1. 准备篇

安装RobotFramework

安装Python

安装pip

安装Selenium：pip install robotframework-selenium2library

安装vc：vcforpython27.msi

安装模拟键盘：pip install sendkeys

1. RF关键字

|  |  |  |  |
| --- | --- | --- | --- |
| open brower | #打开浏览器 | | |
| open brower | url | brower= |  |
| open brower | http://www.bai.com | brower =Google Chrome | #打开浏览器 |

|  |  |  |  |
| --- | --- | --- | --- |
| Set Browser Implicit Wait | 设置浏览器等待时间（单位为秒） | | |
| Set Browser Implicit Wait | Seconds |  |  |
| Set Browser Implicit Wai | 20 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| input text | 向文本框输入值 locator表定位器，text表输入内容 | | |
| input text | iput\_username |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Click |  | | |
| Click Button | locator | #单击按钮 |  |
| Click Link | locator | #单击A标签 |  |
| Click Element | Locator | #单击某个标签 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Element should contain | 验证元素被定位器包含文本预期 | | |
| Element should contain | Expected | Message |  |
|  | 期望值的locator | 想要的消息 |  |

这个例子说明了用户名和密码为空下提示“用户不存在”



1. AutoItLibrary安装

下载

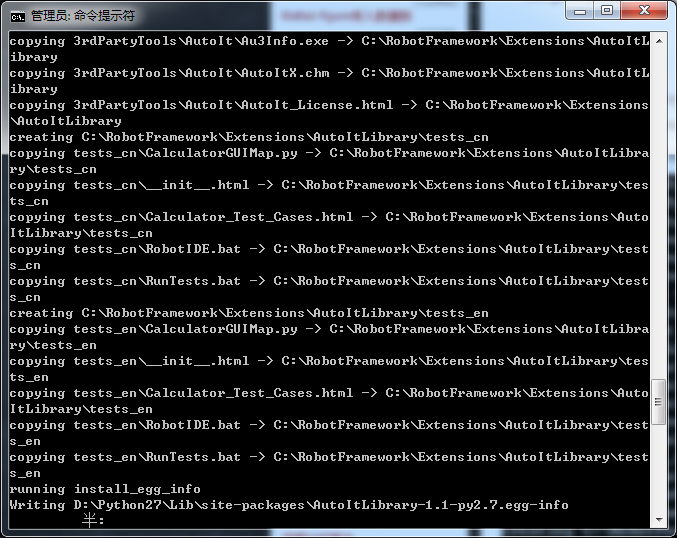
与Python版本和系统匹配的

pywin32：http://sourceforge.net/projects/pywin32/files/pywin32/

AutoItLibrary-1.1.tar.gz （必须先安装pywin32）

安装：

解压AutoItLibrary后，管理员身份运行cmd，cd进入下载包文件夹路径，执行python setup.py install

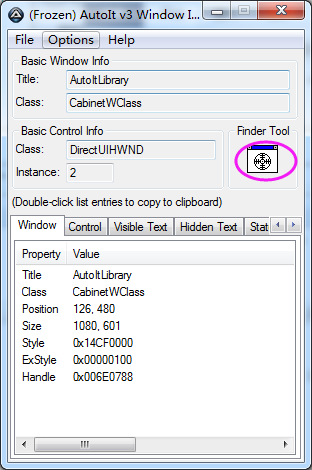


当安装完AutoItLibrary后，User目录盘根目录会多出一个RobotFramework的目录，这里是一些辅助工具，如

AutoItX.chm ——帮助文档

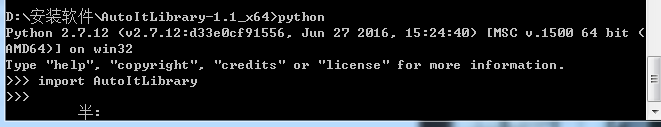
AutoItLibrary.html——测试库关键字文档数目

Au3Info.exe——识别对象工具



在Finder Tool的位置有个十字星，可以用鼠标拖动他到你需要识别的对象上

进入python shell 环境，通过输入import AutoItLibrary，若未保存则表示AutoItLibrary安装成功。



64位机器还需要安装一个AutoIt V3，

http://download.pchome.net/development/linetools/down-20198-7.html 中下载AutoIt V3

通过进入RobotFramework界面，通过导入AutoItLibrary库，查看是否显示正常（黑色）



1. 规范化
2. 规范化
3. Robot Framework
   1. 资源

Find Usages 按钮：Resource界面，查找关键字

加载资源：添加资源的完整文件名（包括扩展名）

External Resource：外部资源，主要指不再Project管辖范围内的资源文件

加载资源路径格式：使用“\\”或“/”，因为“\”会被认为转义字符的符号

资源是带蓝色下划线链接的

* 1. 变量与常量

Scalar变量：${var} 单值变量

List变量：@{lvar} 多值变量

Dictionary变量： &{Dict}

注意变量的作用域

常量：环境变量标识符“%”,其他“$”

