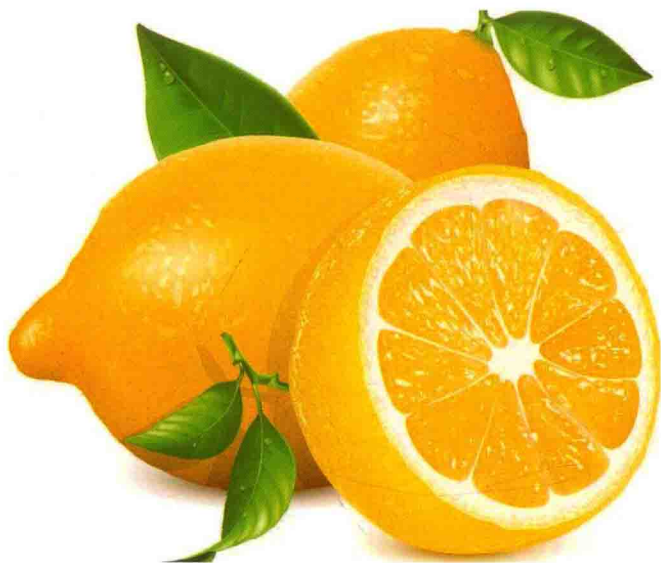


掌握Robot自动化测试开源框架，提高重复功能测试工作效率

- Robot Framework自动化测试框架的基础用法
- 移动端、Web端、接口端等常见自动化测试Lib库的使用
- 自定义测试Lib库及测试案例常用编写技巧
- 自动化框架设计的思想以及其他类型自动化框架简介

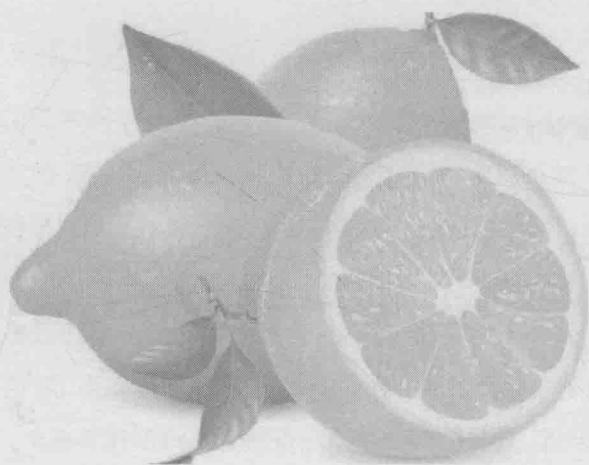


Robot Framework 自动化 测试框架核心指南

张永清 著



清华大学出版社



Robot Framework 自动化 测试框架核心指南

张永清 著

清华大学出版社
北京

内 容 简 介

Robot Framework 是目前众多自动化测试工具或者框架中一个非常流行的开源框架，致力于解决重复功能测试劳动所带来的高额成本，将自动化测试大众化、简单化、通俗化，让更多没有编程基础的人也能成功地完成自动化测试。

本书分 8 章，内容包括 Robot Framework 自动化测试框架的基础用法，框架的常用基础测试 Lib 库的使用（包含移动端、Web 端、接口端等常见自动化测试类型），自定义测试 Lib 库的编写，编写自动化测试案例常用的一些技巧，最后分享一下自动化框架设计的思想以及其他类型自动化框架简介。

本书适合 Robot Framework 初学者、软件测试工程师、软件测试经理阅读，也适合作为高等院校和培训学校相关专业课的配套参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

Robot Framework 自动化测试框架核心指南 / 张永清著. —北京：清华大学出版社，2019

ISBN 978-7-302-52392-5

I. ①R… II. ①张… III. ①软件工具—测试—指南 IV. ①TP311.56-62

中国版本图书馆 CIP 数据核字（2019）第 038781 号

责任编辑：夏毓彦
封面设计：王 翔
责任校对：闫秀华
责任印制：董 瑾

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京鑫丰华彩印有限公司

装 订 者：三河市溧源装订厂

经 销：全国新华书店

开 本：190mm×260mm 印 张：13 字 数：333 千字

版 次：2019 年 4 月第 1 版 印 次：2019 年 4 月第 1 次印刷

定 价：49.00 元

产品编号：081809-01

序

关于自动化测试的工具和框架其实有很多。自动化测试在测试 IT 行业中扮演着越来越重要的角色，不管是在传统的 IT 行业还是高速发展的互联网行业或是如今的大数据和火热的人工智能领域，都离不开测试，也更加离不开自动化测试。自动化测试已经发展了很多年，在很多大的互联网公司里他们不但有自己的自动化测试框架，甚至基本都建立了自己的自动化测试平台或者已经对外开放使用的自动化测试云平台，也就是说自动化测试几乎已经成为一名测试工程师必须掌握的一个技能，并且随着很多自动化测试工具和框架的不断发展和完善，自动化测试也变得越来越简单。Robot Framework 是目前众多自动化测试工具或自动化测试框架中一个非常流行的开源框架，致力于解决重复功能测试劳动所带来的高额成本，将自动化测试大众化、简单化、通俗化，让更多没有编程基础的人也能成功地完成自动化测试，降低自动化测试带来的学习成本。

