

Robot Framework + Selenium2/WebDriver 测试技术培训

@吴穹Adam



- 基本概念



- 定义一种统一的方式来书写和组织测试用例（测试步骤，测试数据与期待返回值）；
- 集成不同的测试驱动技术；
- 控制测试用例的执行过程；
- 生成测试报告和测试日志；



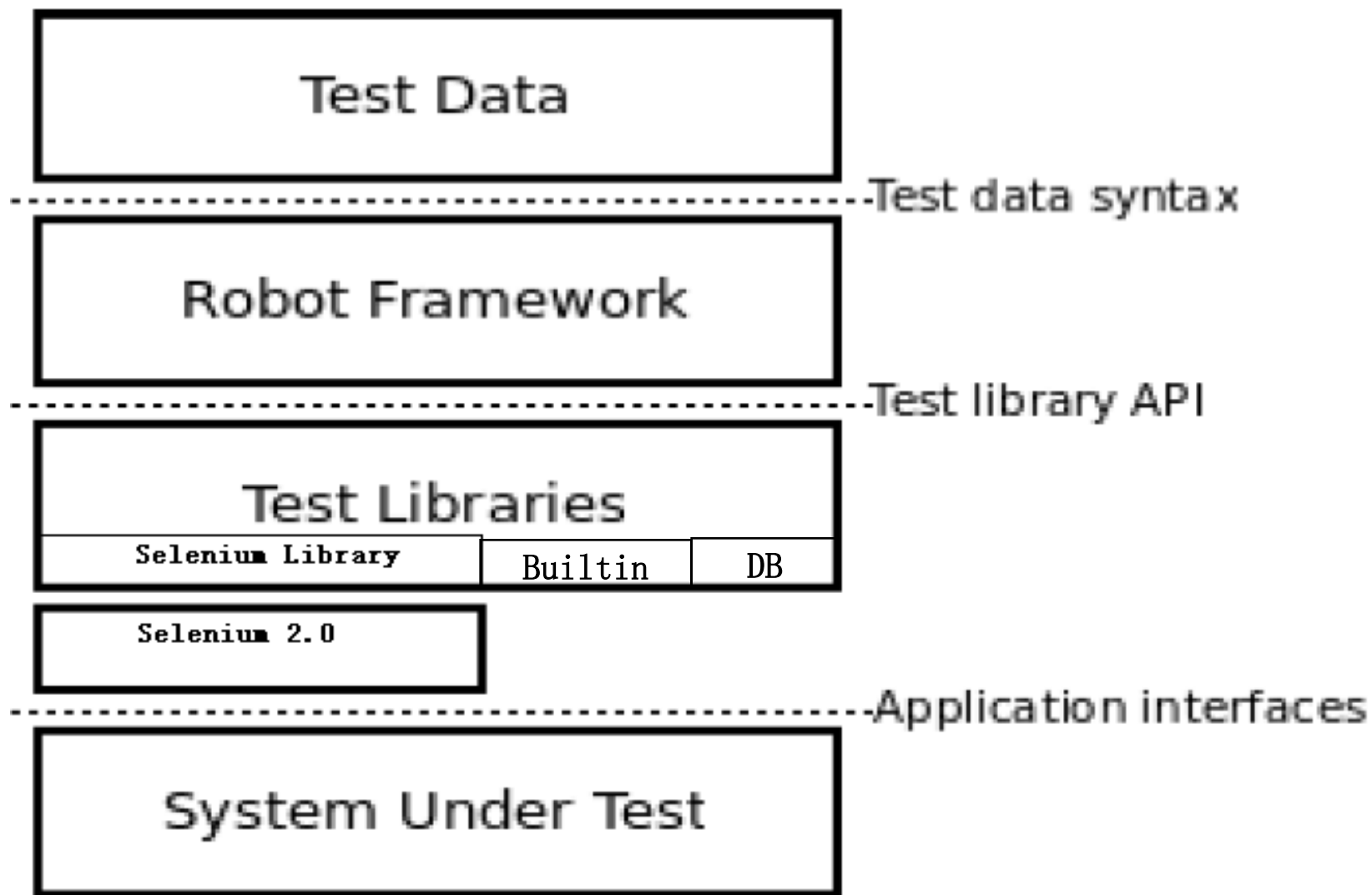
- **Generic test automation framework**
 - Utilizes the keyword-driven testing approach
- **Implemented with Python**
 - Jython supported fully and IronPython initially
 - Can be extended natively using Python or Java
 - Other languages supported via XML-RPC based remote interface
- **Open source**
 - Apache 2.0 license
 - Sponsored by Nokia Siemens Networks
 - Active development and growing community
 - Hosted on Google Code



- 易于使用，采用表格式语法，统一测试用例格式；
- 重用性好，可以利用现有关键字来组合新关键字；
- 支持变量；
- 支持创建基于数据驱动的测试用例。
- 结果报告和日志采用HTML格式，易于阅读；
- 提供标签以分类和选择将被执行的测试用例；
- 平台、应用无关；
- 功能全面，支持WEB测试 (Selenium)、Java GUI 测试，启动线程、终端、SSH等；
- 易于扩展，提供了简单API，用户可以自定义的基于Python或者Java的测试库；
- 易于集成，提供了命令行接口和基于XML的输出文件；
- 易于与版本管理集成；

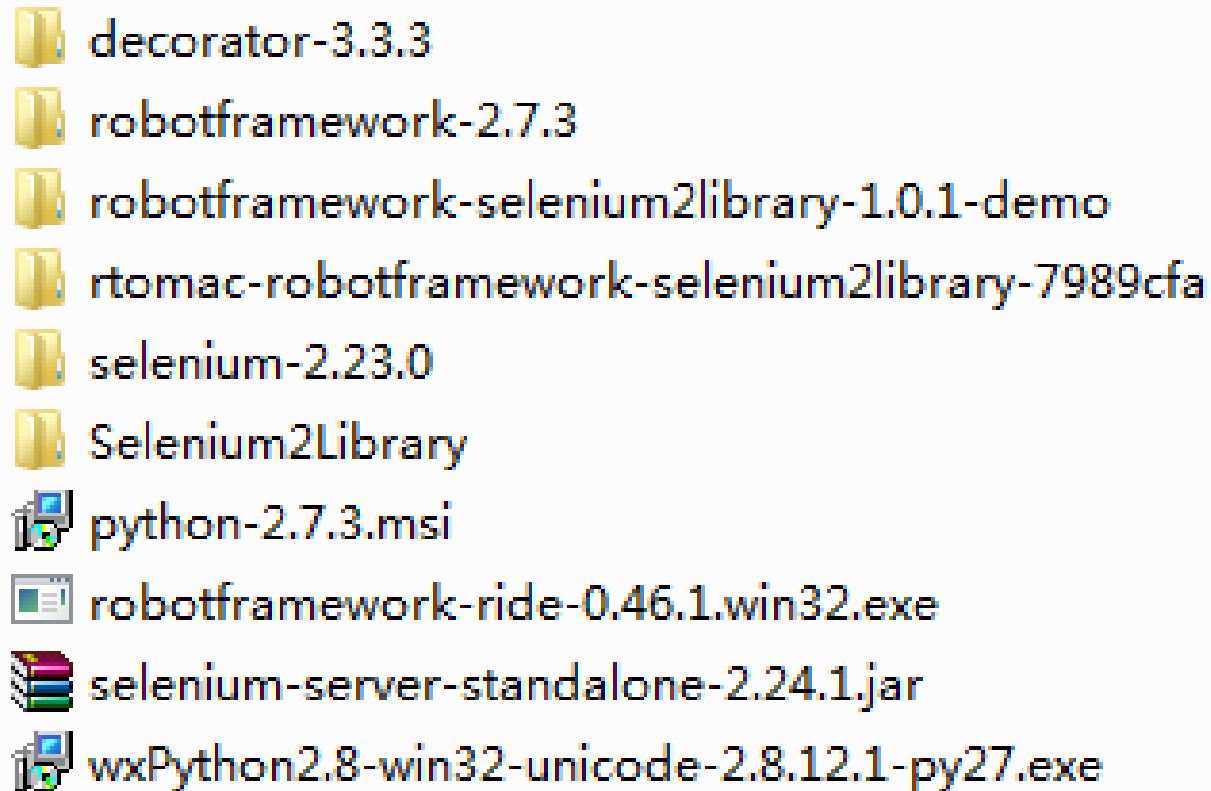


Robot Framework 方案 架构



Agenda

- 基本概念
- 安装



- **运行**

- python-2.7.3.msi (需要本机管理员权限)

- 请选择将Python加入Path

- wxPython2.8-win32-unicode-2.8.12.1-py27.exe,

- robotframework-ride-0.46.1.win32.exe

- 用管理员帐户打开一个cmd

- robotframework-2.7.3>python setup.py install

- decorator-3.3.3>python setup.py install

- selenium-2.23.0>python setup.py install

- rtomac-robotframework-selenium2library-7989cfa>python setup.py install

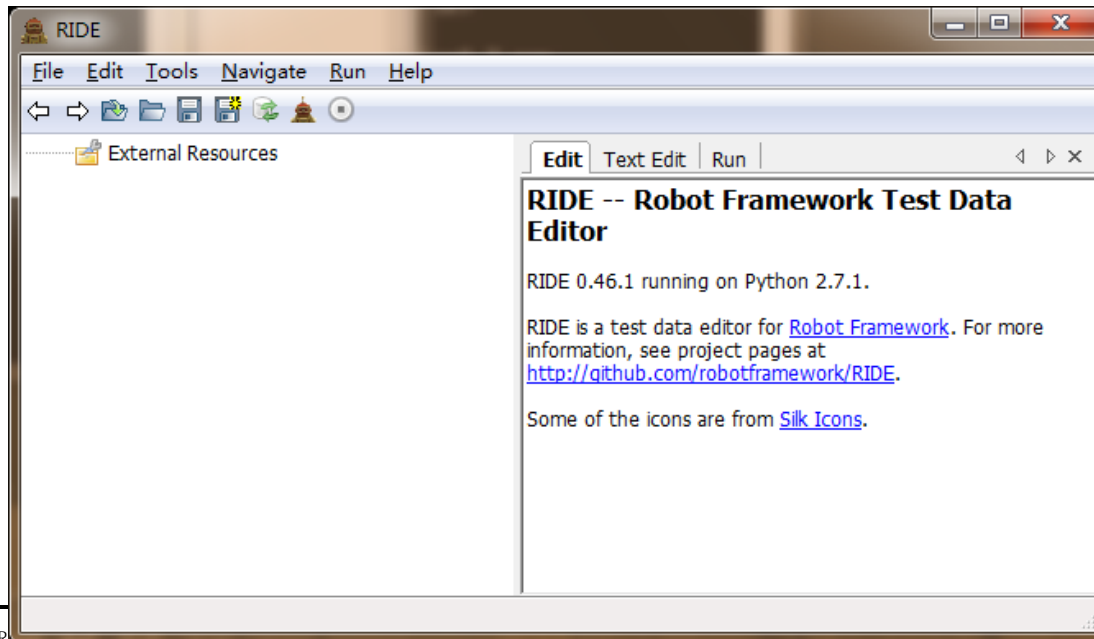


- 覆盖

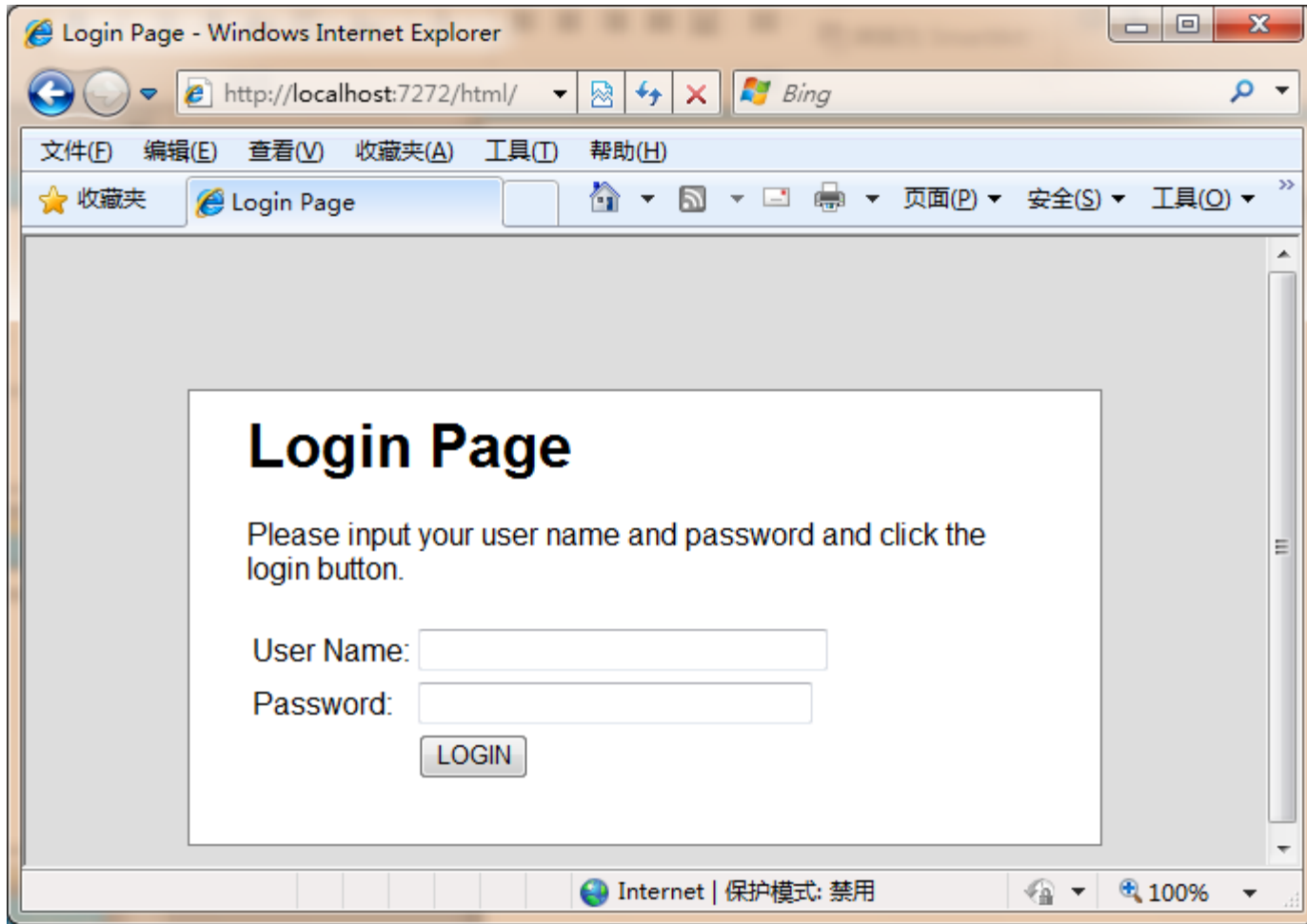
- 用Selenium2Library内容覆盖其中的两个文件

- 从RF Install\Selenium2Library到
 - [PythonDir]\Lib\site-packages\robotframework_selenium2library-1.0.1-py2.7.egg\Selenium2Library

- 执行[PythonDir]\Scripts\ride.py



- robotframework-selenium2library-1.0.1-demo>python rundemo.py demoapp start

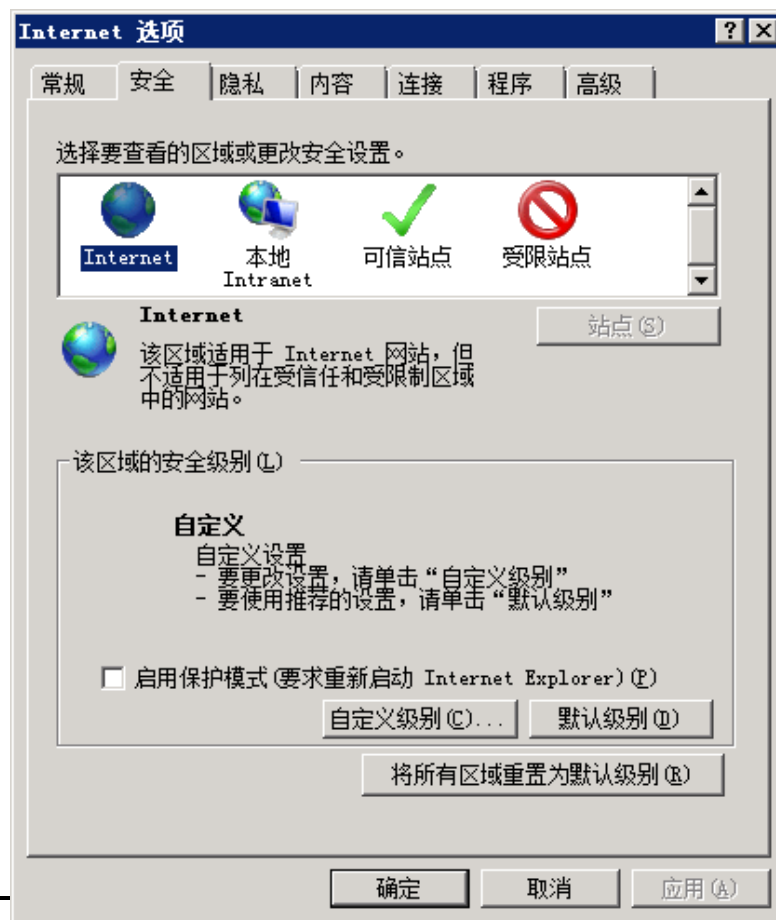


- 在 [PythonDir]\Lib\site-packages\robot\utils 下的 encoding.py 文件中，找到 def _get_output_encoding(): 原来的编码是 'cp437' 将其改成 'cp936' (简体中文, gbk)。