log %{ANDROID\_HOME}

特殊字符常量：${/} 、${:} 、${EMPTY}、${False} 、${None} 、${null} 、${SPAACE} 、${True}

变量赋值：Set赋值、Get 赋值、命令行赋值

Set Variable 关键字

1. 用于设置Scalar变量
2. 把一个List变量或者多个Scalar变量形成的List转换给一个Scalar变量
3. 设置List变量（不建议使用）

Create List 关键字

①创建新List

关键字的参数，如果变量名前有※，则支持传入List参数，否则只能Scalar

List元素使用：@{变量名}[index]或${变量名[index]}

* 1. 循环&分支

FOR循环结构：

：FOR 循环变量 IN RANGE end；

：FOR 循环变量 IN RANGE start end[step]

或

：FOR 循环变量 IN a[b,c……]

注： []里为非必需参数

测试用例中尽量减少分支、以保证测试结果的准确性

* + 1. IN RANGE end

IN RANGE 用于一个整数序列做循环的计数器，仅写end，表示从默认0开始

* + 1. IN RANGE start end

start end，可以实现从非0开始的循环

step，表示步进，默认+1，可以设置为其他正数或负数

若 start ＞ end，则step使用负数

若 start ＜ end，则step使用正数，否则循环将无法执行

正数：

:FOR ${i} IN RANGE 1 10 2

LOG i=${i}

负数：

:FOR ${j} IN RANGE 10 2 -2

LOG j=${j}

* + 1. IN

可见IN后的内容当做List的全部元素，IN循环相当于遍历每个元素

* + 1. 双重循环

RF中目前不支持，但是可以通过将一个循环设置为关键字，调用，来间接实现

分支，ELSE IF、ELSE前需添加“…”三个点，否则RF无法识别

* + 1. 常用关键字

1. 浏览器管理

Open Browse/Close Browser 打开关闭浏览器

Go to/Go Back 前进后退

Select Windows/Close Windows 打开关闭窗口

Switch Browser 切换浏览器，参数是打开浏览器的index顺序或alias别名

Close All Browsers 关闭所有浏览器

Select Windows，通过handle去操作窗口的关键字，

select window by handle：通过handle选择window；

get window handle：获取当前线程所有窗口的handles，返回list

get current window handle：获取当前窗口的handle

Select Frame/Unselect Frame：采用了frame或iframe，这里的对象无法直接操作，要先选择fram，在选择frame里的元素

1. Cookie

Cookie是网站为了辨别用户身份存储在用户本地的数据（之一？）

add cookie：添加Cookie

delete cookie

delete all cookie

get cookie：获取所有cookie

get cookie value：获取cookie的值

1. Elements

Element

locator，定义页面元素，常见的locator有id、name（可能存在同名），其他定位手段包括xPath、css、jQuery、sizzle等

Assign id to element:如果一个对象没有id，可以指定一个id，然后在操作，为locator瘦身

鼠标操作模拟：mouse down，mouse up，mouse over，mouse out等

坐标定位：Get horizontal position，Get vertical position，获取当前对象的左上角顶点距离整个页面的横纵左边

获取元素的属性值，Get value、Get Text，若要获取元素其他属性值，可使用Get element attribute，如获取class值：Get element attribute\_id@class

formElement

Sumbit Form:提交表单

CheckBox部分：Select CheckBox/ Unselect CheckBox 勾选/不勾选复选框

Radio buttons：Select Radio button

text fields部分：

Choose File：用于上传文件时选择文件

input text：输入文字内容

input Password：与text实质一样，只是输入文本为密文

为了精简用例，方便后续优化用例，可以将整个用例分层设计，如

用例层：测试流程+数据

流程层：元素操作 如打开浏览器、输入用户名

元素层：测试库封装 如Open Browser封装成打开浏览器

1. AutoItLibrary

主要是针对Windows平台设计的 模拟键盘、鼠标、窗口、控件操作进行自动化

1. 安装pywin32
2. 安装AutoItLibrary
3. 安装AutoIt（针对64位系统）
   1. AutoItLibrary关键字
      1. 关键字

主要分为以下几大类

Window操作：如果有多个窗口，则需要把要操作的窗口激活后才能操作

Control操作：窗口上的按钮、文本框等，先激活窗口，在操作

Mouse操作：鼠标操作要用到真实坐标，和Selenium2Library里的坐标有差异

Process操作：对进程操作，用RF中自带的Process库就OK了

Run操作： 启动一个程序或运行一个命令，用RF中自带的Process库就OK

Reg操作： 对注册表进行操作

* + 1. 对象识别

1. 的萨达三发