :----- send_message step6 Fail Method Cannot Find Element
2017-05-05 17:19:30 :---- step6 结束
2017-05-05 17:19:30 :---- step7 开始
2017-05-05 17:19:32 :----- 查找结果共: 0条
2017-05-05 17:19:32 :----- 查找结果页面无此元素 随便说说
2017-05-05 17:19:33 :---- step7 结束
2017-05-05 17:19:33 :---- step8 开始
2017-05-05 17:19:36 :----- send_message step8 Fail Method Cannot Find Element
2017-05-05 17:19:36 :---- step8 结束
2017-05-05 17:19:36 :---- step9 开始
2017-05-05 17:19:37 :----- send_message step9 Fail Method Cannot Find Element
2017-05-05 17:19:37 :---- step9 结束
2017-05-05 17:19:37 :---- step10 开始
2017-05-05 17:19:40 :----- 查找结果共: 0条
2017-05-05 17:19:40 :----- 查找结果页面无此元素 随便说说
2017-05-05 17:19:41 :---- step10 结束
2017-05-05 17:19:41 :-- Case: [send_message] 运行结束 (Pass) 共执行案例数: 7 失败案例数: 4
2017-05-05 17:19:44 :-- Case: [pay_message] 开始运行
2017-05-05 17:19:44 :---- step1 开始

```

日志文件清楚的记录了程序运行时间，运行案例，case，步骤，执行情况，结果等信息。

当回放时，发现有步骤报错，根据报错信息查看对应步骤，检查报错原因，然后修改，优化我们的案例。

因为pyswat框架在定位查找元素同样是依据selenium的提供的API进行查找，只是在框架里对查找进行了智能算法优化，并提供给使用者多种查找组合。

举个简单的例子，看下面的案例截图

#	A	B	C	D	E	F	G	H
1	stepID	Action	思考时间	坐标	属性	步骤	输入	参数
2	step1	Mouse	3.28	560,92	tagName:A.id,;name,;value,;placeholder,;class:dropdown-toggle,href:https://xueqiu.com/hq,innertext:行情,docker:	点击---->行情		
3	step2	Mouse	5.33	569,267	tagName:A.id,;name,;value,;placeholder,;class:href:https://xueqiu.com/hq#xoss,innertext:新股上市,docker:	点击---->新股上市		
4	step3	Mouse	10.87	1204,248	tagName:A.id,;name,;value,;placeholder,;class:followStock,href:https://xueqiu.com/hq#,innertext:关注,docker:	点击---->关注	at:今创集团	
5	step4	Mouse	1	791,630	tagName:INPUT,id,;name,;value,;placeholder,;class:cancel button,href,;innertext,docker:	点击---->INPUT		
6	stepOver	Over						
7	step4	Mouse	1	791,630	tagName:INPUT,id,;name,;value,;placeholder,;class:cancel button,href,;innertext,docker:	点击---->INPUT		
8								
9								
10								

在step3中，我们录制的“属性”信息为：

“tagName:A,id,;name,;value,;placeholder,;class:followStock,href:https://xueqiu.com/hq#,innertext:关注,docker:”

可以看到该元素的属性只有tag, class, innertext几个属性值。这时在selenium里只能通过find_elements_by_class_name('followStock')去查找元素，并且会找出一个元素集合，这时回放时，程序是无法判断你是要点击具体哪个元素的。这时有几种方法去处理这样的情况，比如说

1.在pyswat框架里是可以写python脚本的，我们可以将案例中属性内容清除，通过手写脚本实现。像这样browser.find_elements_by_class_name('followStock')[0].click()。

2.由于框架在录制时是记录了点击位置坐标的，这时我们就可以把案例中属性内容清除，换成“失去焦

点”，不需要引号。此时程序在运行到这个step时，会自动切换回放模式到物理点击上，放弃selenium的获取方式，通过鼠标的模拟点击达到点击操作的目的。

3.最后这种方式是pyswat框架比较建议的，在“输入”中通过at指定的方式关联查找内容（见截图）。具体用法会在后面的高阶部分涉及。

2.4.2 案例调试及优化

我们先来看下一个case，雪球网站登录后发布消息的案例。截图见下方

stepID	Action	思考时间	坐标	属性	步骤	输入	参数