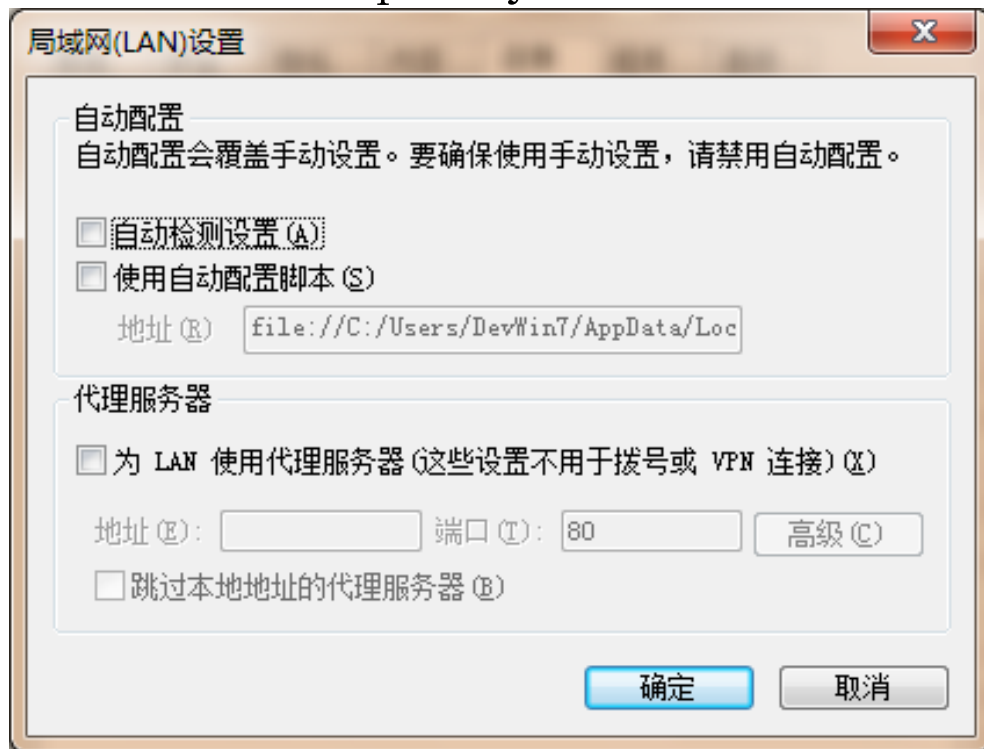
```
44 def _get_output_encoding():
45     # Jython is buggy on Windows: http://bugs.jython.org/issue1568
46     if os.sep == '\\':
47         return 'cp936' # Default DOS encoding
48         #return 'cp437' # Default DOS encoding
49     encoding = _get_encoding_from_std_streams()
50     if encoding:
51         return encoding
52     if os.sep == '/':
53         return _read_encoding_from_unix_env()
54     return 'cp936' # Default DOS encoding
55     #return 'cp437' # Default DOS encoding
```

- 需要以管理员身份运行IE，将所有四个区域的“启用保护模式”选项清除，否则 WebDriver会报这个错误

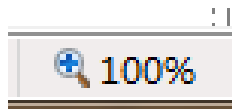
WebDriverException: Message: u'Unexpected error launching Internet Explorer. Protected Mode must be set to the same value (enabled or disabled) for all zones.'



- 确认IE没有设置proxy



- 设置放大比例为100%



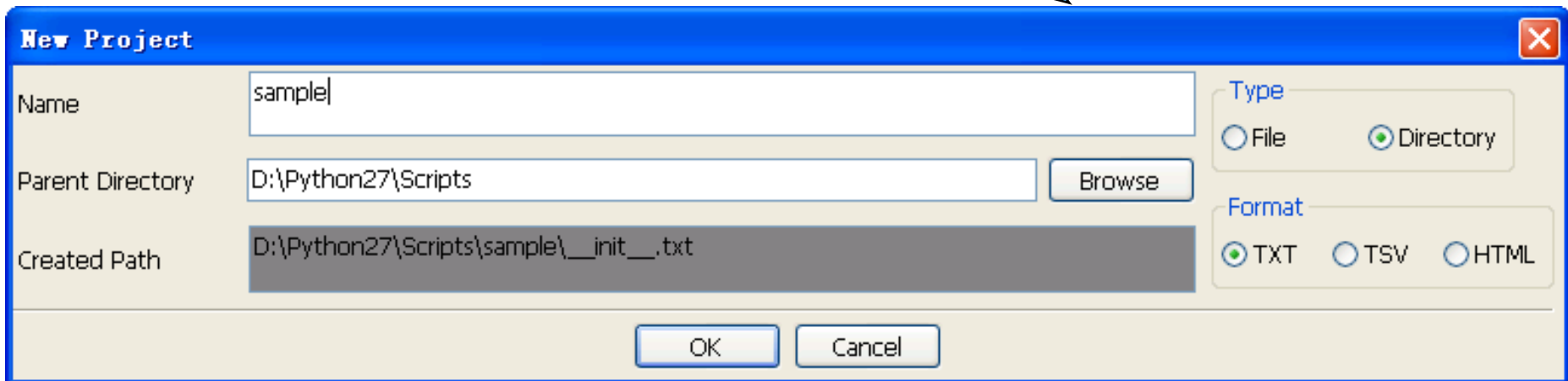
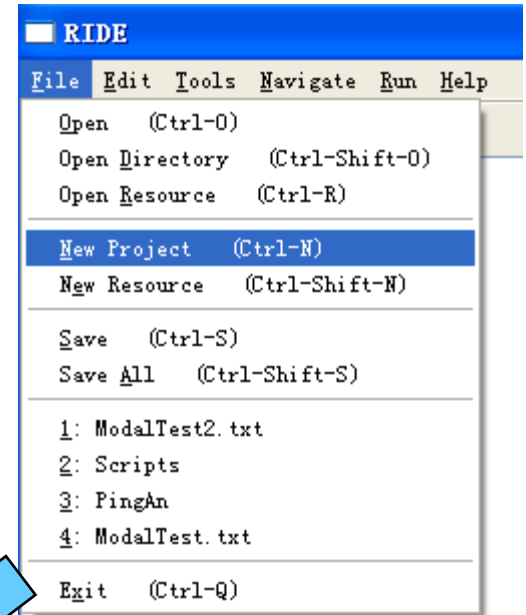
Agenda

- 基本概念
- 安装
- 创建测试用例



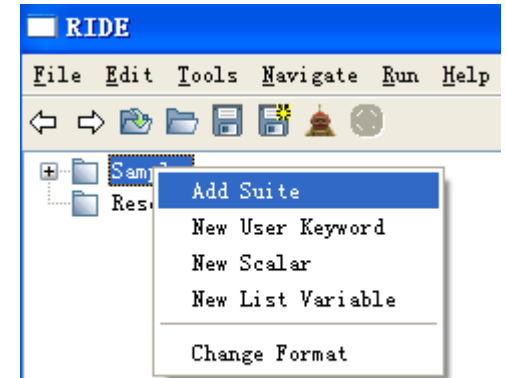
创建测试项目

- 测试项目可以目录或文件存储
- 格式可以用TXT, TSV或HTML格式
- 建议选择目录和TXT, 这样便于管理



创建测试套

- 测试套也可以选择是文件还是目录

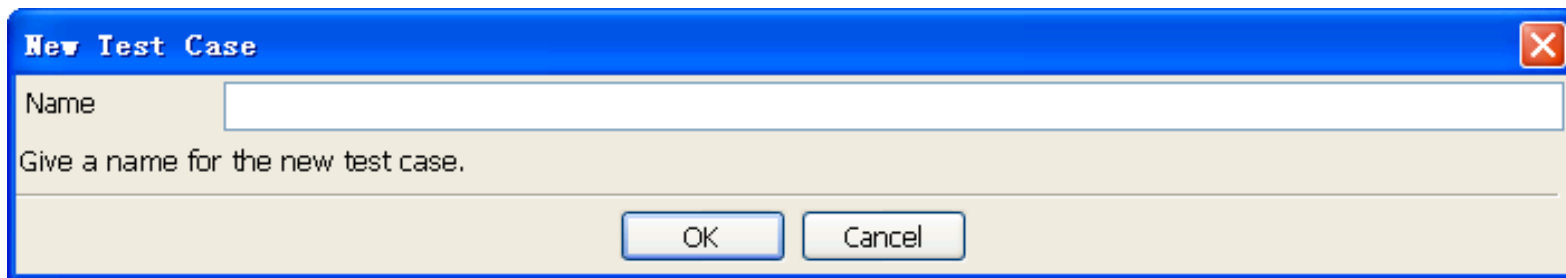
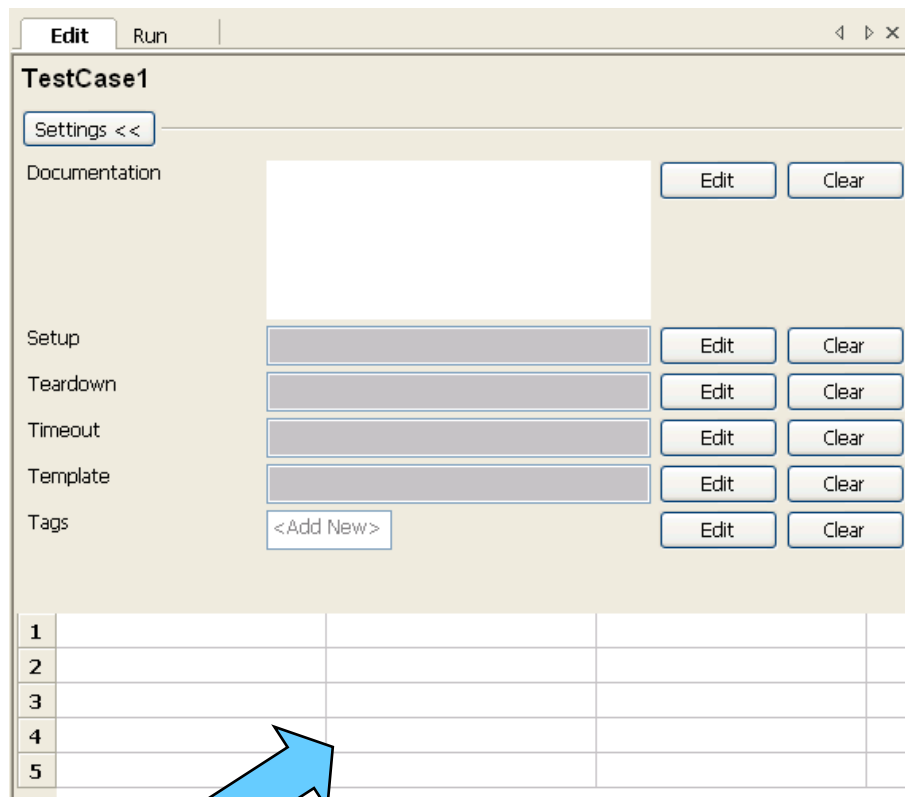
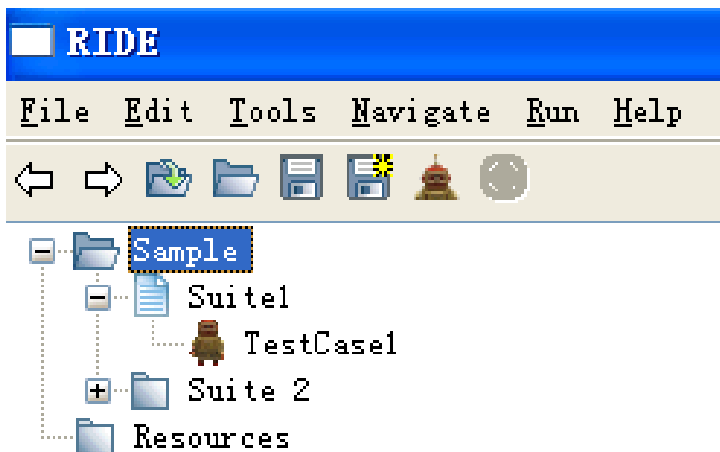


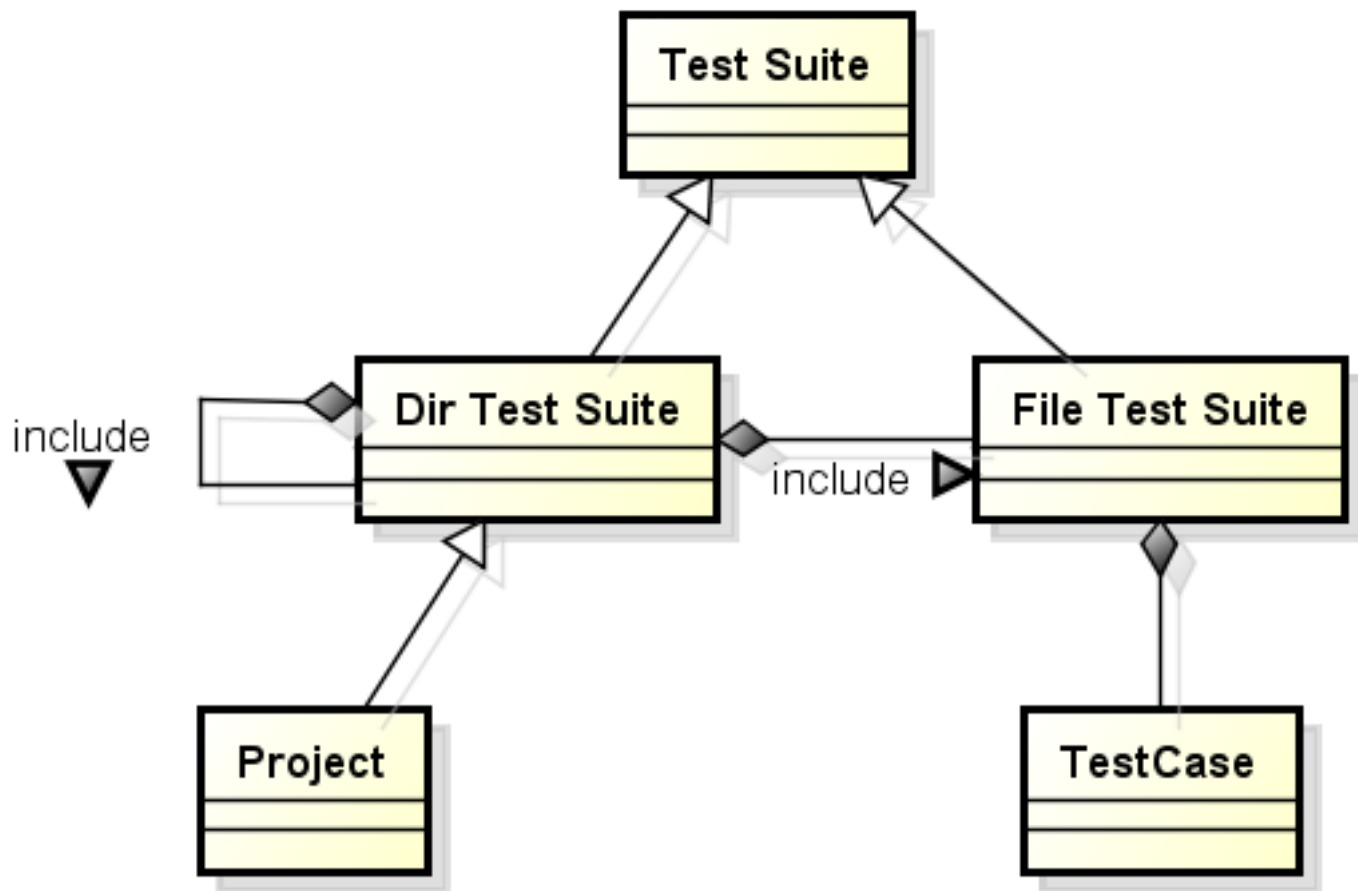
Add Suite

Name	<input type="text" value="Suite1"/>	Type	<input checked="" type="radio"/> File <input type="radio"/> Directory
Parent Directory	<input type="text" value="D:\Python27\Scripts\sample"/>	Format	<input checked="" type="radio"/> TXT <input type="radio"/> TSV <input type="radio"/> HTML
Created Path	<input type="text" value="D:\Python27\Scripts\sample\Suite1.txt"/>		