2016 年年初，作者在规划写这本书的时候，其实已经在开始谋划做职位转型，从一名软件测试工程师转型为一名软件开发工程师，在此之前作者曾经在测试职位上打拼了 8 年多，经历了手工功能测试、自动化测试、性能测试。在写这篇序时，作者已经在转型后的软件开发工程师的岗位上工作了 1 年多，之所以说作者自身的经历，其实主要是想告诉每一位想从事自动化测试的读者，只要想去或者想去转型，时间和年纪都不会是太大问题，哪怕你已经 30 岁或者 30 多岁了，都可以重新开启一个新的奋斗起点。作者写这本关于自动化测试框架的书，除了分享一些自身浅薄的经验外，还有一个目的就是想鼓励更多还在从事手工功能测试的读者去学习自动化测试，去从事自动化测试。

这本书并不是完全面向初学者来进行设计的，更多的是比较适合有一定自动化基础的朋

友。在本书的后半部分，更多的谈到如何去设计一个自动化测试框架， 对于一个刚刚接触自动化测试的朋友来说，刚开始可能会稍显吃力，但是随着您对自动化测试的逐步深入，相信您会越来越轻松、越来越喜欢。

感谢孟瑞迪、Monica 等众多挚友在我最困难的时候给予我很多的帮助，正是有了大家的帮助才有了这本关于自动化测试框架的书。由于作者水平有限，书中难免会存在一些不足之处，恳请读者提出宝贵的意见和建议。

作者于南京
2019年1月

目 录

第 1 章 初识 Robot Framework.....	1
1.1 如何创建一个自动化测试项目	2
1.1.1 创建测试项目	3
1.1.2 创建测试套件	3
1.1.3 创建测试用例	4
1.2 Robot Framework 基础关键字	4
1.2.1 如何搜索 Robot Framework 的关键字	4
1.2.2 关键字 log	5
1.2.3 如何在用例中定义一个变量	6
1.2.4 如何快速查询某一个关键字的 API 说明	6
1.2.5 如何快速补全关键字	7
1.2.6 如何定义一个列表	7
1.2.7 如何定义一个字典	8
1.2.8 如何拼接两个字符串	8
1.2.9 如何使用 for 循环	9
1.2.10 如何中断 for 循环	9
1.2.11 Run Keyword If 判断的使用	10
1.2.12 Comment 关键字的使用	10
1.2.13 Return From Keyword 和 Return From Keyword If 关键字的使用	11
1.3 Robot Framework 断言关键字	13
1.3.1 Should Be Equal 关键字的使用	13
1.3.2 Should Be True 关键字的使用	13
1.3.3 Should Contain 关键字的使用	14
1.3.4 Should End With 关键字的使用	14
1.3.5 其他常用断言关键字	15
1.4 BuiltIn 库剩余关键字	16
1.4.1 常用转换类型关键字	16
1.4.2 常用 Get 类型关键字	17
1.4.3 常用 Import 类型关键字	18
1.4.4 常用 Set 类型关键字	18
1.4.5 常用 Run Keyword 类型关键字	19
1.4.6 其他关键字	19

