



Selenium WebDriver 3.0

自动化测试框架 实战指南

吴晓华 王晨昕 编著

Python语言版

75个项目实例 5种测试框架

- ★ 分布式测试框架
- ★ 行为驱动测试框架
- ★ 数据驱动测试框架
- ★ 关键字驱动测试框架
- ★ 数据关键字混合驱动测试框架

清华大学出版社





Selenium WebDriver 3.0 自动化测试框架 实战指南

吴晓华 王晨昕 编著

清华大学出版社
北京

内 容 简 介

本书以分享 Selenium WebDriver 实战经验为要,致力于为 Tester 讲解开源的 Web 自动化测试工具的原理、api 接口实例、unittest、五大框架实战(行为驱动、分布式测试框架、数据驱动测试框架、关键词驱动测试框架、混合模式测试框架)以及自动化测试中常见的问题处理。

本书既可以让初学者从入门到精通,循序渐进;也可帮中高级 Tester 夯实技能,从形象到抽象,提供测试思想中更多的可能性。

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

Selenium WebDriver 3.0 自动化测试框架实战指南/吴晓华,王晨昕编著. —北京:清华大学出版社, 2017

ISBN 978-7-302-48317-5

I. ①S… II. ①吴… ②王… III. ①软件工具—自动检测—指南 IV. ①TP311.56-62

中国版本图书馆 CIP 数据核字(2017)第 218517 号

责任编辑:贾 斌 薛 阳

封面设计:刘 键

责任校对:李建庄

责任印制:沈 露

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.75 字 数:646 千字

版 次:2017 年 9 月第 1 版 印 次:2017 年 9 月第 1 次印刷

印 数:1~3000

定 价:69.00 元

产品编号:075534-01



吴晓华

软件测试界的一个老兵，光荣之路软件测试开发免费公开课创始人，从事软件测试工作13年，热衷于传递软件测试行业的最佳实践经验，曾任职于搜狐公司、搜狗公司、Tom在线等知名企业。

自光荣之路软件测试开发免费公开课创立以来，听众数万人，公开课视频下载数十万次，受到广大测试从业人员的好评。



王晨昕

某大型互联网高级测试开发，多年的测试开发经验，喜欢专研各种测试开发技术，并热衷于分享这些技术。



“光荣之路”公众号

“光荣之路”微信公众号: gloryroadtrain

QQ: 875821166

欢迎大家一起分享和交流测试技术!

试读结束 需要全本请在线购买: www.ertongbook.com

前言

随着互联网的高速发展,中国的互联网达到了一个空前的繁荣水平,数亿量级用户的产品登上了中国的互联网发展舞台,阿里巴巴、腾讯、百度等多个互联网巨头也开始在世界的互联网舞台崭露头角,互联网行业的从业人员也达到了上百万人的规模,中国的互联网产品已经深入到网民生活的各个方面。

随着互联网行业在中国的迅猛发展,对于中国的软件开发和测试行业也提出了更高的技术要求与质量要求,软件测试从业者的技术水平也被提升到空前的高要求阶段。以往我们看到测试人员的招聘重点都是仅限于对测试用例设计和业务的理解,现今所看到的更多测试职位对测试人员提出了更高的技术能力要求。例如,精通一门编程语言,熟悉 MySQL 或者 Oracle 数据库,精通自动化测试和性能测试,能独立开发测试工具等。为了能更好地适应互联网社会的发展潮流,软件测试从业者必须在技术能力上不断地提升自己,才能真正站在职业发展的巅峰。

自动化测试技术对测试人员来说,是一个必要的高级技能要求,越来越多的测试从业者并不甘于手工测试,都非常希望通过自动化的方式来减少枯燥无味且不断重复的手工测试劳动。尽管,主流的 Web 自动化测试开源工具 Selenium WebDriver,已经成为众多软件测试从业者学习的热点,但是市面上针对 Selenium 自动化测试方面的书籍很少,基于实践方式来讲解 Selenium 应用技术的书籍更是凤毛麟角。我有幸受吴老邀请,将我工作中实践内容与吴老教学内容融合,一起编写了这本基于 Python 语言实践操作的 Selenium 3 教学书籍,来解决软件测试人员学习自动化测试的需求。