创建测试用例





- 测试库提供针对特定应用的关键词

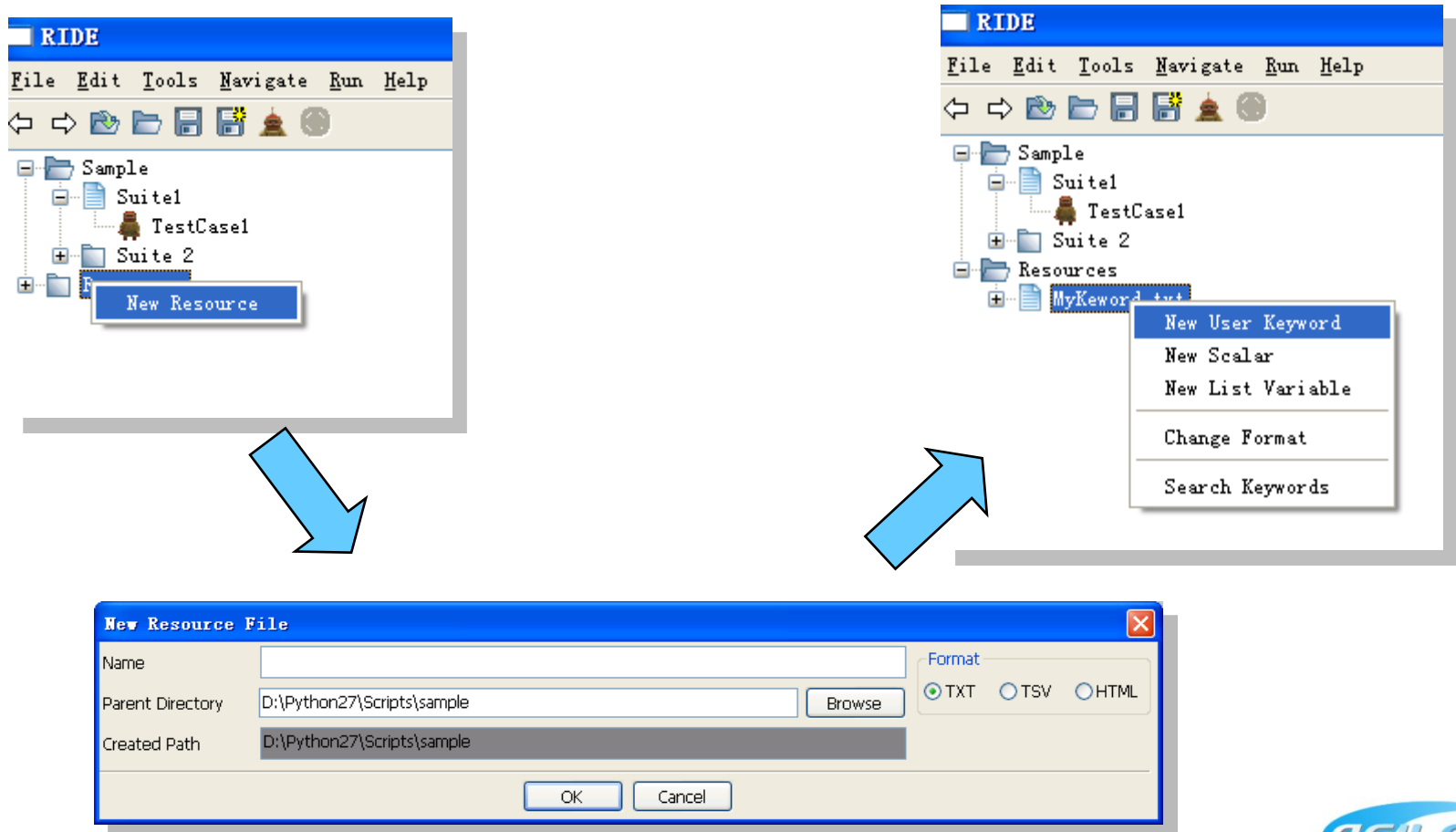
必须注意
大小写

The screenshot illustrates the process of adding a test library in Selenium IDE. It shows the 'Settings >>' window with the 'Add Library' button circled in red. A blue arrow points from this button to the 'Library' dialog box. In the dialog, the 'Name' field contains 'Selenium2Library'. Another blue arrow points from the 'OK' button to the 'Library' entry in the table below. The table has columns for 'Import', 'Name / Path', 'Arguments', and 'Comment'. The 'Library' entry is circled in red, and the 'Add Library', 'Add Resource', and 'Add Variables' buttons are visible on the right.

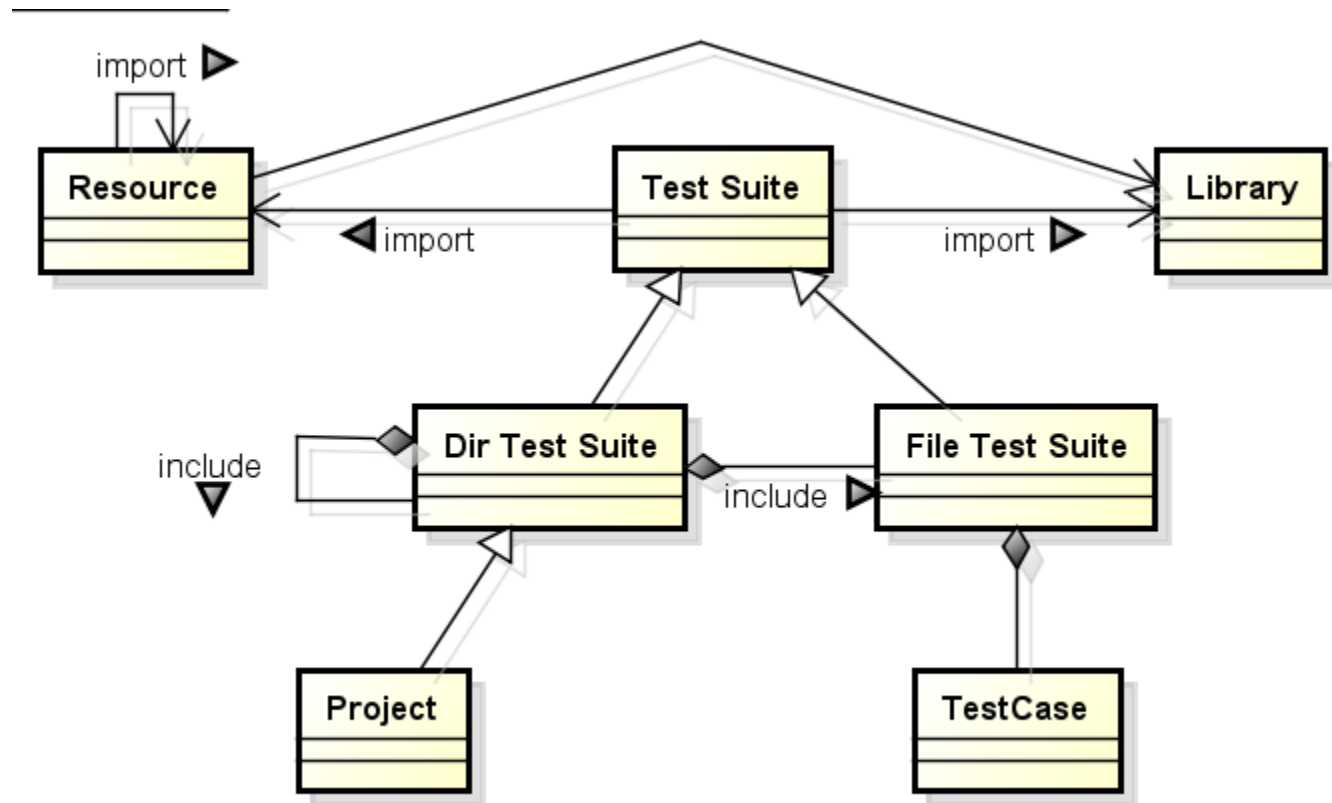
Import	Name / Path	Arguments	Comment
Library	Selenium2Library		

添加资源

- 资源提供定制公用关键词的能力
- 注意：资源也需要引用用到的Library



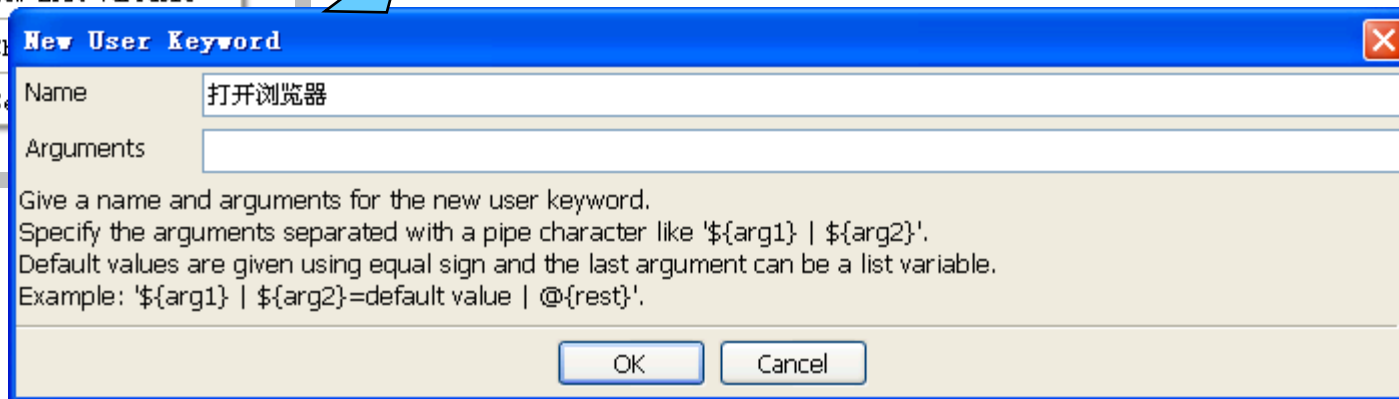
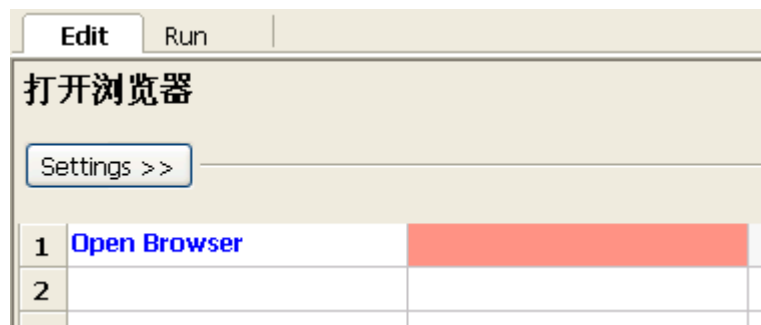
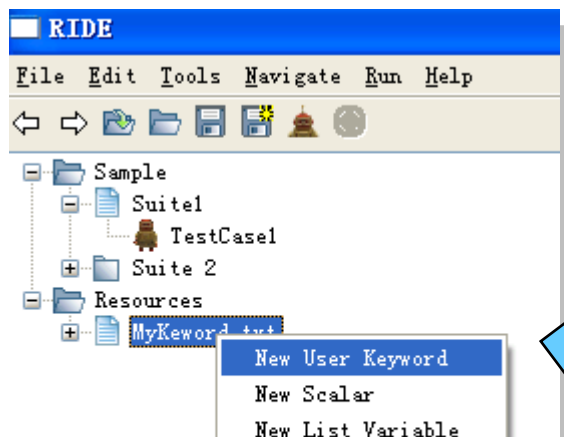
关系总结



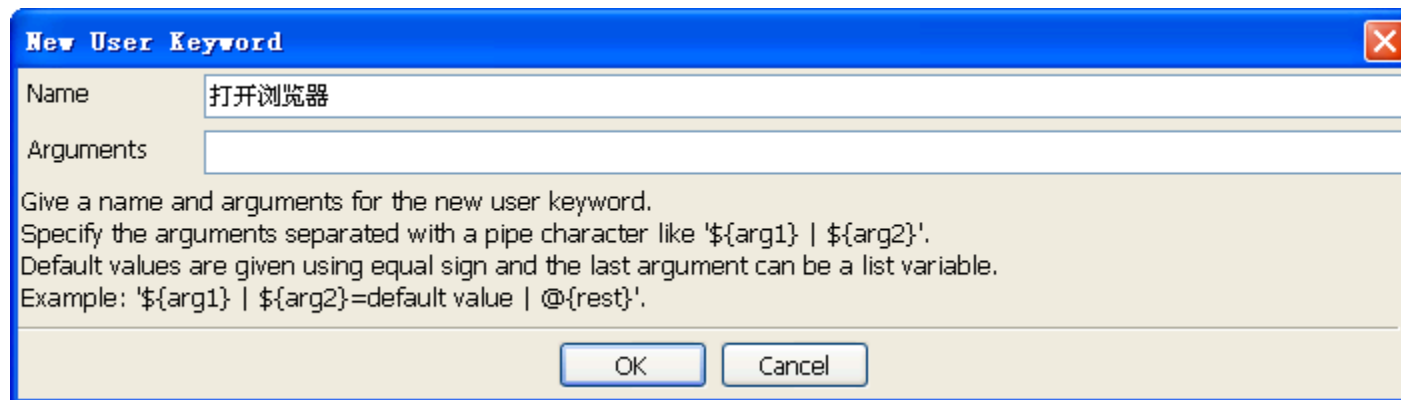
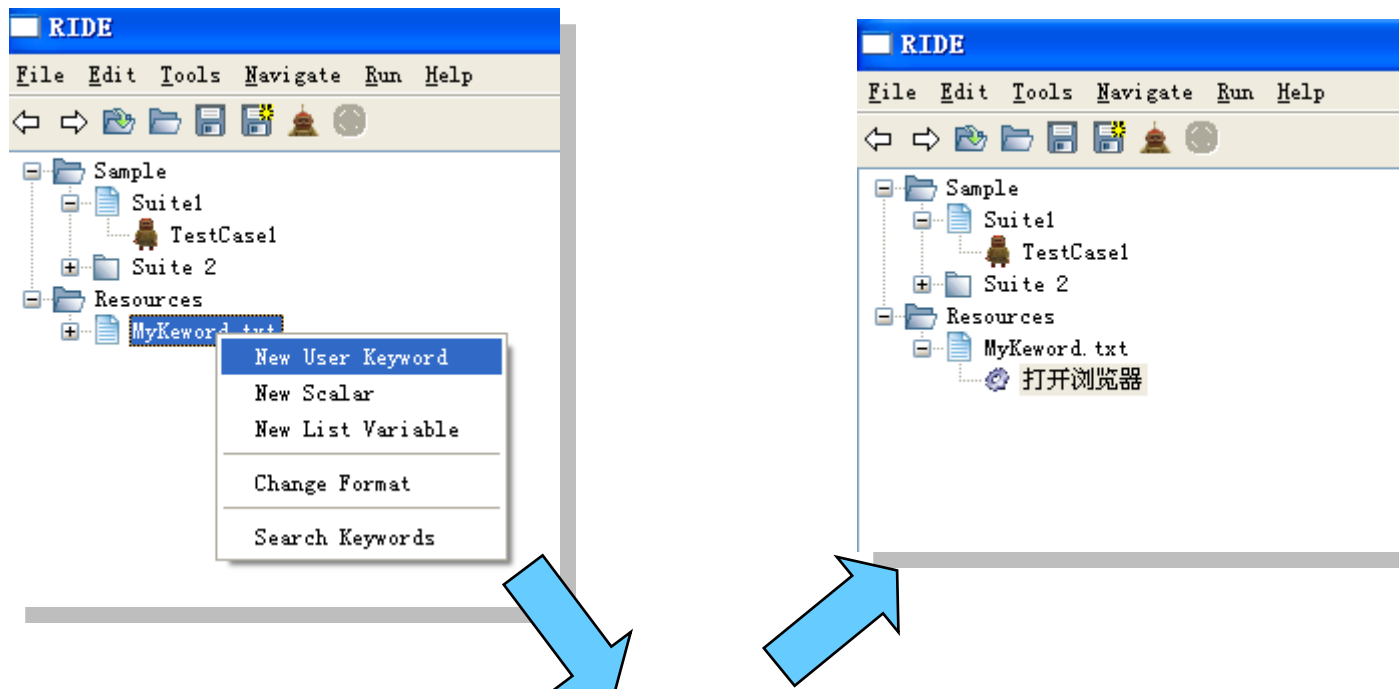
powered by astah*



增加自定义关键词



增加自定义关键词



- F5可以搜索关键词

The image shows two screenshots from Selenium IDE. The left screenshot is the 'Search Keywords' dialog, and the right screenshot shows the configuration for the 'Open Browser' keyword.