step5	Mouse	2.08	541,462	tagName:DIV,id:,name:,value:,placeholder:,class: edui-body-container,href:,innerText:docker:			
step6	Mouse	1.23	949,564	tagName:BUTTON,id:,name:,value:,placeholder:,class: btn btn-medium status-submit,href:,innerText:发布,doctype: sleep(1)	点击---->发布	random:随便说说+int(3)	p1
step7	Key_Fun		pys	check(browser,innerText:随便说说)			
step8	Mouse	1.23	795,717	tagName:A,id:,name:,value:,placeholder:,class: btn-delete,href:https://xueqiu.com/8871640539#,innerText:删除,doctype: sleep(1)	点击---->删除		
step9	Mouse	1	724,390	tagName:BUTTON,id:,name:,value:,placeholder:,class: btn btn-medium sure,href:,innerText:确定,doctype: sleep(1)	点击---->确定		
step10	Key_Fun		pys	check(browser,innerText:随便说说)			
stepOver	Over						

在引言中带大家认识了案例的每列的含义，这里就不再赘述了。由于我们的案例是自动生成的，原则上除了“参数”这列是需要后期大家自己定义，其他的列都是程序根据实际录制结果而自动生成。

优化脚本建议思路：

- 1. “步骤” 由于在回放中不影响程序执行，只是为了让大家看清楚每一个step的操作含义，这里建议大家再录制的案例中对于表述不清楚的步骤项手动添加，比如“点击---->DIV”这个步骤确实没有什么实际意义，还是希望大家替换成“点击---->关闭”这类更有参考价值的文字内容；
- 2. “输入” 实际录制的是录制时大家输入的文本内容，比如在step5中我们在发布一条消息输入“test”，如果系统对于输入有做唯一性的判断或要求，这时我们的案例回放是会失败的（难道要每次回放都要修改内容？），其实只要对输入做参数化定制，修改为“random:随便说说+int(3)”。在实际回放时会自动生成文本“随便说说456”（数字是3位的随机数），这样我们的案例就会更丰满（原谅我想到了这个词）一些。具体的用法会放在后面的高阶教程中，这里只是提供给大家优化案例的思路。
- 3. “参数” 在案例中也是起到很重要的，在本case中，对于输入内容指定了参数p1，这样后面的案例某个步骤需要用到这个参数时，比如还有个输入步骤我想输入同样的内容，这时我们可以把“输入”项中原有的内容删除，只需要在“参数”项中写入p1就可以了。
- 4. “属性” 项原则上是不需要大家修改的，但对于调试失败的步骤，我们需要查看案例中对应步骤的属性是不是和实际页面有差别。这里也举个例子，在录制百度页面时，对“设置”的录制，原本在页面设置的“class : pf”，而鼠标移到“设置”上点击后变为“class=pf pfhover” class会多出一个，所

以针对这种情况，如果碰到失败的步骤，建议还是手动删掉这个多出的名称。

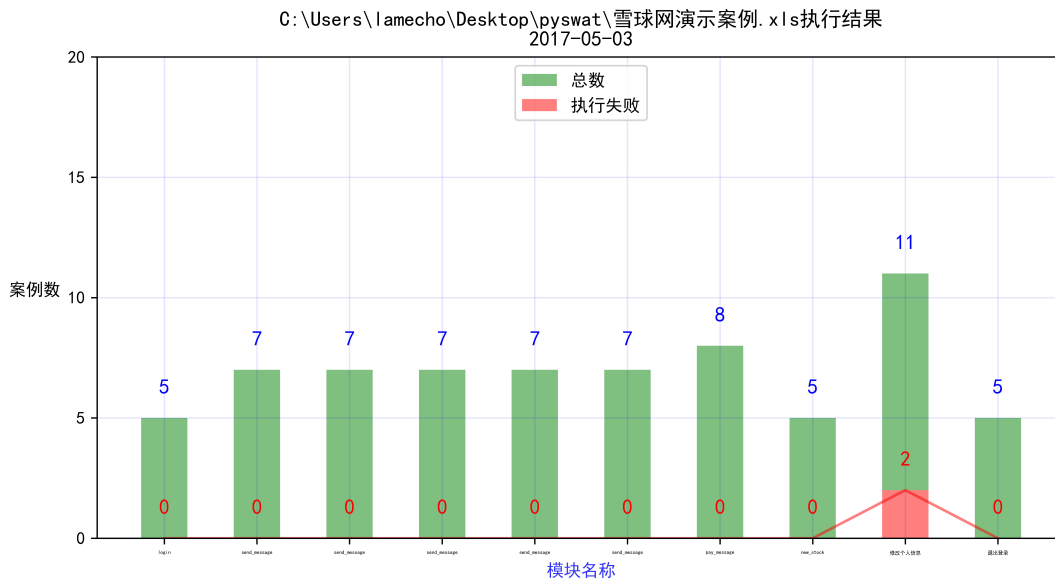
2.5 执行结果查看

2.5.1 检查日志

日志文件“run.log”会记录每次的执行情况，需要时可以根据时间快速定位问题

2.5.2 结果柱状图

将回放结果生成“res.png”，生成的柱状图直观的反应本次的执行结果。



2.5.3 发送邮件

此功能暂时未开发，因为最近一直在优化框架，而本功能的开发难度也比较低，暂时先拖拖。

3.pyswat高阶使用

3.1 “输入” 和 “参数”

3.2 “at” 指定

3.3 pyswat中写python

3.4 index属性

3.5 innertext属性关联参数

3.1 “输入” 和 “参数”

3.1.1 “输入” 的使用进阶

“输入” 和 “参数” 算是一对好基友。在案例中这两个地方的内容一般是需要搭配使用的，在前面也多次提到过他们的使用。

比如说这样的测试场景，在每一次回放时需要针对某个步骤的输入是不同的值，但我们的录制是固定的一个输入内容的，这个时候我们就需要把原来的固定“输入”内容替换。这时就要派上我们的random出场了，将输入内容替换成random试试吧！

花式输入之 random

首先我先列举出一个比较全的用法---random：测试+int+str(6)

代表每时输入的内容是：“测试xL06am” 其中“测试”是前缀，“xL06am”是随机字母加数字的6位数。

同样的最简单的用法是直接输入random，随机生成数字+字母。也可以这样random：int+str(10)不要前缀，或者这样random：test+int(4)。具体组合自己开发，哈哈。

花式输入之 sequence

sequence与random不同在于指定按照顺序生成输入内容。比如我们在“输入”项里填上sequence：test，在执行案例的时候会按照顺序生成“test0001”，“test0002”以此类推。

这里要强调一点，指定输入为sequence，一定要在“参数”项里定义参数名称。而且sequence是需要搭配我们的case运行次数才有实际效果的，比如在配置文件中指定某个case运行次数为10次（运行次数在配置文件中指定），而这个case中有个step的输入是sequence方式，那么在这10次的执行过程中，我们这个step的输入会按顺序依次生成。第一遍会输入“test0001”，第二遍执行输入“test0002”这样。当然你还可以这样写sequence：测试(10),这又是什么意思呢？表示我们生成的起始位数是从第11位开始，第一次执行生成“测试0011”。