Selenium 是一个开源的测试工具,代表了未来测试工具的趋势;而 Python 则是全世界都在用的一门简洁、高效、易用、优雅的编程语言,初学者只需要花少量的时间就能上手,并完成一定量的开发任务。但本书着重点在讲解 Selenium 的使用技巧上,对学习 Python 语言有需要的朋友,请自行购买相关资料或者利用丰富的网络资源。

本书采用图文并茂的方式分步骤讲解 Selenium 的各种实用技巧,并且提供被测试对象的实现代码或者被测试对象的访问网址,方便读者在本地搭建自己的测试环境或者访问互联网上的被测试网址,从而能顺利地进行自动化测试技术的实践。经过我们数月的不懈努力,此书终于跟大家见面了,希望能够让读者通过本书深入掌握 Selenium 3 的使用技巧,帮助大家自动化测试方向上能大展身手。我们相信,通过我们不断的努力,一定可以改变中国测试行业技术含量低的现状。

作为一个土生土长的测试人员,不懂技术,工作中经常会被各种大咖人物丢来一个不屑的眼神,甚至有的人连不屑的眼神都懒得丢,直接无视,此时我的内心是很崩溃的(大哭)。2015年,在一个机缘巧合下认识了吴老,他丰富的测试经验和过硬的技术都让我折服,我就

像是一粒罅隙中的种子遇到了阳光,拼命地将自己置身于这片透着温暖的金色的阳光浴中。因为好不容易遇见这么一位低调奢华有内涵的大神,岂能放过(偷笑),于是我很乐意就死皮赖脸地揪着吴老,套他的各种本领。当时的每一天除了吃饭、工作及少量的睡觉时间,其余时间不是吃斋(看书),就是念佛(敲代码),那日子甚是枯燥,甚是无味,期盼着咸鱼能有翻身的一天。我待 Code 如初恋,Code 虐我千百遍(苦笑),半年时间过去了,被虐得千疮百孔的我也算是摸清 Code 的脾气了,闲来无事也能写上几百行了,算是有点欣慰吧。

有了一定的 Code 功底后,开始正式进入自动化学习。自动化测试的学习是一个不断实践,不断总结,不断积累的过程。很多人会有一个错误的认识,认为自动化测试不就是一个工具的使用,一个调用别人写好的 API 的过程么,需要那么深厚的 Code 功底有何用?殊不知你此时仅是一个 ToolBoy 或者 ToolGirl,要想随心所欲地完成各种自动化测试,无论是 Web 自动化、移动端自动化,还是性能自动化,都必须在拥有一定的 Code 功底后,方能理解这些工具底层实现的原理,并且能在当工具本身不能满足测试需求时,还能随心所欲地扩充或更改。理解了这些工具的实现思想,也就为您搭建属于自己的测试框架打下了坚实的基础。经过半年的蹂躏与被蹂躏,加之工作中的不断实践,也能搭建那么几个自己还算满意的测试框架,做得还甚是开心(偷笑),而且还能被吴老盯上,叫来编写这本书,也着实有点小开心(得意)。

这是我第一次写一本专著技术类的书籍,深深地感觉到把知识点用通俗易懂的语言描述清楚是一件多么不容易的事,为此我投入了大量的时间与精力来组织本书的语言,如果还是存在不那么浅显易懂的语句,请先尝试从代码层面进行理解,但如果在实践本书中代码时发生了错误,请不要怀疑是我们代码的问题(坏笑),请先检查您的环境是否有问题,浏览器版本与驱动版本是否匹配,所使用的 Python 包版本之间是否存在版本兼容的问题,Python 代码是否存在缩进问题等。如果仍然解决不了您的问题,欢迎把错误丢到作者的脸上,我们定会马不停蹄地帮您解决。

冰冻三尺,非一日之寒,希望大家能在自动化测试学习的道路上做到博观而约取,厚积而薄发。最后,祝大家工作顺利,万事如意。

本书内容介绍如下:

第一篇 基础篇(第 1~8 章)

第 1 章介绍 Selenium 的发展历史及组成 Selenium 的工具套件,列举了 Selenium 1 和 Selenium 2 支持的浏览器和平台,讲解 Selenium RC 和 WebDriver 的实现原理,同时也介绍了 Selenium 1、Selenium 2 和 Selenium 3 的各自特点及区别。