第 2 章 Robot Framework 对数据库的操作.....	21
2.1 DatabaseLibrary 库的使用.....	21
2.1.1 如何连接数据库.....	22
2.1.2 如何断开数据库.....	23
2.1.3 如何对数据库的表进行查询.....	23
2.1.4 如何插入和删除数据.....	24
2.1.5 如何执行数据库脚本文件.....	26
2.1.6 DatabaseLibrary 库的其他操作关键字.....	27
2.2 MongoDBLibrary 库的使用.....	28
2.2.1 MongoDB 数据库的连接和断开.....	29
2.2.2 Get Mongoddb Databases 和 Get Mongoddb Collections.....	32
2.2.3 Save Mongoddb Records.....	34
2.2.4 Retrieve All Mongoddb Records.....	35
2.2.5 Update Many Mongoddb Records.....	36
2.2.6 Remove Mongoddb Records.....	37
2.2.7 MongoDBLibrary 库的其他关键字.....	38
第 3 章 HTTP 接口自动化测试.....	40
3.1 HttpLibrary.HTTP 库的使用.....	40
3.1.1 Create Http Context.....	41
3.1.2 Get.....	42
3.1.3 Get Response Body.....	44
3.1.4 Get Response Status.....	44
3.1.5 Get Response Header.....	45
3.1.6 Set Request Header.....	48
3.1.7 Set Request Body.....	49
3.1.8 Post.....	50
3.1.9 Follow Response.....	51
3.1.10 HttpLibrary.HTTP 库的其他关键字.....	52
3.2 RequestsLibrary 库的使用.....	56
3.2.1 Create Session 和 Get Request.....	57
3.2.2 Post Request.....	59
3.2.3 RequestsLibrary 库的其他关键字.....	60
3.3 RESTinstance 库的使用.....	61
第 4 章 移动手机自动化测试.....	62
4.1 Appium 介绍.....	62
4.1.1 Appium 中的常用术语.....	63
4.1.2 Appium 服务关键字.....	64

4.2	Appium Library 库的使用	69
4.2.1	Open Application	71
4.2.2	Input Text 和 Click Button	75
4.2.3	Clear Text	82
4.2.4	Click Element	84
4.2.5	Click A Point	85
4.2.6	Click Element At Coordinates	85
4.2.7	Get Element Location	86
4.2.8	Get Current Context	87
4.2.9	Get Contexts	87
4.2.10	Switch To Context	87
4.2.11	Get Elements	88
4.2.12	Get Element Attribute	88
4.2.13	Get Network Connection Status 和 Set Network Connection Status	89
4.2.14	Element Attribute Should Match	90
4.2.15	Element Name Should Be 和 Element Value Should Be	91
4.2.16	AppiumLibrary 库其他的常见自动化关键字	91
第 5 章	Web 自动化测试	94
5.1	Selenium Web 自动化	94
5.1.1	Selenium 和 Robot Framework Selenium2Library 库介绍	94
5.1.2	Open Browser 和 Close Browser	96
5.1.3	Input Text	98
5.1.4	Click Button	99
5.1.5	Click Element	101
5.1.6	Click Link	101
5.1.7	Add Cookie、Get Cookie 和 Delete Cookie	103
5.1.8	Get All Links	105
5.1.9	Choose File	106
5.1.10	Get Text	111
5.1.11	Get Title	112
5.1.12	Get Value	113
5.1.13	Get Webelements 和 Get Webelement	114
5.1.14	Get Window Titles	115
5.1.15	Go Back 和 Go To	115
5.1.16	Get List Items	117
5.1.17	Get Selected List Value	117
5.1.18	Select From List	119
5.1.19	Selenium2Library 库其他的自动化测试关键字介绍	120

5.2	SikuliLibrary 库的使用.....	128
5.2.1	Sikuli 简介.....	128
5.2.2	SikuliLibrary 的使用.....	129
5.2.3	SikuliLibrary 的工作原理.....	133
5.2.4	SikuliLibrary 常用关键字介绍.....	135
第 6 章	编写自定义的 Robot Framework Lib.....	137
6.1	使用 Python 编写自定义的 Robot Framework Lib	137
6.1.1	使用 Python 构建 Lib 工程	137
6.1.2	使用 Python 编写自定义的 Lib	141
6.1.3	打包自定义的 Lib.....	143
6.1.4	Remote 远程库.....	145
6.2	使用 Java 编写自定义的 Robot Framework Lib	150
6.2.1	在 Robot Framework 中调用 Java Lib 库	150
6.2.2	使用 Java 编写自定义的 Lib.....	156
第 7 章	自动化测试用例的编写技巧	164
7.1	自动化测试用例的常用技巧	164
7.1.1	自动化测试用例的容错	164
7.1.2	自动化测试用例的测试数据初始化和脏数据的处理	166
7.2	如何高效地维护好自动化测试用例	167
7.2.1	提取出共用变量统一维护	167
7.2.2	在单个自动化测试用例中多使用变量	170
7.2.3	提取复用的业务或者步骤, 封装自定义的用户关键字.....	171
7.2.4	封装全局可用的资源文件	173
第 8 章	自动化测试框架的设计	177
8.1	Jenkins 下自动化测试的调度管理	177
8.1.1	Jenkins 介绍	177
8.1.2	在 Jenkins 上运行 Robot Framework 自动化测试用例	183
8.2	如何做好自动化测试平台框架的设计	188
8.3	其他常用的自动化测试框架介绍	193
8.3.1	RedwoodHQ 介绍	193
8.3.2	Cucumber 介绍	198