3.1.2 “参数” 的使用进阶

参数的使用一方面是保存我们的输入，另一方面是关联我们的输入内容。说白了就是将我们的输入内容保存起来。我们来想想这样的测试场景，在某个case中某个step输入了内容，比如说这个输入内容是我们新建的一个网页元素，而另一个case在执行时需要用到这个内容作为输入内容，这时如果在前面的步骤我们指定了参数，那么这个时候在后面的案例中不需要在“输入”项中填写内容，只需要把前面指定的参数名称填到当前步骤的参数项中即可。

在2.2配置文件的讲解中，我们提到了配置文件中保存参数的地方有两块，一个是[parameters]还有一个是[parameters_seq]。其中注意[parameters_seq]是保存的是sequence方式生成的内容。而且使用sequence的参数要这样写：seq：p1（加入参数名位p1）。但是如果写成p1程序也不会报错，此时会关联到sequence最后一次生成的内容上，也就是配置文件中[parameters]里参数p1的内容。

具体在使用上，某个case使用了seq：p1作为参数化输入，就表示此步骤的输入是按照参数p1的集合内容

取值，而且是按照顺序从第一位取。当然也可以指定取固定位置像这样seq : p1[0]取第一位的值。

seq参数使用场景举例：拿登录输入用户名，密码来说，提前准备好[parameters_seq]里两个参数uname和psw（[parameters_seq]里的参数内容允许修改，也就是说你可以自己添加到配置里去），uname : test1 , test2 , test3 ; psw : 123456 , 123456 , 888888。这样就有两组参数内容，在登录的case中不需要输入登录名和密码，只需要指定参数名即可，case执行三遍按照顺序自动取值。如果执行10遍会怎么样呢，程序会从头开始取。

注意：输入如果指定位random或是sequence，必须要指定参数名。因为参数的使用按照使用思路是本次执行中使用前面案例的内容，当然random不会做强制校验，可以在随机生成内容时不去指定参数名，只会在日志里打印警告信息。同时后面的步骤中指定了一个没有的参数时，程序会自动查找配置文件的原始记录

3.2 “at” 指定

3.2.1 “at” 关联方式

什么是“at”关联？

我们先来看看下方的案例截图

	A	B	C	D	E	F	G	H
1	stepID	Action	思考时间	坐标	属性	步骤	输入	参数
2	step1	Mouse	3.28	560.92	tagName:A,id:,name:,value:,placeholder:,class:dropdown-toggle,href:https://xueqiu.com/hq,innerText:行情,docker:	点击--->行情		
3	step2	Mouse	5.33	569.267	tagName:A,id:,name:,value:,placeholder:,class:,href:https://xueqiu.com/hq#xass,innerText:新股上市,docker:	点击--->新股上市		
4	step3	Mouse	10.87	1204.248	tagName:A,id:,name:,value:,placeholder:,class:followStock,href:https://xueqiu.com/hq#,innerText:关注,docker:	点击--->关注	at:今创集团	
5	step4	Mouse	1	791.630	tagName:INPUT,id:,name:,value:取消,placeholder:,class:cancel button,href:,innerText:,docker:	点击--->INPUT		
6	stepOver	Over						
7	step4	Mouse	1	791.630	tagName:INPUT,id:,name:,value:取消,placeholder:,class:cancel button,href:,innerText:,docker:	点击--->INPUT		
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

step3的步骤是“点击---关注”，表示在雪球网站上关注某一只新股，而在网站页面上新股的关注不止一只股票，在一个table列表中有10只。像下图这样

The screenshot shows the Xueqiu website interface. At the top, there is a navigation bar with the Xueqiu logo and search bar. Below it, there is a section for '行情中心 > 沪深股市 > 新股上市'. A table titled '新股申购' (New Stock Subscription) is displayed, listing various stocks with columns for '股票简称', '申购代码', '申购日期', '发行量(万)', '网上发行量(万)', '申购上限(万)', '发行价', '中签号公布日', '中签率(%)', '上市日期', and '操作'. The '操作' column contains '关注' (Follow) buttons for each stock. The '关注' button for the first stock, '中孚信息', is circled in red.