第 2 章介绍在日常测试工作中常见的自动化测试目标,讲解如何获得公司管理层对于开展自动化测试的支持、如何衡量自动化测试工作的投入产出比及在敏捷开发中的应用、以及自动化测试工作的分工及测试工具的选择与推广,分享在实际项目中最佳的实践经验,说明了学习 Selenium 工具的能力要求。

第 3 章介绍使用 Selenium 工具时所需要的相关辅助插件 FireBug 和 FirePath 的安装及使用方法。

第 4 章介绍 Selenium IDE 的安装、界面选项的含义、IDE 的使用方法、录制脚本及导出脚本等。

第 5 章介绍 Python 开发环境和 Pycharm 集成开发环境的安装、配置及使用。



第 6 章介绍 WebDriver 的安装和配置方法。

第 7 章主要介绍单元测试的基本知识,如何在自动化测试中使用以及生成测试报告。

第 8 章主要讲解自动化测试过程中使用的页面元素定位方法,包括 ID 定位、Name 定位、链接文字定位、Class 定位、XPath 定位及 CSS 定位,推荐使用 XPath 作为页面元素定位的主要方法。

第二篇 实战应用篇(第 9~11 章)

第 9 章讲解如何使用 WebDriver 工具分别驱动 IE 浏览器、Chrome 浏览器以及 Firefox 浏览器,进行自动化测试。

第 10 章通过实例全面讲解 WebDriver 基础 API。

第 11 章通过实例全面讲解 WebDriver 高级 API,并提供一些解决实际问题的方法。

第三篇 自动化测试框架搭建篇(第 12~15 章)

第 12 章讲解数据驱动的概念,并基于 Excel、XML、MySQL 及单元测试框架结合 ddt 进行数据驱动测试。

第 13 章讲解 lettuce 行为驱动框架在自动化测试中的使用,分别基于英文和中文进行实例讲解。

第 14 章通过实例全面讲解如何基于 Selenium Grid 进行分布式自动化测试。

第 15 章深入讲解如何从零开始搭建一个数据驱动测试框架、关键字驱动测试框架以及数据驱动与关键字驱动混合的测试框架,并提供完整的框架代码。此章节为本书最综合、最重要的章节,建议读者在阅读完前面所有章节后再阅读此章节。

第四篇 常见问题和解决方法(第 16 章)

第 16 章讲解 WebDriver 使用过程中常见的疑难问题及解决办法,方便读者在使用 WebDriver 的过程中遇到问题时进行查阅。

特别致谢

感谢我们的好朋友陈良军、李江和王浩花费大量的时间与精力帮我们校对书稿,发现了不少书写的错误、晦涩难懂的语句以及代码的错误,在此笔者们真诚地感谢他们对本书做出的贡献,帮助我们完成这件非常有意义的事情。

吴晓华 王晨昕

2017 年 6 月



第一篇 基础篇

第 1 章 Selenium 简介	1
1.1 Selenium 的前世今生	1
1.2 Selenium 工具套件介绍	2
1.3 Selenium 支持的浏览器和平台	2
1.3.1 Selenium IDE、Selenium 1 和 Selenium RC 支持的浏览器和平台	2
1.3.2 Selenium 2(WebDriver)支持的浏览器	3
1.4 Selenium RC 和 WebDriver 的实现原理	4
1.4.1 Selenium RC 的实现原理	4
1.4.2 WebDriver 的实现原理	6
1.4.3 Selenium 1.0 和 WebDriver 的特点	6
1.5 Selenium 3 的新特性	7
第 2 章 自动化测试那点事儿	8
2.1 自动化测试目标	8
2.2 管理层的支持	11
2.3 投入产出比	11
2.4 敏捷开发中的自动化测试应用	12
2.5 自动化测试人员分工	14
2.6 自动化测试工具的选择和推广使用	14
2.6.1 自动化测试工具的选择	14
2.6.2 Selenium WebDriver 和 QTP 工具的特点比较	14
2.7 在项目中实施自动化的最佳实践	15
2.8 学习 Selenium 工具的能力要求	17
第 3 章 自动化测试辅助工具	18
3.1 安装 Firefox 浏览器	18
3.2 安装 Firebug 插件	18
3.3 Firebug 插件的使用	20
3.3.1 启动 Firebug 插件	20
3.3.2 Firebug 插件的常用功能	20