第 1 章

初识Robot Framework

Robot Framework 是一款基于 Python 编程语言设计的、可扩展的、关键字驱动模式的测试自动化框架，具备良好的可扩展性，可以通过 XML-RPC 服务扩展支持其他的常用编程语言，可以同时测试多种类型的客户端或者接口，可以支持进行分布式测试执行。

Robot Framework 具体的特点如下：

- 易于使用，采用表格式输入语法以及统一的测试用例（Test Case，也叫测试案例）格式。
- 重用性好，可以利用现有关键字来组合新的用户自定义关键字。
- 支持资源文件，支持多种变量类型，包括字符串变量、List 列表变量、Dictionary 字典变量等。
- 测试用例执行结果报告和日志采用 HTML 格式，易于阅读和邮件转发。
- 提供标签以分类来选择将被执行的测试用例，使得测试用例的选择更加灵活。
- 支持 Web 界面测试、Web 接口服务测试、GUI 测试、多种终端测试。
- 支持多种数据库的操作，包括常用的关系型数据库、非关系型数据库。
- 易于扩展自定义的 Lib 库，可以通过 Python 或者 Java 等其他开发语言来动态扩展 Lib 库。

Robot Framework 自动化测试框架的组成如图 1-0-1 所示。

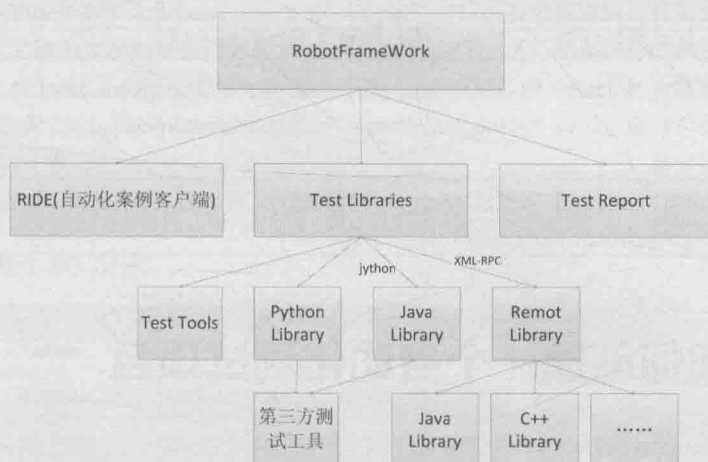


图 1-0-1

- 集成了很多流行的自动化测试工具，比如 Appium、Selenium 等。

- 通过 Jython 的方式, 使得以 Python 为主的 Robot Framework 自动化测试框架无缝地与 Java 语言进行完美集成, 也可以通过 XML-RPC 远程调用协议来支持 Java 或者 C++ 等流行的编程语言, 使对 Python 语言不熟悉的编程爱好者也可以编写自定义的 Library 库。
- 在 Robot Framework 中, 使用 Python 语言实现了自动化测试用例编写的客户端 RIDE, 使用 RIDE 可以非常简单地完成自动化测试用例的编写, 也可以使用 RIDE 完成用户层面的关键字 API 的封装, 使得不懂任何编程语言的自动化测试爱好者也可以封装自己的 API 关键字。

Robot Framework 除了提供了我们上面提到的 Ride 外, 还提供了很多常用的插件工具, 如表 1-0-1 所示。

表 1-0-1 Robot Framework 常用的插件工具

工具插件	说明
Eclipse plugin	Robot Framework 为 Eclipse IDE 开发工具提供的插件, 使得用户也可以在 Eclipse 上编写自动化测试用例, GitHub 地址为: https://github.com/NitorCreations/RobotFramework-EclipseIDE/wiki
Robot Plugin for IntelliJ IDEA	和 Eclipse plugin 类似, 是为另一个常用的 IDE 开发工具 IntelliJ IDEA 提供的插件, 使得用户也可以在 IntelliJ IDEA 上编写自动化测试用例, GitHub 地址为: https://plugins.jetbrains.com/plugin/7430-robot-plugin
Jenkins plugin	这是一个 Jenkins 上使用的插件, 这个插件可以使得 Robot Framework 完美地集成在当今非常流行的持续集成工具 Jenkins 上, 插件的访问网址为: https://wiki.jenkins-ci.org/display/JENKINS/Robot+Framework+Plugin
Maven plugin	这是 Robot Framework 提供的 maven 仓库插件, 可以通过访问网址“ http://robotframework.org/MavenPlugin/ ”获取, 当前最新的版本为 1.4.7
Ant task	这是为另一个打包工具 ant 提供的执行插件, 使得 Robot Framework 可以通过 ant 的方式来运行。可以通过访问网址“ http://code.google.com/p/robotframework-ant/ ”获取该插件
Pabot	Robot Framework 提供的并发执行器, 也就是我们通常说的多线程并发执行模式, 可以通过在 Windows 的 cmd 下执行 <code>pip install -U robotframework-pabot</code> 命令进行在线安装, 也可以通过访问 GitHub 网址“ https://github.com/mkorpela/pabot ”进行下载, 然后离线进行安装
Atom plugin	Robot Framework 为支持 Atom 而开发的插件, 可以通过访问网址“ https://atom.io/packages/language-robot-framework ”进行下载