股票简称	申购代码	申购日期	发行量(万)	网上发行量(万)	申购上限(万)	发行价	中签号公布日	中签率(%)	上市日期	操作
中孚信息	300659	2017-05-17	2040.00	816.00	0.80	-	2017-05-19	-	-	关注
诚邦股份	732316	2017-05-17	0.00	2032.80	2.00	-	2017-05-19	-	-	关注
日播时尚	732196	2017-05-17	0.00	2400.00	2.40	-	2017-05-19	-	-	关注
弘信电子	300657	2017-05-12	2600.00	1000.00	1.00	-	2017-05-16	-	-	关注
华荣股份	732855	2017-05-12	0.00	3300.00	3.30	-	2017-05-16	-	-	关注
鼎瑞股份	300655	2017-05-11	0.00	806.25	0.80	-	2017-05-15	-	-	关注
中马传动	732767	2017-05-11	0.00	2133.00	2.10	-	2017-05-15	-	-	关注
天圣制药	002872	2017-05-10	5300.00	2120.00	2.10	-	2017-05-12	-	-	关注
今创集团	732680	2017-05-10	4200.00	1260.00	1.20	-	2017-05-12	-	-	关注
顶点软件	732383	2017-05-10	2105.00	842.00	0.80	-	2017-05-12	-	-	关注

看到上图的截图，如果我想点击第一个或是第四个股票的“关注”按钮怎么办呢？因为在“关注”的属性中大家是可以看到只有class属性可以用，而如果通过class去找会找出10个“关注”元素。用过selenium API的同学可能知道此时只能靠指定集合的位数去进行点击，然而这样的方式也是有弊端的，因为我们不能和具体的业务关联。

而抛开测试的思路，站在用户使用的角度去看问题，用户不就是想点击一只股票的关注吗？而这个关注是和一只股票名字关联的。所以我们的at就是用来指定与关注关联的文本内容，像截图中那样写“at:今创集

团”即点击今创集团这只股票的关联。

当然除了指定固定的关联内容，你还可以这样写“at:{p1}”关联参数内容。

3.3 pyswat中写python

3.3.1 pyswat中写脚本

其实这个功能的实现目的，纯粹为了好玩，最后慢慢加入到框架中，比如你在使用loadrunner去做性能测试，脚本你可以去录制，当然你认为你很牛也可以直接手写脚本。

那么来看看手写脚本有哪些好处呢？我认为一个原因还是最大化的符合自己项目测试的要求，当然还有我们可以做各种私有化定制，比如说我对一些web操作做了封装，前文也提到了。下面我们来看看如何在案例中添加脚本命令的。

stepID	Action	思考时间	坐标	属性	步骤	输入	参数
1	Mouse	2.08	714,562	tagName: DIV, id, class: pay-at, innertext: 提问, docker:	点击---->提问		
2	Mouse	2.08	504,494	tagName: INPUT, id, name, value, placeholder: 输入用户昵称, class, href, innertext, docker:	输入用户昵称	雪球	
3	Mouse	1	518,518	tagName: DIV, id, class: pay-select-item, innertext: 雪球, docker:	点击---->雪球		
4	Key_Fun	2.08	pys	browser.find_elements_by_class_name('pay-confirm-btn')[0].click()	点击---->确定		
5	Mouse	2.08	924,365	tagName: BUTTON, id, name, value, placeholder, class: btn btn-medium pay-submit, href, innertext: 支付并发布, docker:	点击---->支付并发布		
6	Mouse	2.08	812,511	tagName: DIV, id, class: reward-modal-btn pay-btn, innertext: 立即支付, docker:	点击---->立即支付		
7	Key_Fun	2.08	pys	browser.implicitly_wait(10) browser.find_elements_by_class_name('close')[1].click()	点击---->DIV		
8	Mouse	2.08	924,565	tagName: BUTTON, id, name, value, placeholder, class: btn btn-medium pay-submit, href, innertext: 支付并发布, docker:	点击---->支付并发布		
9	Mouse	2.08	812,511	tagName: DIV, id, class: reward-modal-btn pay-btn, innertext: 立即支付, docker:	点击---->立即支付		
10	Mouse	2.08	932,321	tagName: DIV, id, class: close, innertext, docker:	点击---->DIV		