3.4	安装 FirePath 插件	22
3.5	FirePath 插件的使用	24
3.5.1	FirePath 插件中使用 XPath 定位方式	24
3.5.2	FirePath 插件中使用 CSS 定位方式	26
3.6	IE 浏览器自带的辅助开发工具	28
第 4 章	Selenium IDE	30
4.1	Selenium IDE 的定义	30
4.2	安装 Selenium IDE	30
4.2.1	从 Selenium 官网安装	30
4.2.2	使用离线 XPI 安装文件安装	32
4.3	Selenium IDE 插件界面和功能介绍	33
4.3.1	主界面	33
4.3.2	常用工具栏	34
4.3.3	脚本编辑区域	34
4.4	常用菜单项	34
4.4.1	“文件”菜单	34
4.4.2	“编辑”菜单	35
4.4.3	Actions 菜单	35
4.4.4	Option 菜单	36
4.5	录制和回放脚本实例	38
4.6	Selenium IDE 脚本介绍——Selenese	39
4.7	Selenium IDE 的基本命令使用实例	40
4.7.1	waitForText、verifyText 和 assertText 命令	40
4.7.2	storeTitle 命令和 echo 命令	44
4.7.3	openWindow 命令和 selectWindow 命令	44
4.8	从 Selenium IDE 导出脚本	45
4.8.1	导出脚本文件	45
4.8.2	将 Selenium IDE 插件中的某行命令导出为 Python 脚本	48
第 5 章	搭建 Python 环境和 PyCharm 集成开发环境	49
5.1	安装 Python 并配置 Python 环境	49
5.1.1	下载并安装 Python 解释器	49
5.1.2	配置 Python 环境	50
5.1.3	安装 pip	52
5.2	安装 Python 集成开发环境 PyCharm	53
5.3	新建一个 Python 工程	55
第 6 章	Selenium3(WebDriver)的安装	58
6.1	在 Python 中安装 WebDriver	58
6.2	第一个 WebDriver 脚本	59



6.3	各浏览器驱动的使用方法	62
第7章	单元测试框架的使用介绍	63
7.1	单元测试的定义	63
7.2	unittest 单元测试框架	63
7.2.1	unittest 的定义	63
7.2.2	unittest 框架的4个重要概念	63
7.2.3	单元测试加载方法	64
7.2.4	测试用例	64
7.2.5	测试集合	69
7.2.6	按照特定顺序执行测试用例	71
7.2.7	忽略某个测试方法	74
7.2.8	命令行模式执行测试用例	75
7.2.9	批量执行测试模块	77
7.2.10	常用的断言方法	80
7.2.11	使用 HTMLTestRunner 生成 HTML 测试报告	85
7.3	在 unittest 中运行第一个 WebDriver 测试用例	87
第8章	页面元素定位方法	89
8.1	定位页面元素方法汇总	89
8.2	使用 ID 定位	90
8.3	使用 name 定位	91
8.4	使用链接的全部文字定位	92
8.5	使用部分链接文字定位	92
8.6	使用 HTML 标签名定位	93
8.7	使用 Class 名称定位	94
8.8	使用 XPath 定位	94
8.8.1	XPath 的定义	94
8.8.2	XPath 节点	95
8.8.3	XPath 定位语法	96
8.8.4	XPath 运算符	101
8.9	CSS 定位	103
8.9.1	CSS 的概念	103
8.9.2	CSS 定位语法	103
8.9.3	XPath 定位与 CSS 定位的比较	109
8.10	表格的定位	110
8.10.1	遍历表格所有的单元格	110
8.10.2	定位表格中的某个元素	112
8.10.3	定位表格中的子元素	112