1.1 如何创建一个自动化测试项目

一个 Robot Framework 项目其实就和一个我们平时熟知的单元测试项目结构基本是一样的, 也包含了测试套件和测试用例的概念。我们可以对 Robot Framework 项目结构做如图 1-1-1 所示的划分。

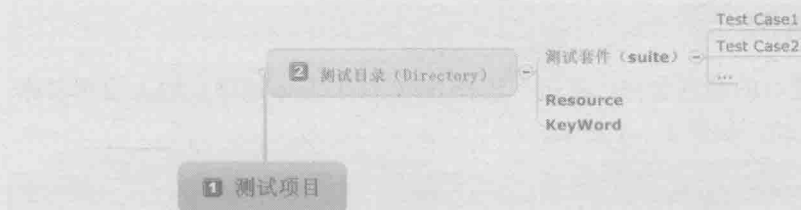


图 1-1-1

1.1.1 创建测试项目

在 Robot Framework 中，Ride 是一款用 Python 语言实现的用来做自动化测试用例编写的客户端工具。通过访问网址“<https://pypi.org/project/robotframework-ride/>”即可下载 Ride 工具包进行离线安装，也可以通过在 Windows 的 cmd 命令行中输入“`pip install robotframework-ride`”进行在线自动安装。安装完成后打开 Ride，选择菜单栏 File→New Project，在 Name 文本框中输入项目名称，此处 Type 我们选择 Directory，单击 OK 按钮，即可创建成功，如图 1-1-2 所示。

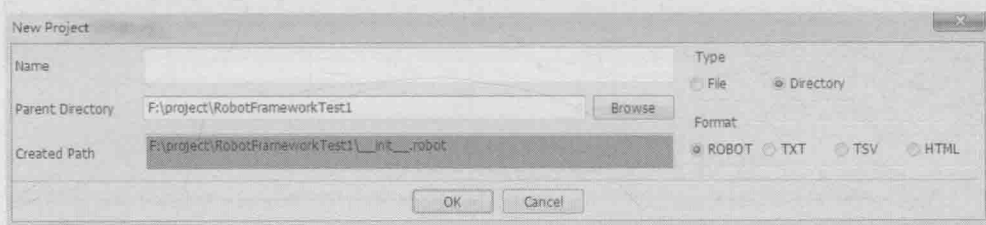


图 1-1-2

存储格式支持多种类型，如表 1-1-1 所示。

表 1-1-1 存储格式支持的类型

储存类型	说明
Type	项目存储方式：文件形式或者目录形式，一般建议选择目录形式
Format	文件存储格式：提供了 ROBOT（默认格式）、TXT、TSV 和 HTML 四种格式