如截图红框所示，手动修改案例或添加一条案例，Action指定为“Key_Fun”，坐标指定为“pys”，属性写脚本，这里要注意如果思考时间的指定。这里我们的python脚本写了

“browser.find_elements_by_class_name('pay-confirm-btn')[0].click()”。你也可以写多行比如step7那样。这里要注意多行的情况如果有缩进，需要严格执行python的缩进要求。

3.3.2 封装了哪些方法？

能够写python语句了，那么我们就开始封装一些常见的操作吧！如下图

stepID	Action	思考时间	坐标	属性	步骤
step1	Key_Fun	2.08	pys	check(browser,'innertext:辣么丑')	
step2	Mouse	3.18	1205,85	tagName: A, id, nav-account-name, name, value, placeholder, class: dropdown-toggle, href: https://xueqiu.com/8871640539#, innertext: 辣么丑	点击---->辣么丑
step3	Mouse	1	1172,251	tagName: A, id, name, value, placeholder, class: exit, href: https://xueqiu.com/user/logout?redirect_uri=/, innertext: 退出, docker:	点击---->退出
step4	Key_Fun	2.08	pys	check(browser,'innertext:辣么丑')	
stepOver	Over				

在退出登录的case中，我们在属性里加入了check()方法，这样写的check(browser,'innertext:辣么丑')，当然在退出前和退出后各加入了一个check。这样在日志中会记录我们的check结果。如下图

本文档使用 [看云](#) 构建

```

2017-05-03 11:06:59 :-- Case: [退出登录] 开始运行
2017-05-03 11:06:59 :---- step1 开始
2017-05-03 11:07:00 :----- 查找结果: 辣么丑
2017-05-03 11:07:00 :----- 查找结果共: 1条
2017-05-03 11:07:02 :---- step1 结束
2017-05-03 11:07:02 :---- step2 开始
2017-05-03 11:07:05 :---- step2 结束
2017-05-03 11:07:05 :---- step3 开始
2017-05-03 11:07:06 :---- step3 结束
2017-05-03 11:07:06 :---- step4 开始
2017-05-03 11:07:07 :----- 查找结果共: 0条
2017-05-03 11:07:07 :----- 查找结果页面无此元素 辣么丑
2017-05-03 11:07:09 :---- step4 结束
2017-05-03 11:07:09 :-- Case: [退出登录] 运行结束 (Pass) 共执行案例数: 5 失败案例数: 0
2017-05-03 11:07:12 :执行 { C:\Users\lamecho\Desktop\pyswat\雪球网演示案例.xls } 结束 共执行案例数: 41 失败案例数: 2
2017-05-03 11:07:12 :-----

```

- 1.check()方法

检查页面信息，用于做执行结果的校验。目前支持的校验类型有

1. innertext (元素文本校验)。写法check(browser,'innertext:具体text')
2. tagname (标签校验)。写法check(browser,'tagName:A')，检查所有如A标签的内容
3. class (class属性校验)。写法check(browser,'class:名称')，检查所有class为指定名称的内容
4. id (id属性校验)。写法check(browser,'id:id名称')，检查所有id为指定名称的内容

- 2.uploadFile()方法

上传文件，支持多文件。写法uploadFile("D:\1.jpg")，多个文件用逗号隔开。

- 3.scroll()方法

1. 屏幕滚动方法，这里有几种使用方式：
2. 屏幕滚动 到底scroll(browser,'bottom')
3. 屏幕滚动 到顶scroll(browser,'Top')
4. 屏幕滚动 到底 平顺的滚动，会很慢 scroll(browser,'smooth')
5. 屏幕滚动 到数字位置scroll(browser,500) 移到500高度
6. 屏幕滚动 到元素位置scroll(browser,el) 移到el元素的位置（注意el元素你需要手写脚本获取）
像这样

```

el=browser.find_elements_by_class_name('title')[3]
scroll(browser,el)

```

3.3.1 check()方法

*3.3.1check()方法

检查页面信息，用于做执行结果的校验。目前支持的校验类型有