第二篇 实战应用篇

第 9 章 WebDriver 的多浏览器测试	115
9.1 使用 IE 浏览器进行测试	115
9.2 使用 Firefox 浏览器进行测试	116
9.3 使用 Chrome 浏览器进行测试	117
第 10 章 WebDriver API 详解	119
10.1 访问某个网址	119
10.2 网页的前进和后退	120
10.3 刷新当前网页	120
10.4 浏览器窗口最大化	120
10.5 获取并设置当前窗口的位置	121
10.6 获取并设置当前窗口的大小	121
10.7 获取页面的 Title 属性值	122
10.8 获取页面 HTML 源代码	122
10.9 获取当前页面的 URL 地址	123
10.10 获取与切换浏览器窗口句柄	123
10.11 获取页面元素的基本信息	124
10.12 获取页面元素的文本内容	125
10.13 判断页面元素是否可见	125
10.14 判断页面元素是否可操作	127
10.15 获取页面元素的属性	128
10.16 获取页面元素的 CSS 属性值	128
10.17 清空输入框中的内容	128
10.18 在输入框中输入指定内容	129
10.19 单击按钮	130
10.20 双击某个元素	130
10.21 操作单选下拉列表	131
10.21.1 遍历所有选项并打印选项显示的文本和选项值	131
10.21.2 选择下拉列表元素的三种方法	132
10.22 断言单选列表选项值	133
10.23 操作多选的选择列表	133
10.24 操作可以输入的下拉列表(输入的同时模拟按键)	134
10.25 操作单选框	135
10.26 操作复选框	136
10.27 断言页面源码中的关键字	137
10.28 对当前浏览器窗口截屏	138
10.29 拖拽页面元素	138



10.30	模拟键盘单个按键操作	139
10.31	模拟组合按键操作	140
10.31.1	通过 WebDriver 内建的模块模拟组合键	140
10.31.2	通过第三方模块模拟组合按键	141
10.31.3	通过设置剪贴板实现复制和粘贴	146
10.32	模拟鼠标右键	147
10.33	模拟鼠标左键按下与释放	148
10.34	保持鼠标悬停在某个元素上	149
10.35	判断页面元素是否存在	150
10.36	隐式等待	151
10.37	显式等待	152
10.38	显式等待中期望的场景	154
10.39	使用 Title 属性识别和操作新弹出的浏览器窗口	157
10.40	通过页面的关键内容识别和操作新浏览器窗口	159
10.41	操作 Frame 中的页面元素	160
10.42	使用 Frame 中的 HTML 源码内容操作 Frame	162
10.43	操作 IFrame 中的页面元素	163
10.44	操作 JavaScript 的 Alert 弹窗	165
10.45	操作 JavaScript 的 confirm 弹窗	166
10.46	操作 JavaScript 的 prompt 弹窗	167
10.47	操作浏览器的 Cookie	169
10.48	指定页面加载时间	170
第 11 章	WebDriver 高级应用	172
11.1	使用 JavaScript 操作页面元素	172
11.2	操作 Web 页面的滚动条	173
11.3	在 Ajax 方式产生的浮动框中,单击选择包含某个关键字的选项	174
11.4	结束 Windows 中浏览器的进程	177
11.5	更改一个页面对象的属性值	178
11.6	无人工干预地自动下载某个文件	181
11.7	无人工干预地自动上传附件	183
11.7.1	使用 WebDriver 的 send_keys 方法上传文件	183
11.7.2	模拟键盘操作,实现上传文件	184
11.7.3	使用第三方工具 AutoIt 上传文件	187
11.8	右键另存为下载文件	192
11.9	操作日期控件	195
11.10	启动带有用户配置信息的 Firefox 浏览器窗口	196
11.11	UI 对象库	199
11.12	操作富文本框	201
11.13	精确比较页面截图图片	207

11.14	高亮显示正在操作的页面元素	210
11.15	浏览器中新开标签页(Tab)	211
11.16	测试过程中发生异常或断言失败时进行屏幕截图	212
11.17	使用日志模块记录测试过程中的信息	215
11.18	封装操作表格的公用类	218
11.19	测试 HTML5 语言实现的视频播放器	221
11.20	在 HTML5 的画布元素上进行绘画操作	222
11.21	操作 HTML5 存储对象	223
11.22	使用 Chrome 浏览器自动将文件下载到指定路径	225
11.23	修改 Chrome 设置伪装成手机 M 站	226
11.24	屏蔽 Chrome 的--ignore-certificate-errors 提示及禁用扩展插件并实现窗口最大化	228
11.25	禁用 Chrome 浏览器的 PDF 和 Flash 插件	229
11.26	禁用 IE 的保护模式	230
11.27	启动 Firefox 的同时打开 Firebug	231
11.28	禁用 Chrome 浏览器中的 Image 加载	233
11.29	禁用 Firefox 浏览器中的 CSS、Flash 及 Image 加载	234