Search Keywords Dialog:

- Search term: Open
- Source: <all keywords>
- Table of keywords:

Name	Source	Description
Open Browser	Selenium...	Opens a new browser instance
Open Context Menu	Selenium...	Opens context menu on element
Close All Browsers	Selenium...	Closes all open browsers
Close Window	Selenium...	Closes currently opened window
Press Key	Selenium...	Simulates user pressing key
Select From Flex Element	Selenium...	Selects 'value' from an element found by 'locator'
Select Window	Selenium...	Selects the window found with 'locator' as the container
Set Selenium Timeout	Selenium...	Sets the timeout used by various keywords.
Start Selenium Server	Selenium...	Starts the Selenium Server provided with SeleniumLibrary
Switch Browser	Selenium...	Switches between active browsers using index or alias

Open Browser Keyword Configuration:

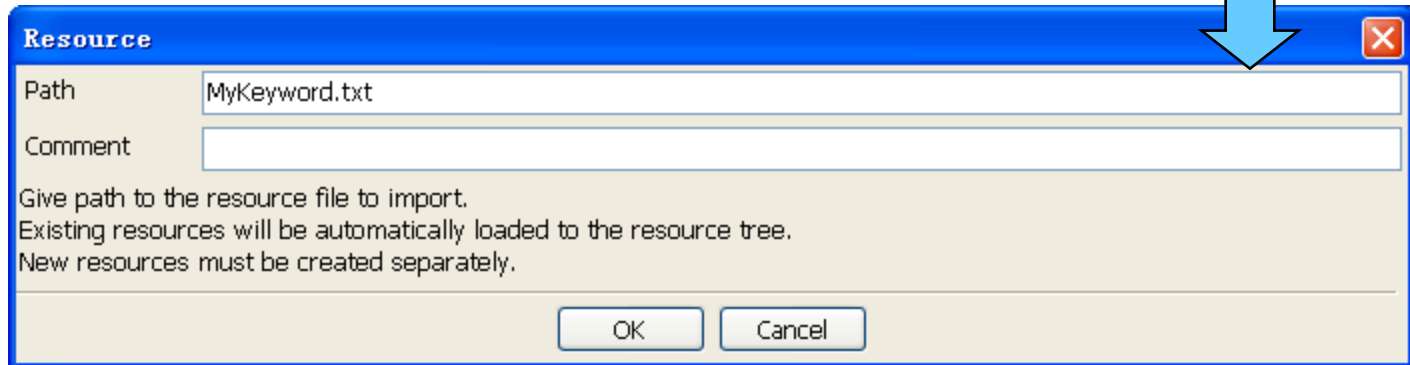
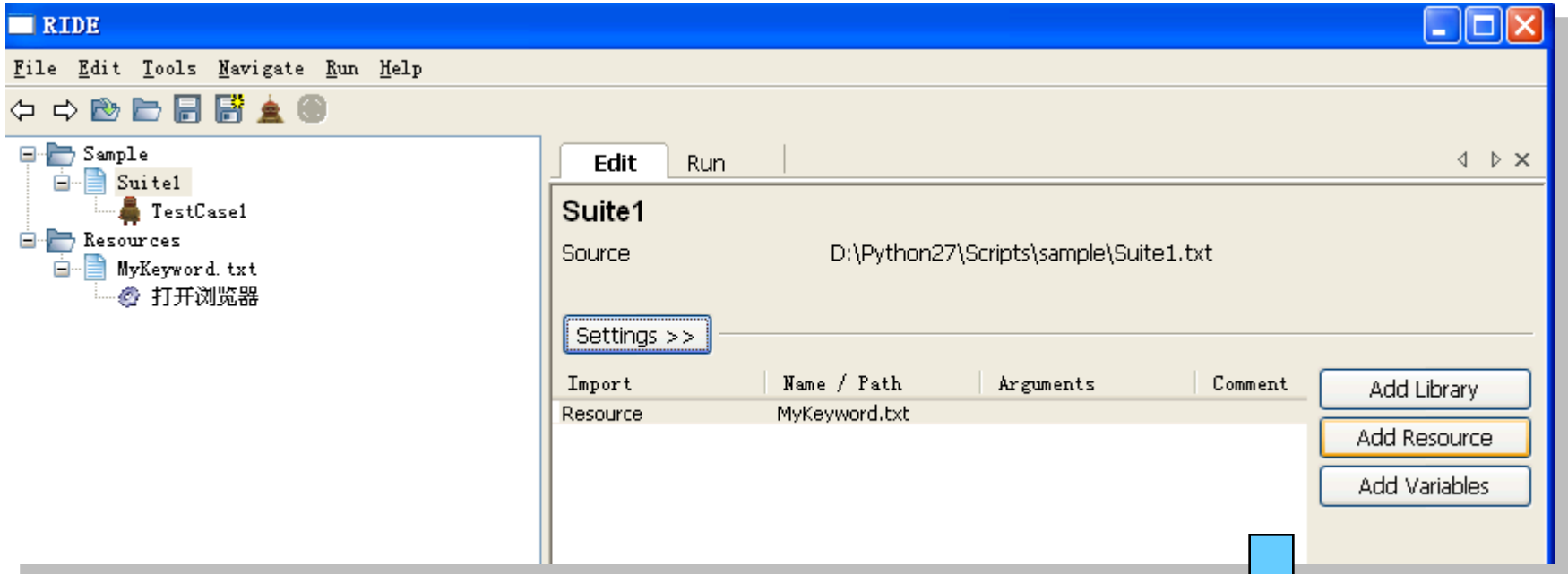
Keyword: 1 Open Browser

Index	Keyword	Value	Alias
1	Open Browser	http://localhost:7272/html/	ie

Red text annotation: 红色表明需要输入参数 (Red indicates that parameters need to be entered).

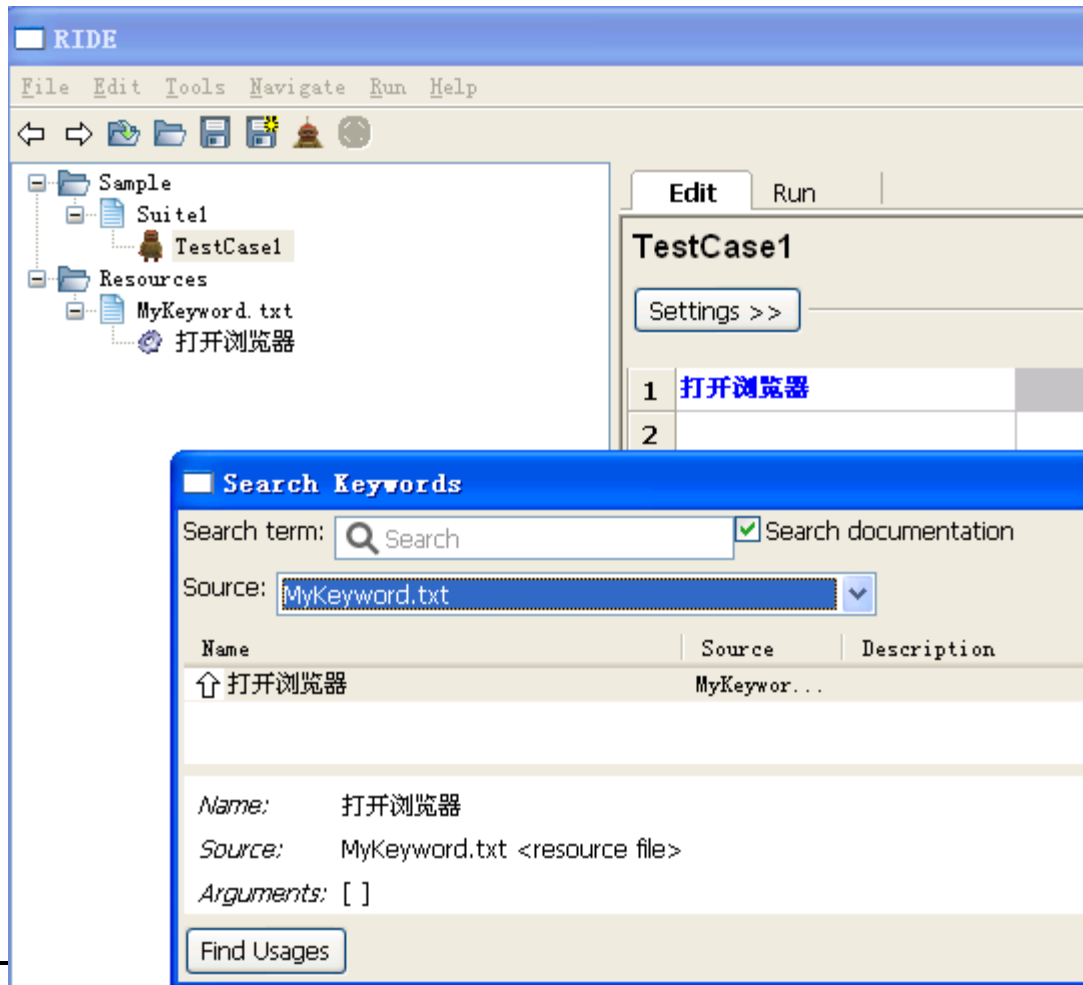
在测试用例里面引用资源

- 测试用例用到的资源，需要在测试套里面直接引用



引用自定义关键词

- F5可以查找关键词
- 匹配的关键词会显示为蓝色



Agenda

- 基本概念
- 安装
- 创建测试用例
- 执行测试



执行测试

The image displays two screenshots of the RIDE (Robot IDE) interface, illustrating the process of starting a test run.

Top Screenshot: The RIDE window is shown with the 'Run' tab selected. The 'Start' button (represented by a bell icon) is circled in red. The 'Execution Profile' is set to 'pybot'. The 'Arguments' field is empty. The 'Only run tests with these tags' and 'Skip tests with these tags' checkboxes are unchecked. The tree view on the left shows the project structure: Sample, Suite1, Resources, and MyKeyword.txt.

Bottom Screenshot: The RIDE window is shown after the test run has started. The 'Start' button is now disabled, and the 'Stop' button (represented by a stop icon) is active. The 'Arguments for the test run. Arguments are space separated' field is visible. The tree view on the left shows the test results: Sample, Suite1, and Test Case 1. The output window on the right displays the following information:

```
elapsed time: 0:00:13 pass: 0 fail: 0  
working directory: D:\Python27\Scripts  
command: pybot.bat --argumentfile c:\docu  
=====
```

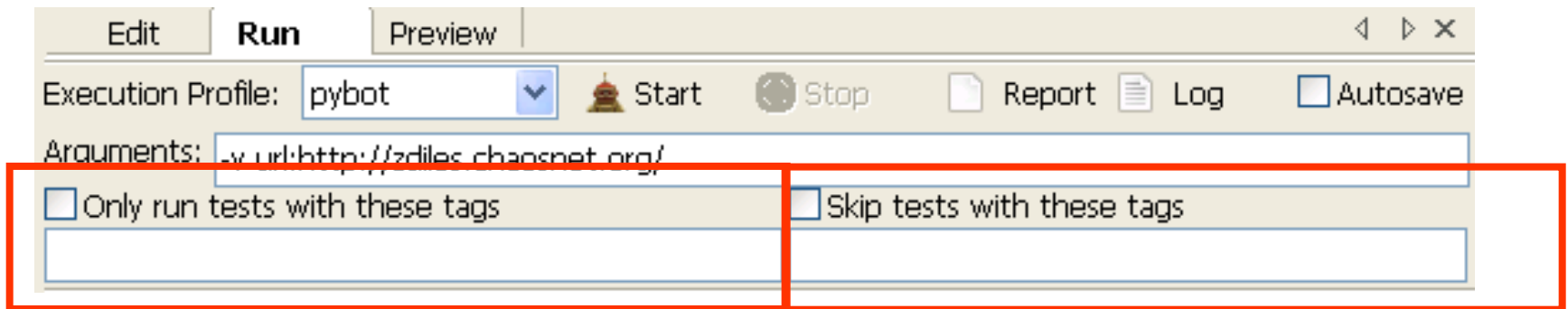
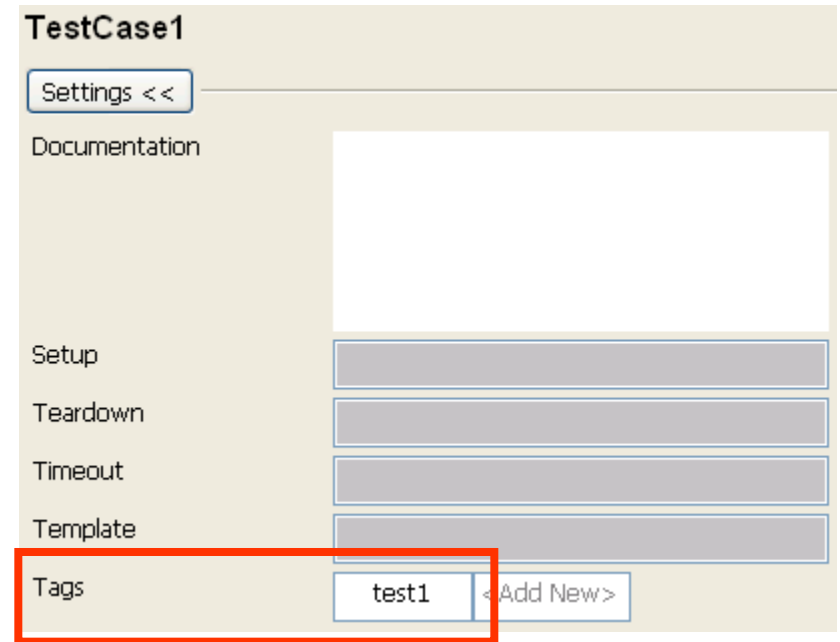
Sample
=====

Sample.Suite1
=====

Test Case 1

按Tag执行

- 测试用例可以标记多个Tag
- 可以按Tag来执行或跳过用例



- 测试日志按树形组织
- 可自动提供错误步骤的屏幕截图

Sample Test Log

Test Statistics

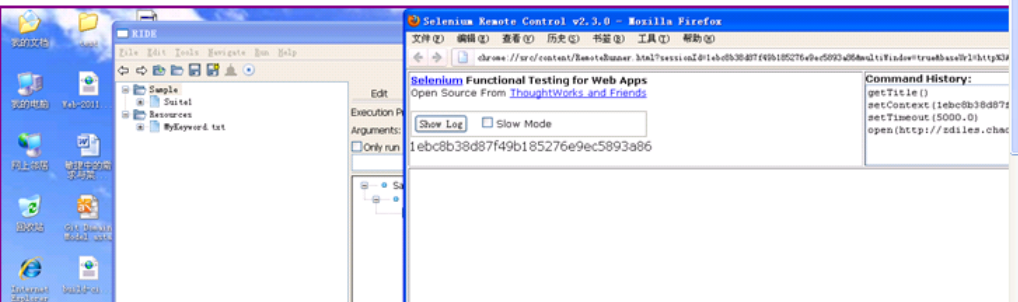
Total Statistics	
Critical Tests	
All Tests	

Statistics by Tag	
No Tags	

Statistics by Suite	
Sample	
Sample.Suite1	

Test Execution Log

<input type="checkbox"/> TEST SUITE: Sample	
Full Name:	Sample
Source:	d:\python27\scripts\sample
Start / End / Elapsed:	20110902 15:00:36.203 / 20110902 15:00:50.1
Overall Status:	FAIL
Message:	1 critical test, 0 passed, 1 failed 1 test total, 0 passed, 1 failed
<input checked="" type="checkbox"/> SETUP: SeleniumLibrary.Start Selenium Server	
<input checked="" type="checkbox"/> TEARDOWN: SeleniumLibrary.Stop Selenium Server	
<input type="checkbox"/> TEST SUITE: Suite1	
Full Name:	Sample.Suite1

<input type="checkbox"/> TEST SUITE: Suite1	
Full Name:	Sample.Suite1
Source:	d:\python27\scripts\sample\suite1.txt
Start / End / Elapsed:	20110902 15:00:36.515 / 20110902 15:00:50.108 / 00:00:13.593
Overall Status:	FAIL
Message:	1 critical test, 0 passed, 1 failed 1 test total, 0 passed, 1 failed
<hr/>	
<input type="checkbox"/> TEST CASE: TestCase1	
Full Name:	Sample.Suite1.TestCase1
Start / End / Elapsed:	20110902 15:00:36.530 / 20110902 15:00:50.108 / 00:00:13.578
Status:	FAIL (critical)
Message:	Timed out after 5000.0ms
<input type="checkbox"/> KEYWORD: mykeyword.打开浏览器	
Start / End / Elapsed:	20110902 15:00:36.530 / 20110902 15:00:50.108 / 00:00:13.578
<input type="checkbox"/> KEYWORD: SeleniumLibrary.Open Browser http://zfiles.chaosnet.org/	
Documentation:	Opens a new browser instance to given URL.
Start / End / Elapsed:	20110902 15:00:36.546 / 20110902 15:00:50.108 / 00:00:13.562
15:00:50.108 INFO Opening browser 'firefox' to base url ' http://zfiles.chaosnet.org/ '	
15:00:50.108 INFO	
	

- **设置调试信息级别**

 - Set Log Level | Trace

- **拷屏**

 - Capture Page Screenshot | filename=None

- **将页面内容加入Log**

 - Log Location

 - Log Source | loglevel=INFO

 - Log `message` | `message, level=INFO`

- **Log特定信息 (Built In Library)**

 - Log Variables | `level=INFO`

- **Log变量 (Built In Library)**



- 底色表示成功与失败

Generated
 20110902 15:06:53 GMT +09:00
 13 seconds ago

Sample Test Report

Summary Information

Status: All tests passed
Start Time: 20110902 15:06:42.717
End Time: 20110902 15:06:53.078
Elapsed Time: 00:00:10.361