1. innertext (元素文本校验)。写法check(browser,'innertext:具体text')
2. tagname (标签校验)。写法check(browser,'tagName:A')，检查所有如A标签的内容
3. class (class属性校验)。写法check(browser,'class:名称')，检查所有class为指定名称的内容
4. id (id属性校验)。写法check(browser,'id:名称')，检查所有id为指定名称的内容

3.3.2 uploadFile()方法

3.3.2uploadFile()方法

上传文件，支持多文件。写法uploadFile("D:\1.jpg")，多个文件用逗号隔开。

uploadFile("D:\1.jpg","D:\2.jpg"),必须你的上传支持多文件同时上传。

3.3.3 scroll()方法

3.3.3scroll()方法

**1. 屏幕滚动方法，这里有几种使用方式：

1. 屏幕滚动 到底scroll(browser,'bottom')
2. 屏幕滚动 到顶scroll(browser,'Top')
3. 屏幕滚动 到底 平顺的滚动，会很慢 scroll(browser,'smooth')
4. 屏幕滚动 到数字位置scroll(browser,500) 移到500高度
5. 屏幕滚动 到元素位置scroll(browser,el) 移到el元素的位置（注意el元素你需要手写脚本获取）

像这样

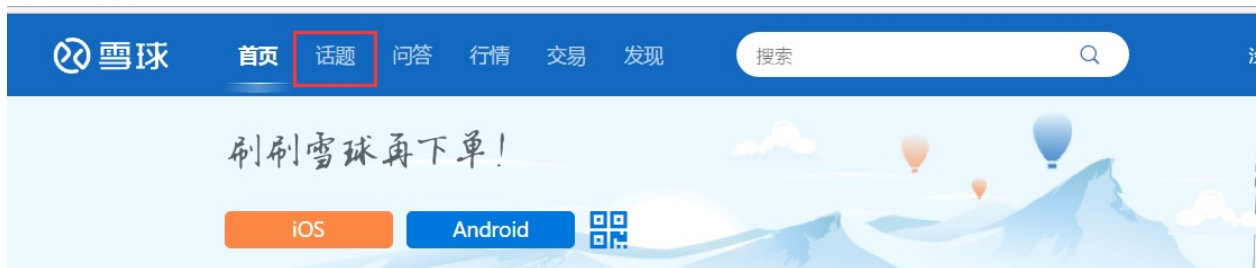
```
el=browser.find_elements_by_class_name('title')[3]
scroll(browser,el)**
```


3.3.4 find_by_text()方法

3.3.4 find_by_text()方法

举例 `el=find_by_text(browser,'BOBO')`

selenium中不支持通过元素的innertext文本内容查找元素。



如雪球首页，有个“话题”，我们可以通过`el=find_by_text(browser,'话题')`，直接获取到这个元素并`el.click()`操作点击。

3.3.5 show()方法

3.3.5show()方法

简单说是封装了selenium的ActionChains(browser).click_and_hold(el).perform()也就是元素悬停。
当然这个是配合find_by_text方法来用的。

举例

```
el=find_by_text(browser,'BOBO')  
show(browser,el)
```

3.3.6 time_start和time_end

3.3.6time_start()和time_end()方法

当需要测试某个页面的展示性能时，我们可以在进入页面的step的前后加上time_start()和time_end(),注意括号中给页面取个名字，如time_start(登录)，time_end(登录)。

A	B	C	D	E	F	G	
stepID	Action	思考时间	坐标	属性	步骤	输入	
step1	Key_Fun		pys	time_start(登录)			
step2	Key_Fun		pys	time_end(登录)			
step3	Mouse	1.91	702,434	tagName:INPUT,id:username,name:username,value:placeholder:请输入用户名,class:input_box,href:,innertext:,docker:	请输入用户名	admin	
step4	Mouse	1.37	703,497	tagName:INPUT,id:password,name:password,value:placeholder:请输入密码,class:input_box,password,href:,innertext:,docker:	请输入密码	1234	
step5	Key_Fun		pys	ActionChains(browser).key_down(Keys.ENTER).key_up(Keys.ENTER).perform()	请输入密码		
stepOver	Over						
step3	Mouse	3.50	739,600	tagName:BUTTON,id:,name:,value:placeholder:,class:button,href:,innertext:立即登录,docker:	点击---->立即登录		
1	stepID	B	C	D	E	F	G
2	step4	Mouse	3.41	54,139	tagName:A,id:a,qdgn_name,value:placeholder:,class:href,javascript:,innertext:更多功能,docker:	点击---->更多功能	输入
3	step5	Key_Fun		pys	time_start(模具维护)		
4	step6	Mouse	3.67	447,545	tagName:A,id:,name:,value:placeholder:,class:btn btn-default btn-app radius-4,href:javascript:,innertext:模具维护,docker:	点击---->模具维护	
5	step7	Key_Fun		pys	time_end(模具维护)		