第三篇 自动化测试框架搭建篇

第 12 章	数据驱动测试	237
12.1	数据驱动概述	237
12.2	数据驱动单元测试的环境准备	237
12.3	使用 unittest 和 ddt 进行数据驱动	238
12.4	使用数据文件进行数据驱动	240
12.5	使用 Excel 进行数据驱动测试	247
12.6	使用 XML 进行数据驱动测试	250
12.7	使用 MySQL 数据库进行数据驱动测试	254
第 13 章	行为驱动测试	261
13.1	行为驱动开发和 lettuce 简介	261
13.2	行为驱动测试的环境准备	262
13.3	第一个英文语言行为驱动测试	263
13.4	通过类模式实现英文行为驱动	266
13.5	lettuce 框架的步骤数据表格	269
13.6	使用 WebDriver 进行英文语言的行为数据驱动测试	272
13.7	使用 WebDriver 进行中文语言的行为数据驱动测试	275
13.8	批量执行行为驱动用例集	279
13.9	解决中文描述的场景输出到控制台乱码	283
第 14 章	Selenium Grid 的使用	285
14.1	Selenium Grid 简介	285



14.2	分布式自动化测试环境准备	286
14.3	Selenium Grid 的使用方法	290
14.3.1	远程调用 Firefox 浏览器进行自动化测试	290
14.3.2	远程调用 IE 浏览器进行自动化测试	294
14.3.3	远程调用 Chrome 浏览器进行自动化测试	295
14.3.4	同时支持多个浏览器进行自动化测试	296
14.4	结合 unittest 完成分布式自动化测试	297
14.5	实现并发的分布式自动化测试	299
第 15 章	自动化测试框架的搭建及测试实战	303
15.1	自动化测试框架	303
15.2	数据驱动框架及实战	305
15.3	关键字驱动框架及实战	333
15.4	关键字 && 数据混合驱动框架及实战	367
第四篇 常见问题和解决方法		
第 16 章	自动化测试常见问题和解决方法	405
16.1	如何让 WebDriver 支持 IE 11	405
16.2	解决“Unexpected error launching Internet Explorer. Browserzoom level was set to 75%(或其他百分比)”的错误	407
16.3	解决某些 IE 浏览器中输入数字和英文特别慢的问题	407
16.4	解决 Firefox 浏览器的 can't access dead object 异常	408
16.5	常见异常和解决方法	409

第一篇 基础篇

第 1 章 Selenium 简介

Selenium 工具诞生的时间已经超过 10 年,目前在软件开发公司中已得到大规模的应用,但是很少有人能够清楚地描述此工具的发展历史和特点,通过本章的介绍让读者和 Selenium 工具来一次亲密的接触,以便了解它的前世今生及其特点。

1.1 Selenium 的前世今生

2004 年在 ThoughtWorks 公司,一个名为 Jason Huggins 的兄弟为了减少手工测试的工作量,自己实现了一套基于 JavaScript 的代码库,使用这套代码库可以进行页面的交互操作,并且可以重复地在不同浏览器上进行各种测试操作。通过不断的改进和优化,这个代码库逐步成为 Selenium Core。Selenium Core 为 Selenium Remote Control (RC)和 Selenium IDE 提供了坚实的核心基础能力。

当时的自动化测试工具比较稀少,现有的工具也无法灵活地支持各种复杂的测试操作,大部分测试人员只能使用手工的方式完成 Web 产品的测试工作。开发人员不断地开发代码,测试人员不断地发现 bug,开发人员不断地修改 bug,测试人员不断地回归测试,以确认 bug 是否被修正,并且确认程序没有引入新的 bug。这样的产品开发模式,导致测试人员必须经常手工回归测试系统的大部分功能,由此产生了大量的重复性手工操作。Jason Huggins 想改变这样的现状,所以他开发了基于 JavaScript 的代码库,希望帮助测试人员从日常的重复性工作中解脱出来,经过不断的努力,Selenium 1.0 版本诞生了。