Test Statistics

Total Statistics	Total	Pass	Fail	Graph
Critical Tests	1	1	0	<div style="width: 100%; height: 10px; background-color: green;"></div>
All Tests	1	1	0	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Tag	Total	Pass	Fail	Graph
No Tags				<div style="width: 0%; height: 10px; background-color: green;"></div>

Statistics by Suite	Total	Pass	Fail	Graph
Sample	1	1	0	<div style="width: 100%; height: 10px; background-color: green;"></div>
Sample.Suite1	1	1	0	<div style="width: 100%; height: 10px; background-color: green;"></div>

Test Details by Suite

Name	Documentation	Metadata / Tags	Crit.	Status	Message	Start / Elapsed
Sample			N/A	PASS	1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed	20110902 15:06:42 00:00:10
Sample.Suite1			N/A	PASS	1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed	20110902 15:06:43 00:00:10
TestCase1			yes	PASS		20110902 15:06:43 00:00:10



练习一

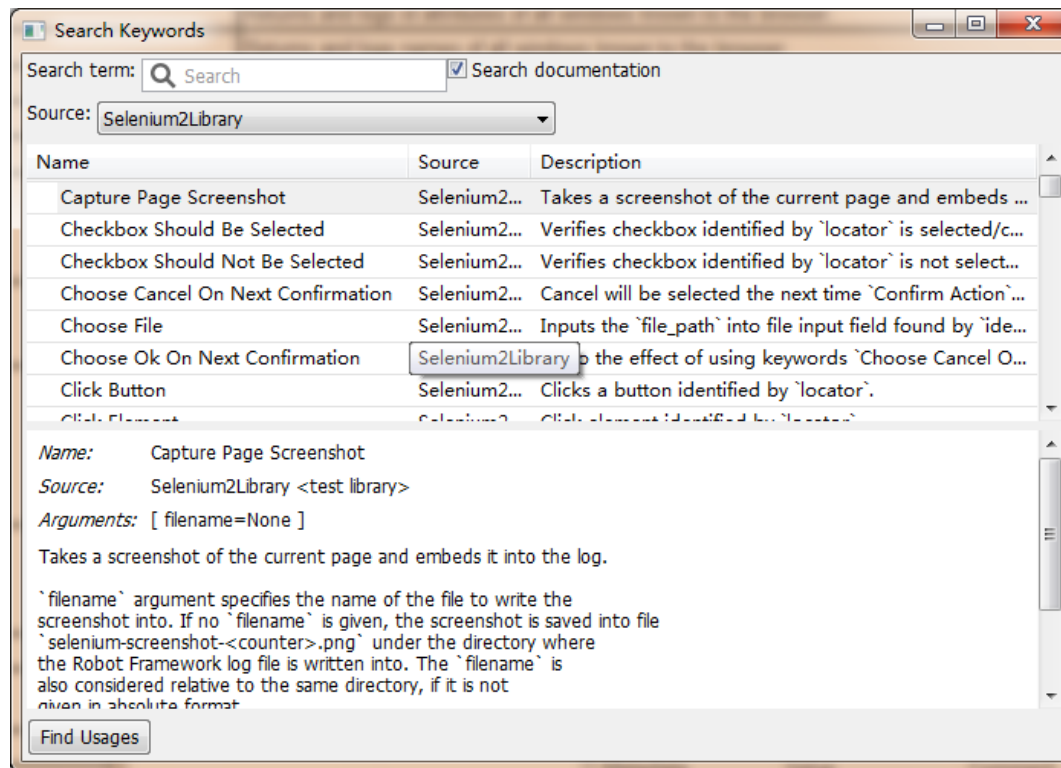
- 建立测试用例
- 查看测试报告



- 基本概念
- 安装
- 创建测试用例
- 执行测试
- 使用Selenium2Library



- Selenium Library将WebDriver与Robot Framework连接起来
- Selenium2Library提供了非常多的内置关键词
- 可以使用F5或以下链接来学习



- Selenium2Lib提供了与浏览器交互的关键词：
 - Open Browser
 - Maximize Browser Window
 - Go To
 - Go Back
 - Reload Page
 - Close Window
 - Close Browser
 - Close All Browsers



- **点击**

- Click Button | locator

- Click Element | locator

- Click Image | locator

- Click Link | locator

- Double Click Element | locator

- **有时在IE中，Click不起作用，可以用如下命令来替代**

- Press Key | <id> | \\13或\\10



定位元素

- Locator
 - 可以id或name来用定位界面元素
 - 也可以使用XPath或Dom, 但是, 必须用XPath=或Dom=来开头
- 最好使用id来定位, 强烈建议强制要求开发为所有需要交互的界面元素设定id, 所以, locator最好以id=, name=, identifier=

Strategy	Example	Description
identifier	Click Element identifier=my_element	Matches by @id or @name attribute
id	Click Element id=my_element	Matches by @id attribute
name	Click Element name=my_element	Matches by @name attribute
xpath	Click Element xpath=//div[@id='my_element']	Matches with arbitrary XPath expression
dom	Click Element dom=document.images[56]	Matches with arbitrary DOM express
link	Click Element link=My Link	Matches anchor elements by their link text
css	Click Element css=div.my_class	Matches by CSS selector
tag	Click Element tag=div	Matches by HTML tag name



- **也可以应用 Assign Id To Element 关键词来为没有ID的元素指定一个ID**



- 有时，需要先进入一个frame或iframe再定位一个元素，这样效率更高，可以使用如下两个关键词
 - 进入一个frame或iframe
 - Select Frame | locator
 - 回到最上层frame或iframe
 - Unselect Frame



- 输入

- Input Password | locator | text

- Input Text | locator | text

- 有时会出现输入之后界面没有的问题，可以用如下命令来触发页面更新

- Press Key | <id> | \\13或\\10



- **按键**

- Press Key | locator | key

- **例子**

- Press Key | text_field | q |

- Press Key | login_button | \\13 | # ASCII code for

- **特殊编码**

- 0: Keys.NULL,

- 8: Keys.BACK_SPACE,

- 9: Keys.TAB,

- 10: Keys.RETURN,

- 13: Keys.ENTER,

- 24: Keys.CANCEL,

- 27: Keys.ESCAPE,

- 32: Keys.SPACE,

- 42: Keys.MULTIPLY,

- 43: Keys.ADD,

- 44: Keys.SEPARATOR,

- 45: Keys.SUBTRACT,

- 56: Keys.DECIMAL,

- 57: Keys.DIVIDE,

- 59: Keys.SEMICOLON,

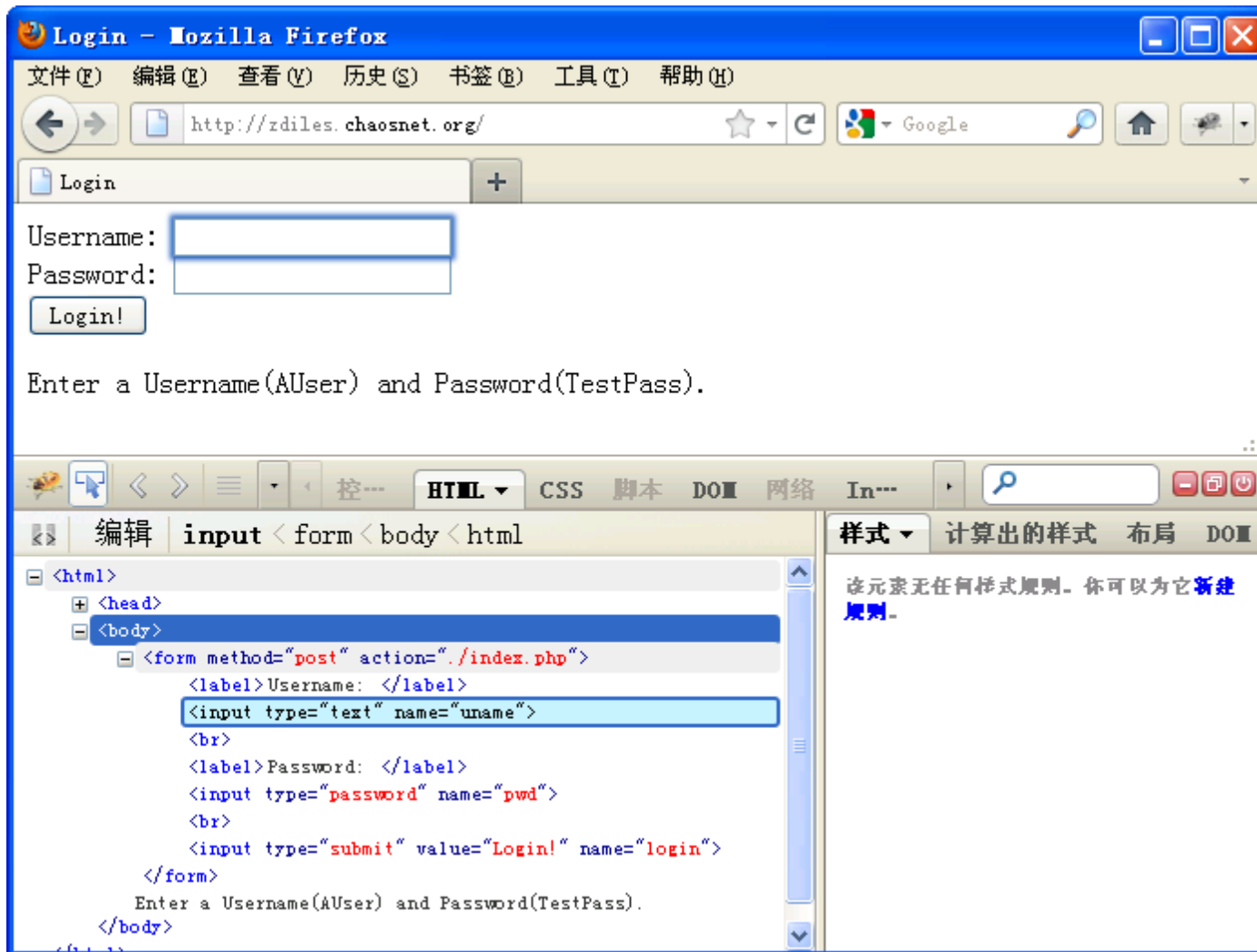
- 61: Keys.EQUALS,

- 127: Keys.DELETE



定位元素

- 可以使用Firefox Firebug或IE8来辅助定位



Firefox browser window showing a login page at `http://zdiles.chaosnet.org/`. The page contains a form with fields for Username and Password, and a Login! button. Below the form, it says "Enter a Username(AUser) and Password(TestPass)."

The Firebug developer tool is open, showing the HTML structure. The selected element is the text input field for the username, with its HTML code highlighted:

```
<input type="text" name="uname">
```

The Firebug interface also shows a message: "该元素无任何样式规则。你可以为它新建规则。"



- 创建更多关键词

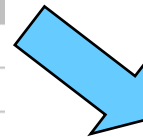
```
1  *** Settings ***
2  Library                Selenium2Library
3
4  *** Keywords ***
5  打开浏览器
6      Open Browser      http://localhost:7272/html/      ie
7
8  输入用户名
9      Input Text        username_field      demo
10
11  输入密码
12      INPUT PASSWORD    password_field      mode
13
14  提交
15      Click Button      login_button
16
17  检查
18      Page Should Contain      Login succeeded
19
20  退出
21      Close All Browsers
22
23
```



- 扩展测试用例



test1	
Settings >>	
1	打开浏览器
2	输入用户名
3	输入密码
4	提交
5	检查
6	退出
7	
8	
9	
10	
11	



Welcome Page

Login succeeded. Now you can [logout](#).

- **验证页面是否包含相应结果**

- Page

- Page Should (Not) Contain Button
- Page Should (Not) Contain Checkbox
- Page Should (Not) Contain Element
- Page Should (Not) Contain Image
- Page Should (Not) Contain Link
- Page Should (Not) Contain List
- Page Should (Not) Contain Radio Button
- Page Should (Not) Contain Textfield

- Location & Title

- Location Should Be
- Location Should Contain
- Title Should Be



- **验证页面是否包含相应结果**
 - Text
 - Current Frame Contains
 - Page Should Contain
 - Page Should Not Contain
 - Textfield Value Should Be
 - Element
 - Element Should Be Disabled
 - Element Should Be Enabled
 - Element Should Be Visible
 - Element Should Contain
 - Element Should Not Be Visible
 - Element Text Should Be



- **验证页面是否包含相应结果**
 - List
 - List Selection Should Be
 - List Should Have No Selections
 - Checkbox
 - Checkbox Should Be Selected
 - Checkbox Should Not Be Selected
 - Radio Button
 - Radio Button Should Be Set To
 - Radio Button Should Not Be Selected



- **验证页面是否包含相应结果**
 - Table
 - Table Cell Should Contain
 - Table Column Should Contain
 - Table Footer Should Contain
 - Table Header Should Contain
 - Table Row Should Contain
 - Table Should Contain
 - Textfield Should Contain
 - Xpath
 - Xpath Should Match X Times

- **默认情况下， Selenium2会自动等待页面加载完成**
- **对于页面AJAX可以使用如下等待页面的关键词：**
 - Wait For Condition | condition | timeout=None | error=None
 - Wait Until Page Contains | text | timeout=None | error=None
 - Wait Until Page Contains Element | locator | timeout
- **一种调试时可以使用， 但是不应该留下来的**
 - Sleep x
 - Sleep**关键词是**Robot Framework Built In Library**提供的，**
<http://robotframework.googlecode.com/hg/doc/libraries/BuiltIn.html?r=2.6.1>

- Selenium2Lib提供了和其他界面元素交互的关键词：
 - Element
 - Focus 聚焦
 - Simulate 模拟时间触发
 - Open Context Menu 打开
 - List
 - Select All From List
 - Select From List
 - Unselect From List
 - Checkbox
 - Select Checkbox
 - Unselect Checkbox
 - Radio Button
 - Select Radio Button
 - Form
 - Submit Form
 - Choose File (For File Upload)



- 更新测试用例



- 基本概念
- 安装
- 创建测试用例
- 执行测试
- 使用Selenium Library
- 数据驱动的测试与测试用例的架构



敏捷测试 - 写好自动化测试用例的原则

- 要做到流程和数据的分离

- 注意：界面元素id不算数据，不应分离

The image shows a Selenium IDE test case named 'test1'. The test case table has the following steps:

Step	Action	Target	Value
1	打开浏览器		
2	输入用户名	username_field	demo
3	输入密码		
4	提交		
5	检查		
6	退出		

The 'demo' value in step 2 is circled in red. In the 'Arguments' field of the 'Input Text' step, the value is `${username}`, which is also circled in red. A blue arrow points to the 'Arguments' field. Below the main screenshot, a smaller version of the test case table is shown, with the 'demo' value in the 'Value' column circled in red.



- 增加一个新的测试用例

Test Case 1		
Settings >>		
1	打开浏览器	http://localhost:7272/html/
2	输入用户名	demo
3	输入密码	mode
4	提交	
5	检查	Login succeeded
6	退出	

Test Case 2		
Settings >>		
1	打开浏览器	http://localhost:7272/html/
2	输入用户名	demo
3	输入密码	error
4	提交	
5	检查	Login failed
6	退出	

这样好吗？

敏捷测试 – 写好自动化测试用例的原则

- 要做到流程和数据的分离
- 通过不断抽象，消除冗余

Test Case 2

Settings >>

1	打开浏览器	http://localhost:7272/html/		
2	输入用户名	demo		
3	输入密码	error		
4	提交			
5	检查	Login failed		
6	退出			
7				
8				
9				
10				
11				

New User Keyword

Name

Arguments

Give a name and arguments for the new user keyword.
Specify the arguments separated with a pipe character like '\${arg1} | \${arg2}'.
Default values are given using equal sign and the last argument can be a list variable.
Example: '\${arg1} | \${arg2}=default value | @-rest}'.