1.1.2 创建测试套件

选择上面我们创建好的项目，右击鼠标键，选择 New Suite 选项，输入测试套件名称，即可创建成功，如图 1-1-3 所示。

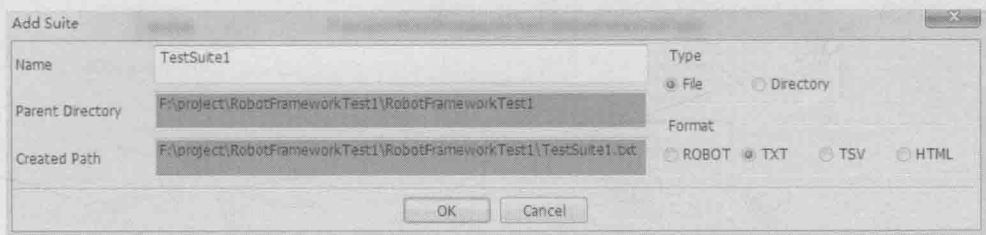


图 1-1-3

1.1.3 创建测试用例

选择上面我们创建好的测试套件，右击，选择 New Test Case 选项，输入用例名称，单击 OK 按钮，即可创建成功，如图 1-1-4 所示。

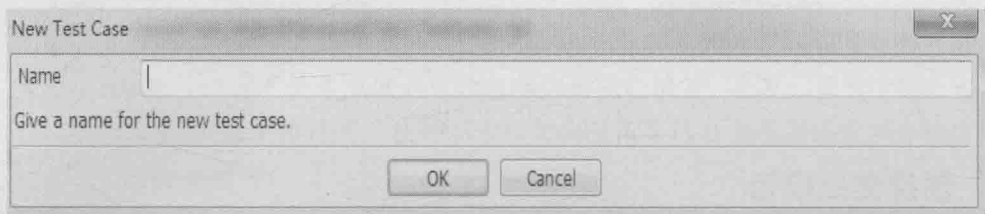


图 1-1-4

创建成功后，即可看到下面的用例编写表格，如图 1-1-5 所示。通过此表格，我们就可以编写测试用例了。

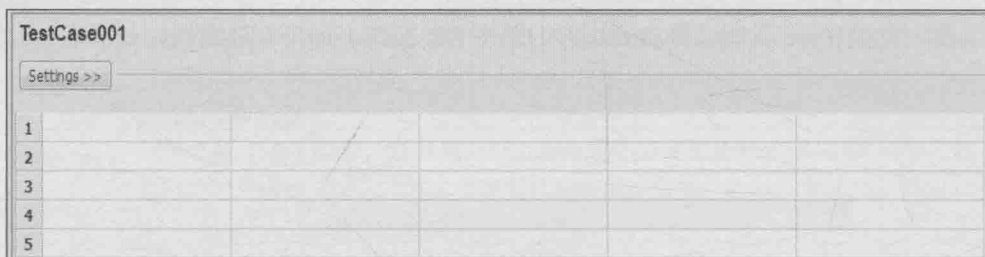


图 1-1-5

1.2 Robot Framework 基础关键字

1.2.1 如何搜索 Robot Framework 的关键字

有两种方式可以快速地打开 RIDE 的关键字搜索对话框。

(1) 选择菜单栏中的 Tools→Search Keywords 选项，然后会出现如图 1-2-1 所示的关键字搜索对话框，这个对话框就类似于提供了一个关键字的 API 功能（提供了关键字的名称、关键字的来源库、关键字的使用描述和关键字的参数）。

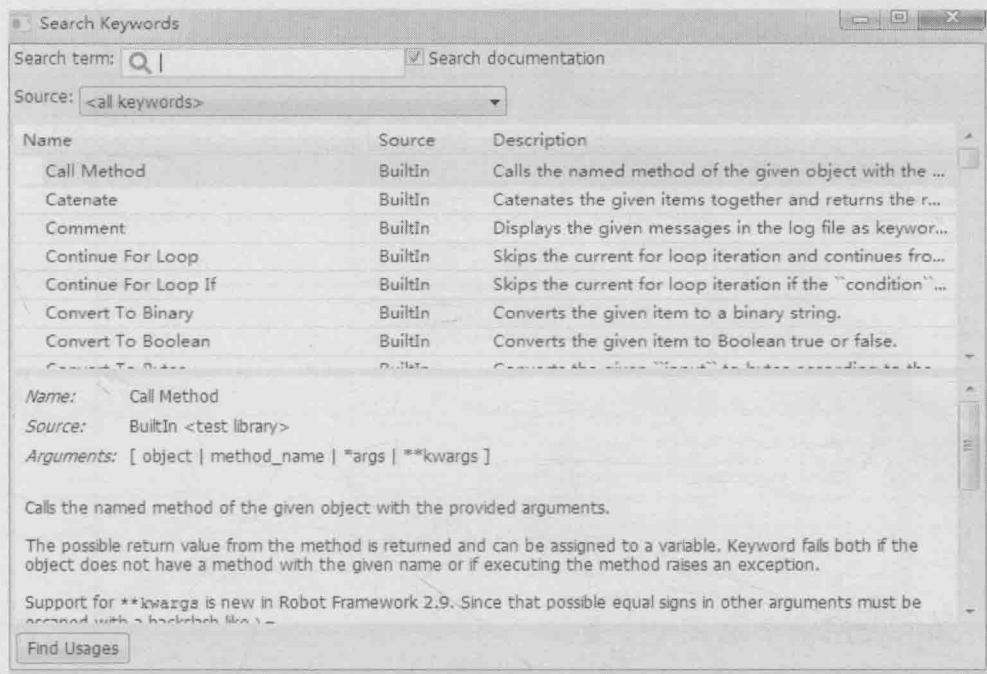


图 1-2-1

(2) 直接按 F5 快捷键，就可以自动弹出我们需要的关键字搜索框。

1.2.2 关键字 log

Log 关键字其实就等同于 Python 语言中的 print 函数，可以输出我们想要输出的内容（也就是我们在编程语言中常说的日志输出），比如我们在 test case 中输入如图 1-2-2 所示的内容。