6	step6	Mouse	1.43	133,126	tagName:BUTTON,id:,name:,value:placeholder:,class:btn btn-white btn-default btn-round,href:,innertext:新建,docker:	点击---->新建	
7	step9	Mouse	3.05	1348,230	tagName:INPUT,id:mouldNo,name:mouldNo,value:placeholder:请输入模具编号,class:form-control,href:,innertext:,docker:IFRAME	请输入模具编号	random:mj+int(4)
8	step10	Mouse	1.44	1305,285	tagName:INPUT,id:mouldName,name:mouldName,value:placeholder:请输入模具名称,class:form-control,href:,innertext:,docker:IFRAME	请输入模具名称	random:int(1)
9	step11	Mouse	0.89	1306,331	tagName:INPUT,id:mouldCycle,name:mouldCycle,value:placeholder:请输入周期,class:form-control,href:,innertext:,docker:IFRAME	请输入周期	random:int(1)
10	step12	Mouse	1.15	1300,383	tagName:INPUT,id:modules,name:modules,value:placeholder:请输入模穴数,class:form-control,href:,innertext:,docker:IFRAME	请输入模穴数	random:int(1)
11	step13	Mouse	1.42	1286,434	tagName:INPUT,id:openBoxNum,name:openBoxNum,value:placeholder:请输入开合数,class:form-control,href:,innertext:,docker:IFRAME	请输入开合数	random:int(1)
12	step14	Mouse	2.50	1338,486	tagName:SELECT,id:status,name:status,value:0,placeholder:,class:,href:,innertext:在用/停用,docker:IFRAME	点击---->在用/停用	
13	step15	Mouse	1.77	1323,526	tagName:SELECT,id:status1,name:status1,value:1,placeholder:,class:,href:,innertext:在用/停用,docker:IFRAME	点击---->在用/停用	
14	step16	Mouse	1	1405,575	tagName:A,id:,class:layui-layer-btn0,innertext:确定,docker:	点击---->	
15	step17	Key_Fun		pys	check(browser,'innertext:mj2035')	点击---->A	
16	step18	Mouse	3	301,256	tagName:A,id:,name:,value:placeholder:,class:fa-pencil,href:javascript:void(0),innertext:,docker:	点击---->A	at:mj5384
17	step19	Key_Fun		pys	sleep(1)	点击---->A	
18	stepOver	Over					
19							
20							
21							
22							

日志记录结果

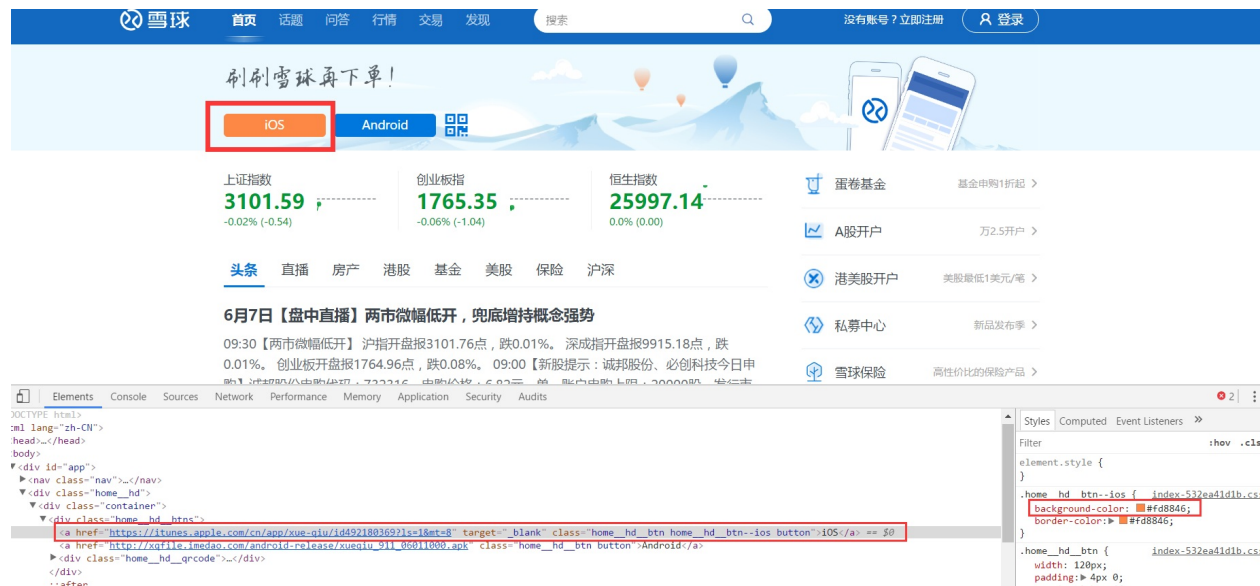
```

---- step2 开始
----- 页面加载总时间 : 427 毫秒
----- Request请求耗时 : 51 毫秒
----- TCP链接耗时 : 0 毫秒
----- 白屏时间 : 202 毫秒
----- DNS查询耗时 : 0 毫秒
----- DOMready时间 : 373 毫秒
----- 解析DOM树结构耗时 : 90 毫秒
----- 完成 登录 页面性能统计
---- step2 结束
  
```


3.3.7 check_style()方法

3.3.2 check_style()方法

当需要检查页面元素的style属性的时候，可以用到这个方法。



大家可以看到截图中，如果我们测试时需要检查红框位置IOS这个元素的背景颜色有没有变化。我们在我们的case里直接书写python语句。

第一行 `el=find_by_text(browser,'IOS')`

这里我们通过封装的find_by_text方法查找这个元素。

第二行 `check_style(browser,el,'background-color','rgb(255, 119, 0)')`

	A	B	C	D	E
1	stepID	Action	思考时间	坐标	属性