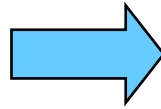
敏捷测试 - 写好自动化测试用例的原则

- 要做到流程和数据的分离
- 通过不断抽象，消除冗余

Test Case 1

Settings >>

1	打开浏览器	http://localhost:7272/html/
2	输入用户名	demo
3	输入密码	mode
4	提交	
5	检查	Login succeeded
6	退出	



登录测试

Settings <<

Documentation

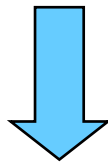
Arguments: \${url} | \${username} | \${password} | \${text}

Teardown

Timeout

Return Value

1	打开浏览器	\${url}
2	输入用户名	\${username}
3	输入密码	\${password}
4	提交	
5	检查	\${text}
6	退出	



Test Case 1

Settings >>

1	登录测试	http://localhost:7272/html/	demo	mode	Login succeeded
---	------	-----------------------------	------	------	-----------------



敏捷测试 - 写好自动化测试用例的原则

- 要做到流程和数据的分离
- 通过不断抽象，消除冗余
- 测试用例应尽量简单易读，避免复杂逻辑

登录测试

Settings <<

Documentation

Arguments: \${url} | \${username} | \${password} | \${text}

Teardown

Timeout

Return Value

1	打开浏览器	\${url}
2	输入用户名	\${username}
3	输入密码	\${password}
4	提交	
5	检查	\${text}
6	退出	

Test Case 1

Settings >>

1	登录测试	http://localhost:7272/html/	demo	mode	Login succeeded
---	------	-----------------------------	------	------	-----------------

Test Case 2

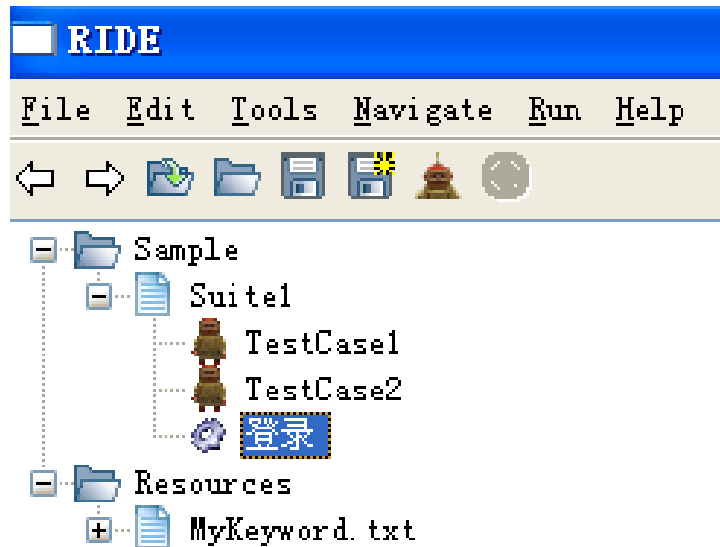
Settings >>

1	登录测试	http://localhost:7272/html/	demo	error	Login failed
---	------	-----------------------------	------	-------	--------------



敏捷测试 – 写好自动化测试用例的原则

- 要做到流程和数据分离
- 通过不断抽象，消除冗余
- 测试用例应尽量简单易读，避免复杂逻辑



问题：“登录” Keyword应该在这里吗？

敏捷测试 – 写好自动化测试用例的原则

- 要做到流程和数据的分离
- 通过不断抽象，消除冗余
- 测试用例应尽量简单易读，避免复杂逻辑
- 建立测试用例分层架构，并坚守

The image displays the RIDE (Robot Framework IDE) interface, illustrating a structured test suite architecture. On the left, a file explorer shows a hierarchy: 'Sample' (Suite1, TestCase1, TestCase2), 'Resources' (Flow.txt, 登录), and 'MyKeyword.txt' (打开浏览器, 输入用户名, 输入密码, 提交). Red boxes highlight these three levels, with arrows pointing to the right. On the right, two windows show the code for 'Suite1' and 'Flow'. The 'Suite1' window shows a 'Resource' entry for 'Flow.txt', and the 'Flow' window shows a 'Resource' entry for 'MyKeyword.txt'. Both 'Resource' entries are circled in red, demonstrating the separation of flow and data.

RIDE
File Edit Tools Navigate

Sample
Suite1
TestCase1
TestCase2

Resources
Flow.txt
登录

MyKeyword.txt
打开浏览器
输入用户名
输入密码
提交

Suite1
Source D:\Python27\Scripts\sample\Suite1.txt
Settings >>

Import	Name / Path	Arguments	Comment
Resource	Flow.txt		

Flow
Source D:\Python27\Scripts\sample\Flow.txt
Settings <<
Documentation

Import	Name / Path	Arguments	Comment
Resource	MyKeyword.txt		

AGILEAN

测试用例的层次架构 (实例)

测试用例 =
测试流程模板 +
测试数据

测试流程模板 =
测试流程构件的排列

测试流程构件 =
测试流程步骤的排列

测试流程步骤 =
界面交互细节

新保

Settings >>

1	新保流程	110000	110100
2	新保流程	310000	310100

新保流程

Settings >>

1	打开页面	\${province-code}	\${city-code}
2	基本信息页-北京	\${mobile}	\${email}
3	NewCarBasicInfo	\${province-code}	\${city-code}

基本信息页-北京

Settings >>

1	新车	
2	手机	\${mobile}
3	电邮	\${email}
4	生效日期	\${valid-date}
5	第二页	

手机

Settings >>

1	INPUT TEXT	pre_insurantMobilePhone	\${arg1}
---	------------	-------------------------	----------

Call Selenium Library

- 分层测试用例架构



- **基本概念**
- **安装**
- **创建测试用例**
- **执行测试**
- **使用Selenium Library**
- **数据驱动的测试与测试用例的架构**
- **使用变量**

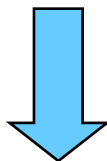


- 使用变量来实现测试目标可变

1	登录测试	http://localhost:7272/html/	demo	mode	Login succeeded
2					
3					
4					

Context menu options:

- Create Keyword
- Extract Keyword
- Extract Variable



Scalar Variable

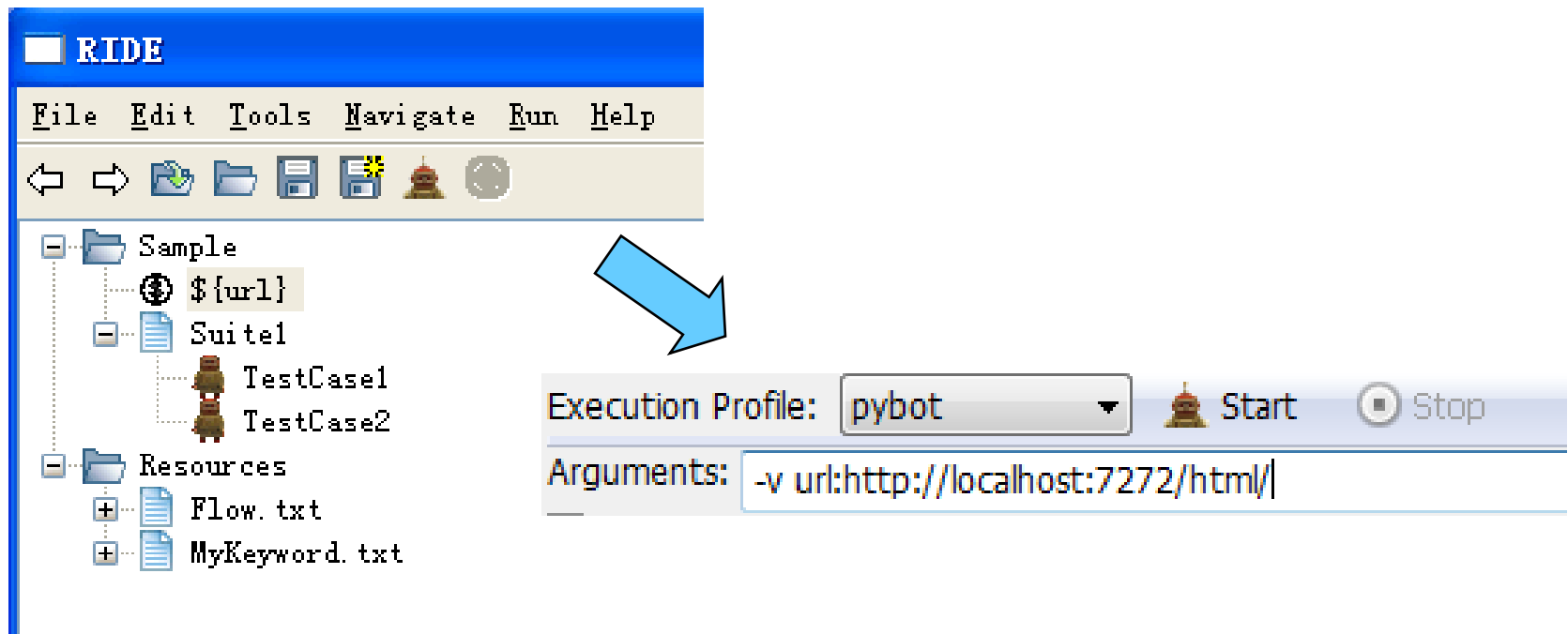
Name:

Value:

Comment:

Give name and value of the variable.

- 使用变量来实现测试目标可变
 - 运行时可以指定变量取值切换不同环境



- 可以使用变量来获得界面返回值，用于验证

```
    | ${updated_password}=      | GET TEXT      | pwd      |
```

- 需要注意关键词里面的返回值变量只在关键词之内有效，
- 可以通过如下关键词来提升变量的可见范围，但是最好只提升到Test级别，因为测试用例最好是相互独立的

Set Global Variable	<i>name, *values</i>
Set Suite Variable	<i>name, *values</i>
Set Test Variable	<i>name, *values</i>



- **相关关键词有：**

- Page

- Get Title
- Get Location
- Get Source
- Get Matching Xpath Count

- Element

- Get Element Attribute
- Get Value
- Get Text (Added By Adam Wu)

- Link

- Get All Links



- **相关关键词有：**

- Table

- Get Table Cell

- List

- Get List Items
 - Get Selected List Label
 - Get Selected List Labels
 - Get Selected List Value
 - Get Selected List Values



- 基本概念
- 安装
- 创建测试用例
- 执行测试
- 使用Selenium Library
- 数据驱动的测试与测试用例的架构
- 使用变量
- 设置Timeout



设置Timeout

- Suite和Test都可以设定Timeout，这限定了测试用例的最长执行时间，因此，需要预留合适的时间，否则，可能导致测试不稳定

The screenshot shows the configuration interface for a test suite named 'Suite1'. The source file is 'D:\Python27\Scripts\sample\Suite1.txt'. The 'Test Timeout' field is highlighted with a red box and contains the value '1 minute'. Other fields include Suite Setup, Suite Teardown, Test Setup, Test Teardown, Test Template, Force Tags, and Default Tags.

The screenshot shows the configuration interface for a test case named 'TestCase1'. The 'Timeout' field is highlighted with a red box and contains the value '1 minute'. Other fields include Setup, Teardown, Template, and Tags (with a 'test1' tag and an '<Add New>' button).



设置Timeout（需要被更新）

- Timeout**也可以通过如下Keyword来设置**（
 - Get Selenium Timeout
 - Set Selenium Timeout
- Selenium**执行速度**
 - Get Selenium Speed
 - Set Selenium Speed



- 要做到流程和数据分离
- 通过不断抽象，消除冗余
- 测试用例应尽量简单易读，避免复杂逻辑
- 建立测试用例分层架构，并坚守
- 自动化测试用例必须非常健壮，避免误报



- 选择式输入与日期处理



选择式输入与日期处理

- 在界面当中，有许多信息使用选择方式来输入，如年龄，日期，密码等等；
- 如下面界面，当出发点击事件，就会弹出一个动态的div层，就是看到的“下拉框”。
- 可以用如下方式直接赋值来进行选择：

```
-input text //select[@id='registeredAgePeriod'] 03
```



- 对于有些日期输入框，也可以直接调用JavaScript
 - 注意这里window. 一定要写不能省略

1 Execute JavaScript

```
window.document.getElementById("app_birthday").value=${app_birthday}
```

投保人资料 导入已有客户>>

* 姓名：

* 出生日期：

* 证件类型：

* 职业大类：

* 职业中类：

* 职业小类：

* 职业名称：

* 是否有社保：

详细资料>>

* 性别： 男 女

* 年收入： 万元

证件号码：

- 二 三 四 五 六 日						
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

- 很多时候输入的日期值，需要根据当前日期计算
- 可以利用BuildIn Library里面的Get Time来解决

Get Time

format=timestamp, time=NOW

<code>\$(time) =</code>	Get Time			
<code>\$(secs) =</code>	Get Time	epoch		
<code>\$(year) =</code>	Get Time	return year		
<code>\$(yyyy)</code>	<code>\$(mm)</code>	<code>\$(dd) =</code>	Get Time	year,month,day
<code>@{time} =</code>	Get Time	year month day hour min sec		
<code>\$(y)</code>	<code>\$(s) =</code>	Get Time	seconds and year	

<code>\$(time) =</code>	Get Time		1177654467	
<code>\$(secs) =</code>	Get Time	sec	2007-04-27 09:14:27	
<code>\$(year) =</code>	Get Time	year	NOW	# The time of execution
<code>\$(day) =</code>	Get Time	day	NOW - 1d	# 1 day subtraced from NOW
<code>@{time} =</code>	Get Time	hour min sec	NOW + 1h 2min 3s	# 1h 2min 3s added to NOW

自动化测试工具获取的当前日期格式与需要校验的日期格式不一致怎么办？

现象：网销会员专区的的保险起期的提示日期在自动化测试中需要去校验，其格式为“YYYY-M-DD”，但是获取的系统日期格式为“YYYY-MM-DD”，怎么样去校验两个日期相等呢？

例如：网销会员专区日期显示为“2012-5-19”
但是获取的当前系统时间“2012-05-19”

解决方法：日期格式差别在于月份，用 get time 关键字将年、月、日分别获取，将获取的月份格式调整为“integer”类型。

Example:expecting the current time is “2012-05-18”

<code>\$(year)</code>	get time	year
<code>\$(month)</code>	get time	month
<code>\$(day)</code>	get time	day
<code>\$(month1)</code>	Convert to integer	<code>\$(month)</code>
<code>\$(date)</code>	Set variable	<code>\$(year)-\$(month1)-\$(day)</code>

=>

```
$(year)='2012'  
$(month)="05"  
$(day)="18"  
$(month1)='5'  
$(date)='2012-5-18'
```



- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框



1. 什么是 AJAX? (Async JavaScript and XML) 非同步的 JavaScript 与 XML 技术

AJAX 技术允许页面以非同步方式和服务器后台通讯, 获取数据; 这样页面可以显得加载很快, 用户可以先进行一些操作, 而其他需要的数据可以在用户操作过程中, 陆续加载过来, 而不用页面等待所有数据加载完成; 另外, 在用户和页面交互过程中, 页面可以按需动态获取数据而不需重新刷新页面;

2. 如何判断页面使用了 AJAX 技术

有时你会看到等待图标 (小圆圈转), 或者你们注意到有些信息要延时显示, 或者自动换验收测试报错

3. AJAX 给自动化测试带来的困扰

因为文档 ready 状态不代表所有 AJAX 已返回, 也可能造成 Selenium 过早地进行页面操作, 而造成执行错误 (选择一个列表框, 如果其内容是 AJAX 加载过来的, 则可能 Selenium 选择时, AJAX 还没完成, 从而造成脚本失败), 这是测试同学往往会使用 Sleep, 这会短期解决问题, 但会造成更慢的测试和不稳定的测试

4. 优雅地等待 AJAX 返回的方式

大多数 AJAX 框架里面都有一个记录当前活跃 AJAX 链接数的变量, 我们可以优雅地等待这个变量为 0 时, 再进行后一步的操作, 这样就可以解决 AJAX 动态等待的问题了, 以 JQuery 1.3.2 为例, 在 RF 中, 可以这样写

```
Wait For Condition | return window.jQuery.active == 0;
```

这里需要注意的是, 每一种 AJAX 框架及其不同版本, 这个计数器都是不同的, 因此, 使用之前要请对应系统的开发人员确认一下



等待AJAX返回结果

- 定位AJAX会动态更新的页面内容
- 定义一个等待动态加载完成的关键词

报价非空			
Settings >>			
1	Element Should Contain	jquery=.recommend > div	¥

中国平安 PINGAN 平安车险

1 基本信息 ➔ 2 商业险报价 ➔ 3 交强险报价 ➔ 4 填写投保信息 ➔ 5 确认投保信息 ➔ 6 支付

每款套餐都省了这么多，还提供平安最优质的服务，太给力了！

★ 我们的优势

- 推荐您选择“性价比”套餐。
- 您也可根据用车情况。

自由定制
¥您来决定

基本保障

尊享保障



等待AJAX返回结果

- 等待界面返回, 可以利用Built In Library里面的Wait Until Keyword Succeeds关键词

1	Wait Until Keyword Succeeds	2 min	15 sec	报价非空
---	-----------------------------	-------	--------	------

Wait Until Keyword Succeeds

*timeout, retry_interval, name, *args*

中国平安 PINGAN 平安车险

1 基本信息 → 2 商业险报价 → 3 交强险报价 → 4 填写投保信息 → 5 确认投保信息 → 6 支付

每款套餐都省了这么多, 还提供平安最优质的服务, 太给力了!