1	log	Hello RobotFramework
2		

图 1-2-2

勾选我们的测试用例，单击菜单栏 Tools→Run Tests（或者直接快捷键 F8）来执行这条测试用例，如图 1-2-3 所示。

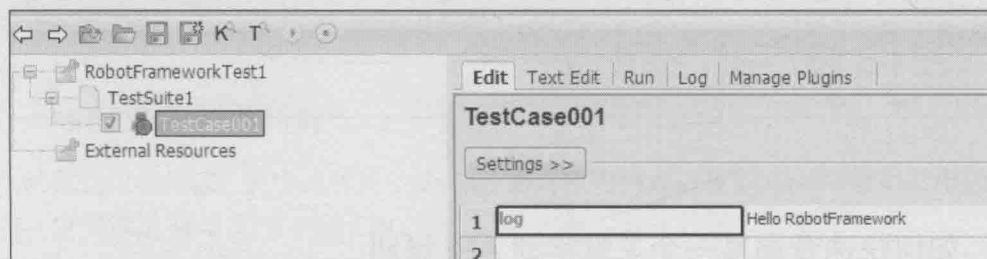


图 1-2-3

执行完成后，切换到 Run 标签，可以看到用例执行的结果。通过运行结果可以看到输出

了我们想要输出的信息 INFO : Hello RobotFramework，如图 1-2-4 所示。

```

elapsed time: 0:00:01  pass: 1  fail: 0
-----
TestCase001 | PASS |
-----
RobotFrameworkTest1.TestSuite1 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
RobotFrameworkTest1 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
Output:  c:\users\yongqing\AppData\Local\Temp\RIDE4nytoe.d\output.xml
Log:     c:\users\yongqing\AppData\Local\Temp\RIDE4nytoe.d\log.html
Report:  c:\users\yongqing\AppData\Local\Temp\RIDE4nytoe.d\report.html

test finished 20170304 23:05:49

Starting test: RobotFrameworkTest1.TestSuite1.TestCase001
20170304 23:05:49.671 : INFO : Hello RobotFramework
Ending test:  RobotFrameworkTest1.TestSuite1.TestCase001

```

图 1-2-4

1.2.3 如何在用例中定义一个变量

我们可以通过 Set Variable 来定义一个变量，比如我们定义一个变量 var1，并且将这个变量赋值为 Robot，然后将这个变量用 log 输出，如图 1-2-5 所示。

`\${var1}`	Set Variable	Robot
log	`\${var1}`	

图 1-2-5

执行结果如图 1-2-6 所示。

```

-----
TestCase001 | PASS |
-----
RobotFrameworkTest1.TestSuite1 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
RobotFrameworkTest1 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
Output:  c:\users\yongqing\AppData\Local\Temp\RIDE4nytoe.d\output.xml
Log:     c:\users\yongqing\AppData\Local\Temp\RIDE4nytoe.d\log.html
Report:  c:\users\yongqing\AppData\Local\Temp\RIDE4nytoe.d\report.html

test finished 20170304 23:16:32

Starting test: RobotFrameworkTest1.TestSuite1.TestCase001
20170304 23:16:32.589 : INFO : Hello RobotFramework
20170304 23:16:32.589 : INFO : ${var1} = Robot
20170304 23:16:32.599 : INFO : Robot
Ending test:  RobotFrameworkTest1.TestSuite1.TestCase001