Web 自动化测试工具 Selenium 是跨时代的,因为它允许测试工程师使用多种开发语言来控制不同类型的浏览器,从而实现不同的测试目标。Selenium 是开源工具软件,用户无须付费就可以使用它,甚至可以根据自己的使用需求来进行深入的定制化,改写其原有的一些功能。基于以上这些优点,越来越多的测试人员开始使用此工具来进行 Web 系统的自动化测试工作。在短短几年时间内,全世界范围内都出现了 Selenium 工具的忠实拥护者,目前中国的几大互联网公司均使用 Selenium 作为 Web 自动化测试实施的主要工具。

但是随着互联网技术的不断发展以及浏览器对于 JavaScript 语言的安全限制,Selenium 的发展也遇到很多难以解决的困难。由于其自身实现的机制,Selenium 无法突破浏览器沙盒的限制,导致很多测试场景的测试需求难以被实现。

2006 年,Google 的工程师 Simon Stewart 开启了一个叫做 WebDriver 的项目,此项目可以直接让测试工具调用浏览器和操作系统本身提供的内置方法,以此来绕过 JavaScript

环境的沙盒限制, WebDriver 项目的目标就是为了解决 Selenium 的痛处, 并且也做到了。2008 年北京奥运会胜利召开的那年, Selenium 和 WebDriver 这两个项目进行了合并, 至此 Selenium 2.0 出现了, 也就是我们现在常常看到的 Selenium WebDriver(简称 WebDriver)。

Selenium2 = Selenium1 + WebDriver

Selenium 的官网地址是 www.seleniumhq.org, 网站提供了 Selenium WebDriver 的安装文件和使用教程。Selenium2 是 Selenium 1 的升级版本, 它本身向下兼容 Selenium 1 的所有功能, 同时又提供了更多新 API 来完成自动化测试的各种复杂需求。现阶段, Selenium1 已经退出历史舞台, 大部分 Web 自动化测试人员已经完全转向使用 Selenium 2 (WebDriver) 来搭建自己的自动化测试框架。

2016 年 10 月, Selenium 3 诞生。开发者在 Selenium 2 的基础上做了很多了不起的工作, 这个版本中很多新特性, 主要实现了把核心 API 跟客户端 driver 进行分离, 同时去掉用得越来越少的 Selenium RC 功能。为了迎合历史的发展潮流, 本书全部的案例均基于 Windows 7 操作系统上的 Selenium 3 的 WebDriver 的 API 进行讲解。

1.2 Selenium 工具套件介绍

- Selenium 2 (Selenium WebDriver): 提供了极佳的特性, 例如, 面向对象 API, 同时提供 Selenium 1 的接口用于向下兼容。
- Selenium 1 (Selenium RC 或 Remote Control): 支持更多的浏览器, 支持更多的编程语言 (Java、JavaScript、Ruby、PHP、Python、Perl、C#)。
- Selenium IDE(集成开发环境): Firefox 插件, 提供图形界面来录制和回放脚本。但此插件只是用来做原型的工具, 此插件需要使用第三方的 JavaScript 代码库才能支持循环和条件判断, 并不希望测试工程师使用此工具来运行大批量的测试脚本。
- Selenium -Grid 可以在多个测试环境以并发的方式执行测试脚本, 实现测试脚本的并发执行, 缩短大量测试脚本的执行时间。

1.3 Selenium 支持的浏览器和平台

Selenium 的一大特点就是能够在多种操作系统上支持多种浏览器的自动化测试, 通过下面的章节我们可以了解到此工具能够支持的操作系统和浏览器类型列表。

1.3.1 Selenium IDE、Selenium 1 和 Selenium RC 支持的浏览器和平台

表 1-1 列出了 Selenium IDE、Selenium 1 和 Selenium RC 支持的浏览器和操作系统。

表 1-1

浏览器	Selenium IDE	Selenium 1.0 (即 Selenium RC)	操作系统
Firefox 3. x	录制脚本和回放脚本	启动浏览器 运行测试脚本	Windows, Linux, Mac