★ 我们的优势

- 推荐您选择“性价比高”套餐。
- 您也可根据用车情况。

自由定制

¥您来决定

基本保障

尊享计划



- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框



- 如果测试用例打开了多个窗口或多个Frame，可以用如下关键词来进行切换：

Switch Browser	<i>index_or_alias</i>
Select Frame	<i>locator</i>
Select Window	<i>locator=main</i>
Unselect Frame	

- 如果打开新窗口的时间不确定，可以结合Wait Until Keyword Succeeds关键词，以确保切换窗口成功



- 另外，可以使用如下关键词来获取所有当前打开窗口的信息，方便调试
 - Get Window Identifiers
 - Get Window Names
 - Get Window Titles



切换窗口（当前一个窗口自动关闭时）

- 前一个窗口用 `window.open()` 打开一个新窗口，之后马上用 `window.close()` 关闭自身
 - 这是可以在触发 `window.open` 的动作之前，执行如下动作：
 - `Execute JavaScript | window.close = function() {};`



- JQuery支持
- Flex支持
- Firefox Profiles支持
- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框



- 在Selenium2/IE中，模式对话框就像一个普通窗口一样，只需用Select Window结合Wait Until Keyword Succeeds切换过去即可，在模式对话框关闭之后再使用Select Window结合Wait Until Keyword Succeeds切换回主窗口即可



- Selenium会修改页面JS代码，所以不会有实际的弹出窗口
- 可以调用如下关键词判断是否执行了弹出动作以及内容是否符合预期

Click Button	Send	# Shows a confirmation dialog
<code>\${message}=</code>	Confirm Action	# Chooses Ok
Should Be Equal	<code>\${message}</code>	Are your sure?
Choose Cancel On Next Confirmation		
Click Button	Send	# Shows a confirmation dialog
Confirm Action		# Chooses Cancel

- 其实这里Confirm Action不会进行点击动作，而只是会去获取需要确认的内容，但是如果不调用Confirm Action，后面就会报错了；
- 而Choose Cancel on Next Confirmation就是规定下一次Confirm选择Cancel



- Alert的处理

Alert Should Be Present	text=
Get Alert Message	

- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框
- 模拟Windows动作



- Robot Framework**还有一个**AutoItLibrary，**提供了一些操作Windows界面的关键词，安装方式如下：**

- **安装**Python Windows Extension(pywin32-216.win32-py2.7.exe) (**需要本机管理员权限**)
- **之后，解压**AutoItLibrary-1.1.zip**到任意目录，然后在目录下执行**

- `setup.py install`

- **注意：**

- **在Win 7或Windows Server上**
- **以你的有管理员权限的用户登录**
 - » **而不要直接用Administrator账号登录**
- **要以管理员身份打开命令行窗口**
- **再执行**`setup.py`



- **如果用Administrator执行了setup.py install, 那么使用其他用户使用AutoIt时, 会报这个错误**
 - >>> Don't think we need to unregister the old one...
 - Traceback (most recent call last):
 - File "C:\Python27\Lib\site-packages\pythonwin\pywin\framework\scriptutils.py", line 325, in RunScript exec codeObject in __main__.__dict__
 - File "C:\pathon2.7.1\AutoItLibrary-1.1\setup.py", line 63, in <module> shutil.copyfile("3rdPartyTools/AutoIt/AutoItX3.dll", instFile)
 - File "C:\Python27\lib\shutil.py", line 81, in copyfile with open(src, 'rb') as fsrc:
 - IOError: [Errno 2] No such file or directory: '3rdPartyTools/AutoIt/AutoItX3.dll'
- **这时需要删除AutoIt Library目录, 同时用regedt32情况注册表里面所有和AutoIt项目的信息**



- **使用时需要Import AutoItLibrary**

- **关键词列表请见**

- <http://robotframework-autoitlibrary.googlecode.com/svn/tags/robotframework-AutoItLibrary-1.0/doc/AutoItLibrary.html>



- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框
- 模拟Windows动作
- String和Collection支持



- Robot Framework**内置提供了**String Library

- Fetch From Left
- Fetch From Right
- Generate Random String
- Get Line
- Get Line Count
- Get Lines Containing String
- Get Lines Matching Pattern
- Get Lines Matching Regexp
- Get Substring

- Replace String
- Replace String Using
- Should Be Lowercase
- Should Be String
- Should Be Titlecase
- Should Be Uppercase
- Should Not Be String
- Split String
- Split String From R
- Split To Lines



- Append To List
- Combine Lists
- Convert To List
- Copy Dictionary
- Copy List
- Count Values In List
- Create Dictionary
- Dictionaries Should Be Equal
- Dictionary Should Contain Key
- Dictionary Should Contain Sub Dictionary
- Dictionary Should Contain Value
- Dictionary Should Not Contain Key
- Dictionary Should Not Contain Value
- Get Dictionary Items
- Get Dictionary Keys
- Get Dictionary Values
- Get From Dictionary
- Get From List
- Get Index From List
- Get Slice From List
- Insert Into List
- Keep In Dictionary
- List Should Contain Sub List
- List Should Contain Value
- List Should Not Contain Duplicates
- List Should Not Contain Value
- Lists Should Be Equal
- Log Dictionary
- Log List
- Remove From Dictionary
- Remove From List
- Remove Values From List
- Reverse List
- Set List Value
- Set To Dictionary
- Sort List



- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框
- 模拟Windows动作
- String和Collection支持
- Cookie支持



- Selenium2Lib提供了如下关键词来操作Cookie:
 - Delete All Cookies
 - Delete Cookie
 - Get Cookie Value
 - Get Cookies



- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框
- 模拟Windows动作
- String和Collection支持
- Cookie支持
- 鼠标动作



- Selenium2Lib提供了如下关键词来操作鼠标：
 - Get Vertical Position
 - Get Horizontal Position
 - Mouse Down
 - Mouse Down On Image
 - Mouse Down On Link
 - Mouse Out
 - Mouse Over
 - Mouse Up



- 日期处理
- 等待AJAX返回结果
- 切换窗口
- Alert处理和模式对话框
- 模拟Windows动作
- String和Collection支持
- Cookie支持
- 鼠标动作
- 证书警告



- 设置这个IE选项，就可以点击忽略证书警告了

允许活动内容在我的计算机上的文件中运行*

- 如果需要，可以改变这个注册表值

To turn the warning off, change the followin registry key to 0.

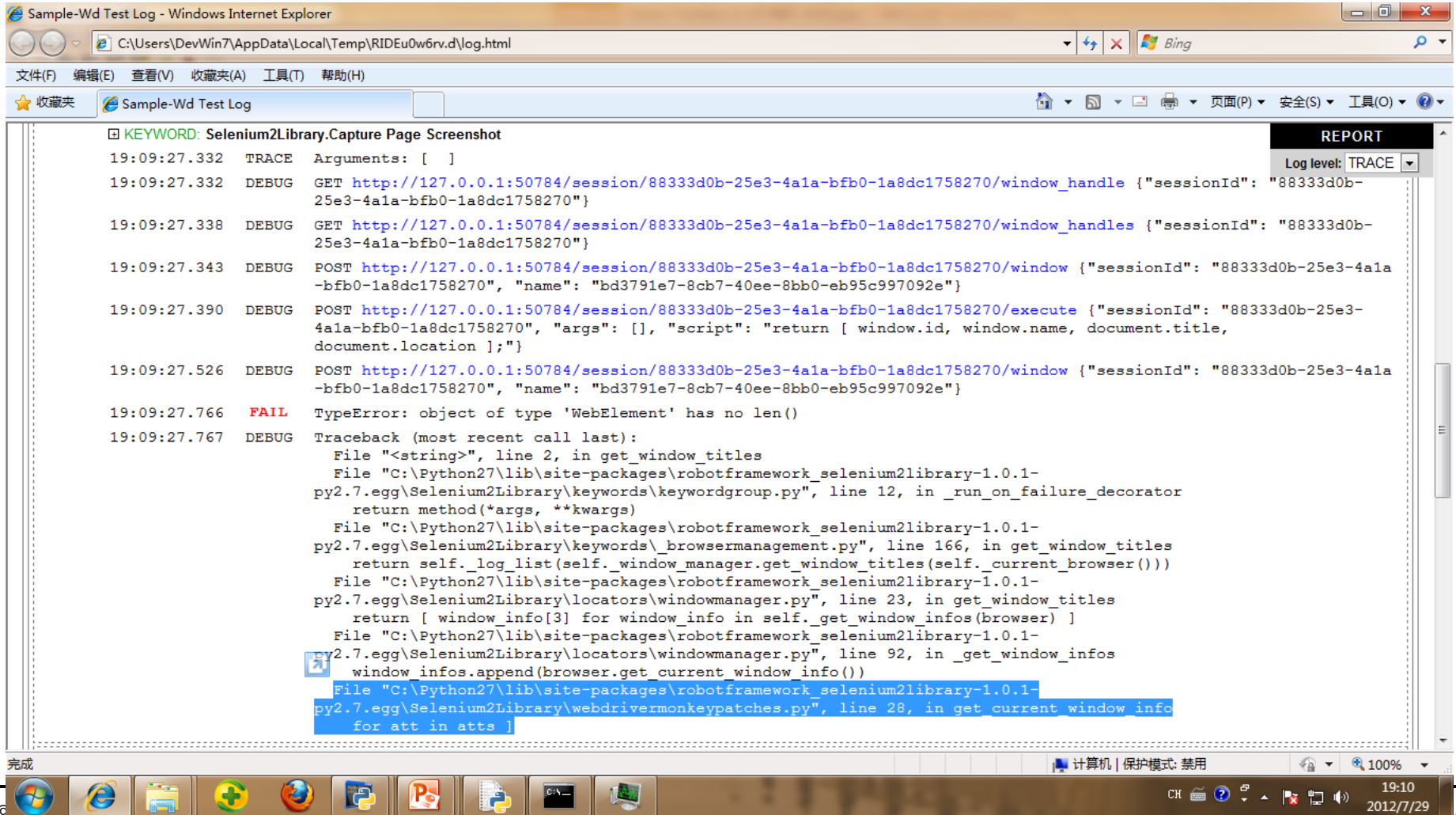
HKEY_USERS\<<SID>\Software\Microsoft\Windows\CurrentVersion\Internet
Settings:WarnonBadCertRecving

- 如何Debug脚本？



如何Debug脚本？

- Set Log Level | Trace
- 查看Log以便定位到的问题



Sample-Wd Test Log - Windows Internet Explorer

C:\Users\DevWin7\AppData\Local\Temp\RIDEu0w6rv.d\log.html

文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

收藏夹 Sample-Wd Test Log

KEYWORD: Selenium2Library.Capture Page Screenshot

19:09:27.332 TRACE Arguments: []

19:09:27.332 DEBUG GET http://127.0.0.1:50784/session/88333d0b-25e3-4a1a-bfb0-1a8dc1758270/window_handle {"sessionId": "88333d0b-25e3-4a1a-bfb0-1a8dc1758270"}

19:09:27.338 DEBUG GET http://127.0.0.1:50784/session/88333d0b-25e3-4a1a-bfb0-1a8dc1758270/window_handles {"sessionId": "88333d0b-25e3-4a1a-bfb0-1a8dc1758270"}

19:09:27.343 DEBUG POST http://127.0.0.1:50784/session/88333d0b-25e3-4a1a-bfb0-1a8dc1758270/window {"sessionId": "88333d0b-25e3-4a1a-bfb0-1a8dc1758270", "name": "bd3791e7-8cb7-40ee-8bb0-eb95c997092e"}

19:09:27.390 DEBUG POST http://127.0.0.1:50784/session/88333d0b-25e3-4a1a-bfb0-1a8dc1758270/execute {"sessionId": "88333d0b-25e3-4a1a-bfb0-1a8dc1758270", "args": [], "script": "return [window.id, window.name, document.title, document.location];"}

19:09:27.526 DEBUG POST http://127.0.0.1:50784/session/88333d0b-25e3-4a1a-bfb0-1a8dc1758270/window {"sessionId": "88333d0b-25e3-4a1a-bfb0-1a8dc1758270", "name": "bd3791e7-8cb7-40ee-8bb0-eb95c997092e"}

19:09:27.766 FAIL TypeError: object of type 'WebElement' has no len()

19:09:27.767 DEBUG Traceback (most recent call last):
File "<string>", line 2, in get_window_titles
File "C:\Python27\lib\site-packages\robotframework_selenium2library-1.0.1-py2.7.egg\Selenium2Library\keywords\keywordgroup.py", line 12, in _run_on_failure_decorator
return method(*args, **kwargs)
File "C:\Python27\lib\site-packages\robotframework_selenium2library-1.0.1-py2.7.egg\Selenium2Library\keywords_browsermanagement.py", line 166, in get_window_titles
return self._log_list(self._window_manager.get_window_titles(self._current_browser()))
File "C:\Python27\lib\site-packages\robotframework_selenium2library-1.0.1-py2.7.egg\Selenium2Library\locators>windowmanager.py", line 23, in get_window_titles
return [window_info[3] for window_info in self._get_window_infos(browser)]
File "C:\Python27\lib\site-packages\robotframework_selenium2library-1.0.1-py2.7.egg\Selenium2Library\locators>windowmanager.py", line 92, in _get_window_infos
window_infos.append(browser.get_current_window_info())
File "C:\Python27\lib\site-packages\robotframework_selenium2library-1.0.1-py2.7.egg\Selenium2Library\webdrivermonkeypatches.py", line 28, in get_current_window_info
for att in atts]

REPORT

Log level: TRACE

完成 计算机 | 保护模式: 禁用 100%

19:10 2012/7/29

- 利用Log
 - Log Source
 - Capture Page Screenshot

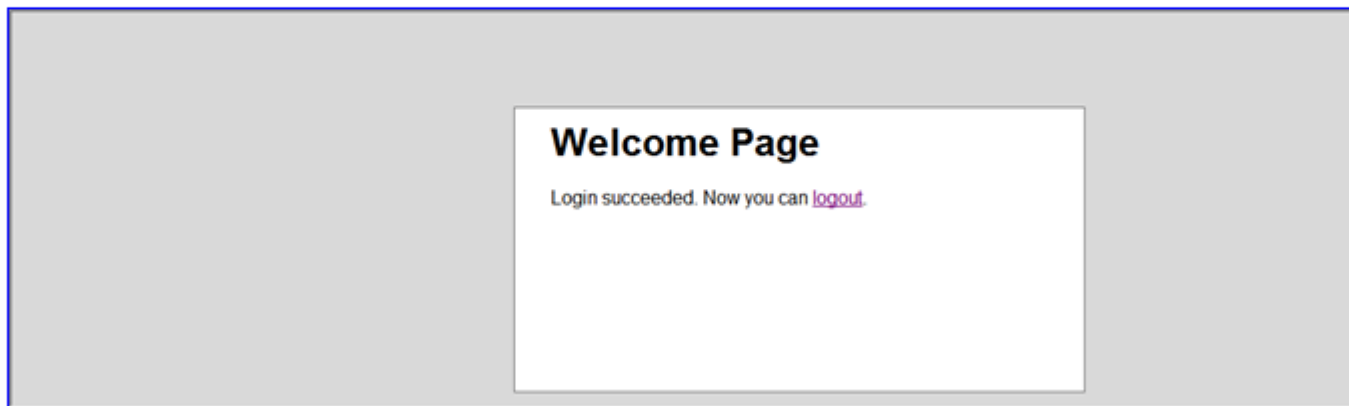
☐ **KEYWORD:** Selenium2Library.Capture Page Screenshot

Documentation: Takes a screenshot of the current page and embeds it into the log.

Start / End / Elapsed: 20120729 19:19:10.999 / 20120729 19:19:11.261 / 00:00:00.262

19:19:11.260

INFO



- 实践表明，pybot在jenkins里和在RIDE下的执行结果可能出现一些不一致，要用好Log发现问题

- **可见性问题**

- Selenium2要求交互的元素必须是可见的，其判定逻辑如下：

- 对于Option或OptGroup，判断包含它的Select是否可见？；
 - 对于Area，判断包含它的Map是否可见？
 - 对于Map，判断MapImage是否可见？
 - hidden input不可见；
 - NoScript不可见；
 - hidden visibility不可见；
 - displayed= 'none' 或任何祖先元素的displayed= 'none' ，则元素不可见；
 - 长宽为零，并且没有长宽不为零的子孙的元素不可见



- 可见性问题

- 需要特别注意这一条，因为IE有一个缺陷使得即便display= “none” 的元素依然可见，

- <http://www.positioniseverything.net/explorer/ienondisappearcontentbugPIE/index.htm>

- display= ‘none’ 或任何祖先元素的display= ‘none’ ，则元素不可见；



- **解决定位问题**
 - Get Vertical Position
 - Get Horizontal Position



如何Debug脚本？

- 测试出现异常时，有时需要手动杀掉IEDriverServer.exe

```
IEDriverServer.exe | DevWin7 | 00 | 1,948 K | Command...
```

- 另外需要杀掉僵尸IE进程

打开浏览器

```
[Arguments]      ${url}
Set Log Level    TRACE
Process Close    IEDriverServer.exe
Process Close    iexplore.exe
Open Browser     ${url}    ie
Get Window Titles
```

- 已经遇到了如下已知缺陷：

Issue [1486](#): session cookie lost when click made to a link that opens a window with window.open()

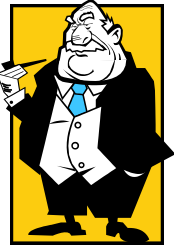
14 people starred this issue and may be notified of changes.