```

图 1-2-6

1.2.4 如何快速查询某一个关键字的 API 说明

选中关键字，同时按住 Ctrl+Alt 组合键，即可显示该关键字的帮助 API 以及使用示例，如图 1-2-7 所示。

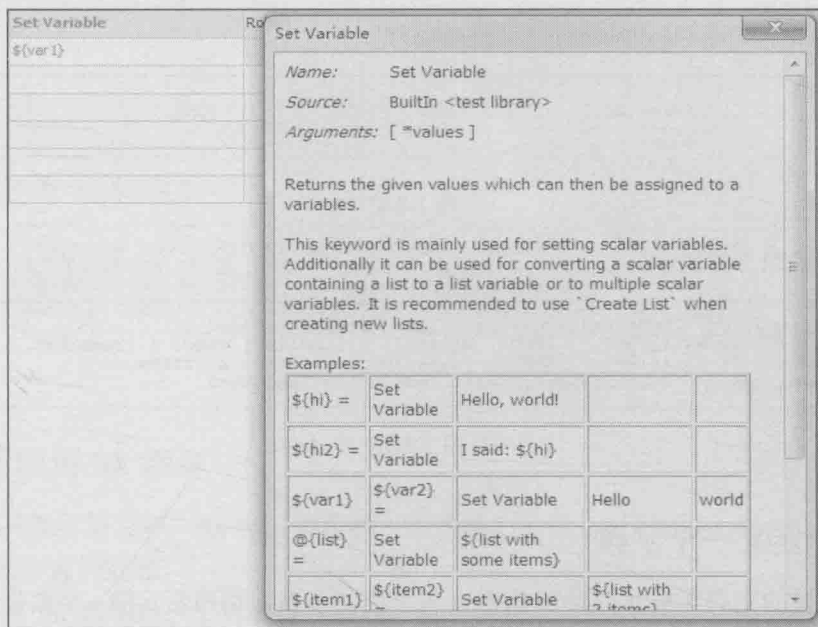


图 1-2-7

1.2.5 如何快速补全关键字

通过键盘输入关键字的前缀，然后同时按住 `Ctrl+Alt+空格键`，即可快速补全某个关键字，如图 1-2-8 所示。

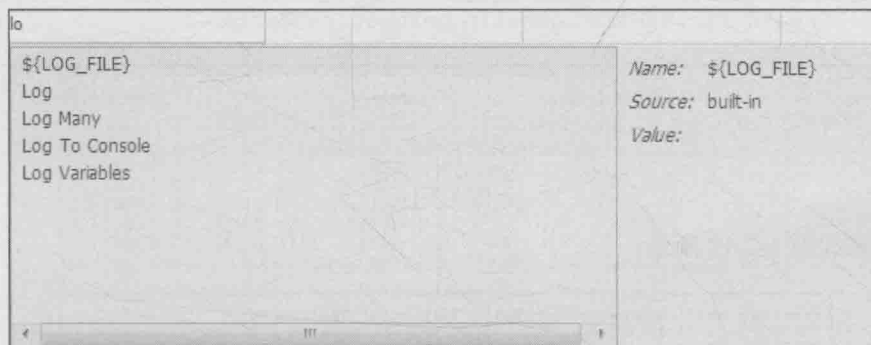


图 1-2-8

1.2.6 如何定义一个列表

此处我们说的列表，其实就等同于 Python 语言中的列表，是 Python 语言中常用的一种数据结构，也类似于 Java 语言中的 List。

在 Robot Framework 中，我们可以使用 `Create List` 来创建一个列表，比如我们定义一个列表 `list1`，并且在创建列表时就添加 3 个元素。然后使用 `log` 关键字将这个列表中的元素全部输出，如图 1-2-9 所示。


```
@{list1} Create List hello robot framework
log ${list1}
```

1	@{list1}	Create List	hello	robot	framework
2	log	\${list1}			

图 1-2-9

执行结果如图 1-2-10 所示。

```
Starting test: RobotFrameworkTest1.TestSuite1.TestCase2
20170305 22:53:44.087 : INFO : @{list1} = [ hello | robot | framework ]
20170305 22:53:44.089 : INFO : [u'hello', u'robot', u'framework']
Ending test: RobotFrameworkTest1.TestSuite1.TestCase2
```

图 1-2-10

1.2.7 如何定义一个字典

此处我们说的字典其实就等同于 Python 语言中的字典，和列表一样，字典也是 Python 语言中非常常用的一种数据结构，也类似于 Java 语言中的 Map。

在 Robot Framework 中，使用 Create Dictionary 来创建一个字典，比如我们定义一个字典 Dict1，并且在创建字典时就添加两个键值对，然后使用 Log Many 关键字将这个字典中的内容全部输出，如图 1-2-11 所示。

Log Many 关键字类似于 log 关键字，不同的是 log 关键字只可以接收一个参数，而 Log Many 关键字可以同时接收多个参数。

```
&{Dict1} Create Dictionary a=hello b=robotframework
Log Many &{Dict1}
```

1	&{Dict1}	Create Dictionary	a=hello	b=robotframework
2	Log Many	&{Dict1}		

图 1-2-11

执行结果如图 1-2-12 所示。

```
Starting test: RobotFrameworkTest1.TestSuite1.TestCase003
20170305 23:23:49.439 : INFO : &{Dict1} = { a=hello | b=robotframework }
20170305 23:23:49.439 : INFO : a=hello
20170305 23:23:49.439 : INFO : b=robotframework
Ending test: RobotFrameworkTest1.TestSuite1.TestCase003
```

图 1-2-12

1.2.8 如何拼接两个字符串

我们可以通过 Catenate 来拼接字符串，比如将“Hello”和“Robot”这两个字符串拼接起来并且输出，如图 1-2-13 所示。

```
${val2} Catenate Hello Robot
log ${val2}
```