2	step1	Mouse	3.91	185,126	tagName:A,id:,name:,value:,placeholder:,class:dd_aTitle,href:http://shop.md168.cn/index/login/index.html,innertext:您好,请登录,docI
3	step2	Mouse	4.47	1130,412	tagName:INPUT,id:login_username,name:username,value:,placeholder:请输入手机号,class:form-control input-lg,href:,innertext:dock
4	step3	Key_Fun		pys	el=browser.find_element_by_id('btn-login') check_style(browser,el,'position','rgb(255,128,0)')
5	step4	Mouse	1.04	1132,471	tagName:INPUT,id:login_password,name:password,value:,placeholder:密码,class:form-control input-lg,href:,innertext:docker:,window
6	step5	Mouse	4.88	1144,544	tagName:BUTTON,id:btn-login,name:,value:,placeholder:,class:btn btn-primary btn-lg btn-block btn-login,href:,innertext:立即登录,dock
7	step6	Mouse	3.12	1023,121	tagName:A,id:,name:,value:,placeholder:,class:dd_aTitle,href:http://shop.md168.cn/member/enhome/index.html,innertext:我是供应商,
8	step7	Mouse	5.20	256,552	tagName:A,id:release,name:,value:,placeholder:,class:,href:http://shop.md168.cn/member/enhome/pronew.html,innertext:发布商品,dc
9	step8	Mouse	11.64	610,508	tagName:SELECT,id:pcate,name:,value:64,placeholder:,class:category-box,href:,innertext:注塑机,docker:,windows:0
10	step9	Mouse	1	910,535	tagName:SELECT,id:psubcate,name:,value:70,placeholder:,class:category-box,href:,innertext:破碎机,docker:,windows:0
11	stepOver	Over			
12					
13					

3.4 index属性

3.4 “index” 属性是什么？

通过增加案例中属性栏里，手动添加index属性，可以方便的指定页面相同属性的元素。因为碰到页面元素属性相同导致我们找到的元素是多个的时候，程序没有办法知道实际需要选择第几个元素。这时我们通过指定index属性，可以方便的使程序定位index指定的元素。

示例：

```
tagName:INPUT,id:,name:,value:上传图片,placeholder:,class:btn btn-success,href:,innertext:,docker:IFRAME,index:0
```

这是我们案例中属性单元格里的内容，index指定为0表示选择页面同属性第一个元素。

日志打印结果

```
---- step13 结束
---- step14 开始
----- 上传文件成功
---- step14 结束
---- step15 开始
----- 进入frame容器 成功
---- step15 结束
---- step16 开始
----- 进入frame容器 成功
----- 找到同属性元素 3 个
----- 当前index指定选择第 2 个
---- step16 结束
---- step17 开始
----- 上传文件成功
---- step17 结束
---- step18 开始
----- 进入frame容器 成功
----- 找到同属性元素 3 个
----- 当前index指定选择第 3 个
---- step18 结束
---- step19 开始
----- 上传文件成功
---- step19 结束
---- step20 开始
----- 进入frame容器 成功
---- step20 结束
```

3.5 innertext属性关联参数

3.5 “innertext” 属性关联参数

因为很多时候录制的元素innertext属性内容是固定的，并不能适用普遍情况。比如上一步动态生成的内容，如果名称是随机的或者是变化的，此时如果我们将这样的步骤输入指定一个参数的话，再后面的步骤需要操作这样生成的元素，而原始的录制元素的innertext属性又是固定的，此时我们只需要将案例中innertext设置为{参数名}即可，从而达到关联的目的。

如：`tagName:A,id:,name:,value:,placeholder:,class:,href:,innertext:{p1},docker:`
替换掉原始的innertext值，用大括号将参数名称传进来。