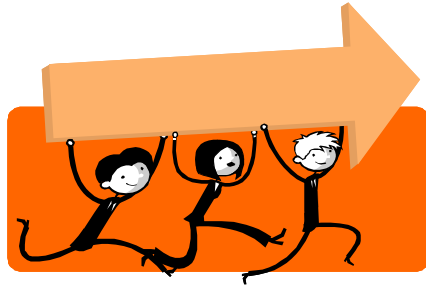
- 敏捷测试过程



Scrum Process Framework



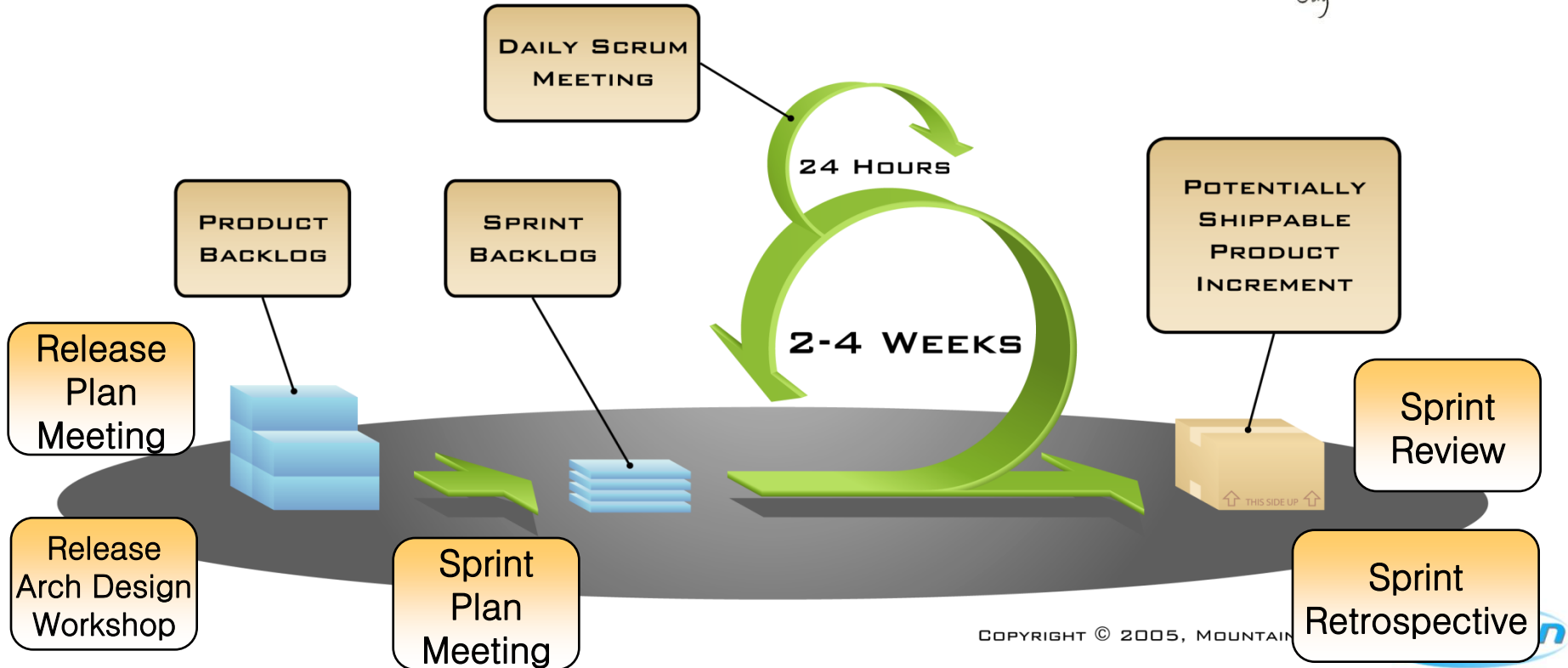
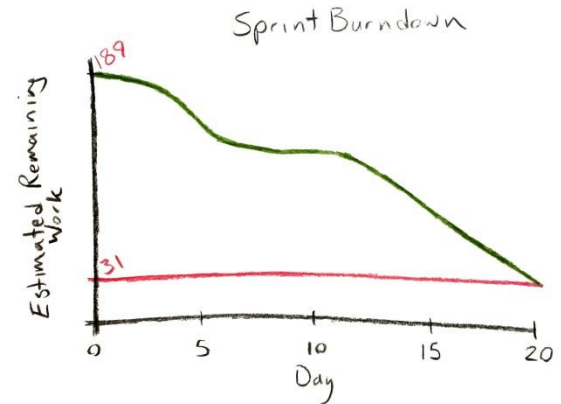
Product Owner



**Developers, Testers, ..
+ Manager/Lead**

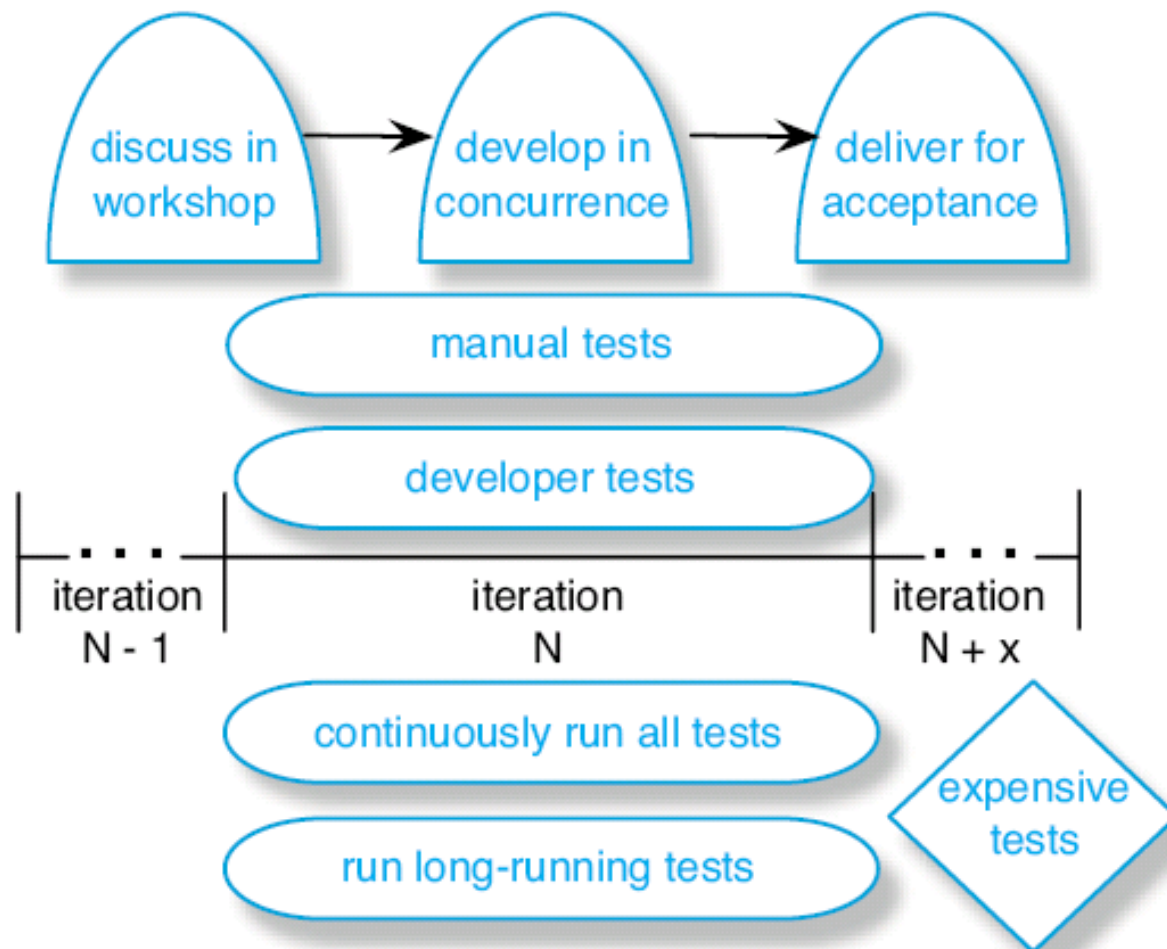


Scrum Master



COPYRIGHT © 2005, MOUNTAIN

迭代过程中的测试 (A-TDD)



- RFCK的使用



一、安装工作

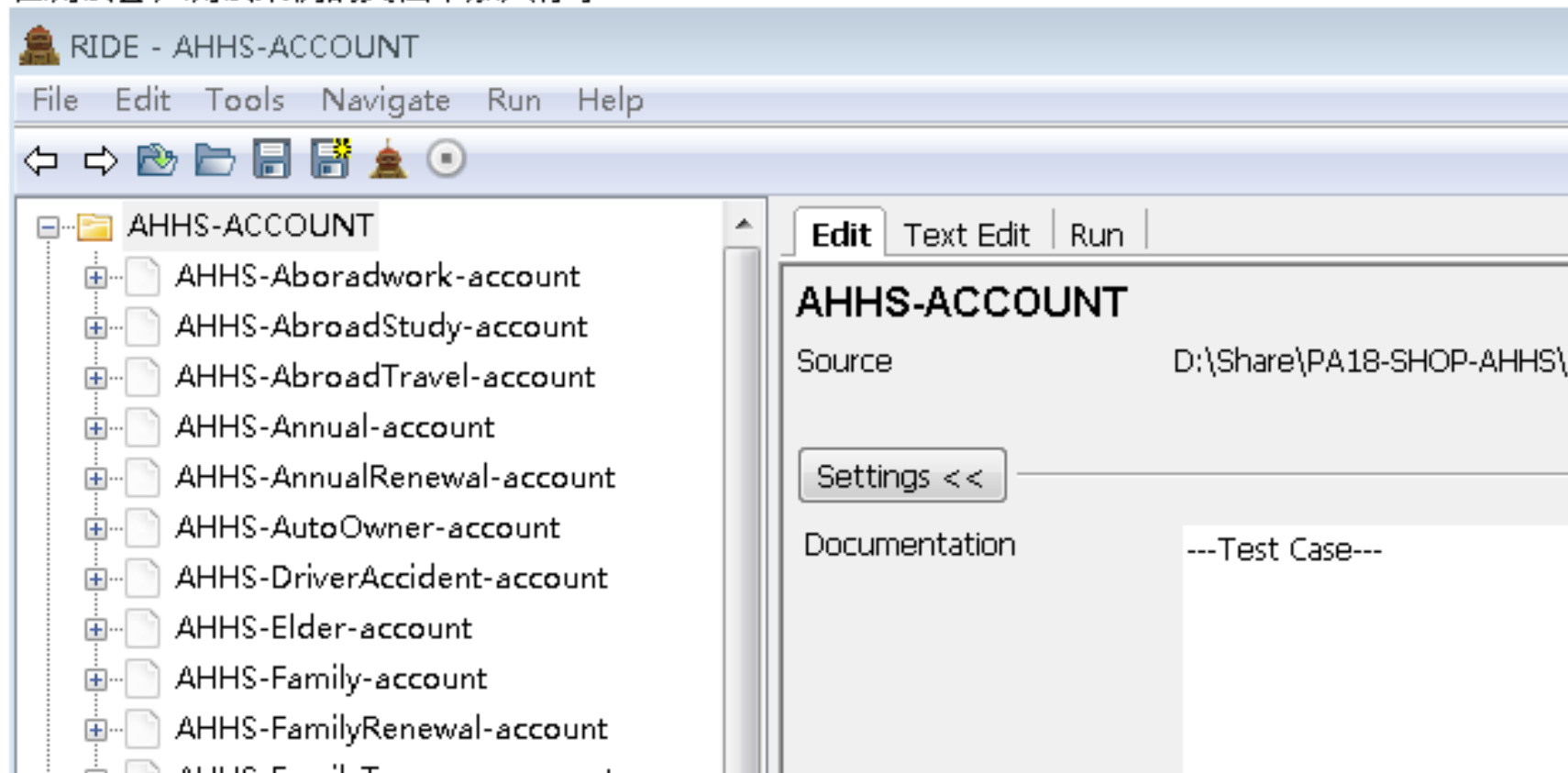
- 安装 scala2.9.2
 - 安装目录 [\\Cnsz120070\Robot Framwork Install\rfcheckstyle\scala-2.9.2.msi](#)
 - 需要允许安装程序将 scala.bat 加入 path
- 将 rfcheckstyle.scala 拷贝到本地
 - 安装目录 [\\Cnsz120070\Robot Framwork Install\rfcheckstyle\rfcheckstyle.scala](#)
- 创建 rfck.bat

```
@echo off
scala d:\rfcheckstyle\rfcheckstyle.scala %1
```



二、脚本准备工作

- 在测试套，测试案例的文档中加入标示 ---Test Case---



- 在流程模板的文档中加入标示 ---Business Flow---
- 在页面对象的文档中加入标示 ---Page Component---
- 在页面交互的文档中加入标示 ---Element Interaction---

三、脚本分析

- 执行 rfck 脚本目录即可
- 也可以将执行结果放到一个文件里面
 - D:\>rfck D:\Share\test\CFS-COUNTER >> d:\rfcheckstyle\cfs.log

四、常见问题

- 文件没有标签：
 - Error:Resource Dependency Violation:script D:\Share\test\CFS-COUNTER\05HTQD.txt has no attribute tag!
- 关键词引用违规：Selenium2Lib 的关键词只能在界面交互层引用
 - Error:Keyword Layer Violation:log source is used in script:D:\Share\PA18-SHOP-AHHS\STG\AHHS-ACCOUNT\compoent_Account.txt of type Page Component
- 资源引用违规：---Test Case--- 引用 ---Business Flow--- 引用 ---Page Component--- 引用---Element Interaction---，其他引用都属违规
 - Error:Resource Dependency Violation:script D:\Share\PA18-SHOP-AHHS\STG\AHHS-ACCOUNT\template_abroadstudy_account.txt of type Business Flow is not allowed to depend on template_Annual_Account.txt of type Business Flow
 - 如果你需要定义新的层次，也可以定义并定义新的检验规则，这个未来再讲；
- 库引用违规：Selenium2Lib 和 AutoItLib 只能在-Element Interaction-层引用
 - Error:Library Import Violation:script D:\Share\PA18-SHOP-AHHS\STG\AHHS-RENEWAL\AHHS-RANEWAL-ACCOUNT__init__.txt of type Test Case is not allowed to depend on lib:Selenium2Library 600
- Sleep 违规：使用 Sleep 关键词会报出违规



- Remote Library
- DB Library



Q&A



- 提供了如下关键词进行错误处理
 - Register Keyword To Run On Failure
 - 例如，可以根据执行结果自动调整Tag



- [Call Method](#) · [Catenate](#) · [Comment](#) · [Convert To Binary](#) · [Convert To Boolean](#) · [Convert To Hex](#) · [Convert To Integer](#) · [Convert To Number](#) · [Convert To Octal](#) · [Convert To String](#) · [Create List](#) · [Evaluate](#) · [Exit For Loop](#) · [Fail](#) · [Fatal Error](#) · [Get Count](#) · [Get Length](#) · [Get Library Instance](#) · [Get Time](#) · [Get Variable Value](#) · [Get Variables](#) · [Import Variables](#) · [Keyword Should Exist](#) · [Length Should Be](#) · [No Operation](#) · [Regexp Escape](#) · [Remove Tags](#) · [Repeat Keyword](#) · [Replace Variables](#) · [Run Keyword](#) · [Run Keyword And Continue On Failure](#) · [Run Keyword And Expect Error](#) · [Run Keyword And Ignore Error](#) · [Run Keyword If](#) · [Run Keyword If All Critical Tests Passed](#) · [Run Keyword If All Tests Passed](#) · [Run Keyword If Any Critical Tests Failed](#) · [Run Keyword If Any Tests Failed](#) · [Run Keyword If Test Failed](#) · [Run Keyword If Test Passed](#) · [Run Keyword If Timeout Occurred](#) · [Run Keyword Unless](#) · [Run Keywords](#) · [Set Global Variable](#) · [Set Library Search Order](#) · [Set Log Level](#) · [Set Suite Variable](#) · [Set Tags](#) · [Set Test Message](#) · [Set Test Variable](#) · [Set Variable](#) · [Set Variable If](#) · [Should Be Empty](#) · [Should Be Equal](#) · [Should Be Equal As Integers](#) · [Should Be Equal As Numbers](#) · [Should Be Equal As Strings](#) · [Should Be True](#) · [Should Contain](#) · [Should Contain X Times](#) · [Should End With](#) · [Should Match](#) · [Should Match Regexp](#) · [Should Not Be Empty](#) · [Should Not Be Equal](#) · [Should Not Be Equal As Integers](#) · [Should Not Be Equal As Numbers](#) · [Should Not Be Equal As Strings](#) · [Should Not Be True](#) · [Should Not Contain](#) · [Should Not End With](#) · [Should Not Match](#) · [Should Not Match Regexp](#) · [Should Not Start With](#) · [Should Start With](#) · [Variable Should Exist](#) · [Variable Should Not Exist](#)

