

自动化测试 主流工具入门与提高

51Testing 软件测试网◎组编

51Testing 教研团队◎编著

51Testing 教研团队倾力打造，讲述测试人员应该掌握的 Selenium、JMeter、Appium 等重要测试工具，有助于读者迅速提升测试效率

自动化测试 主流工具 入门与提高

51Testing 软件测试网◎组编

51Testing 教研团队◎编著

人民邮电出版社

北京

图书在版编目 (CIP) 数据

自动化测试主流工具入门与提高 / 51Testing软件测试网组编 ; 51Testing教研团队编著. -- 北京 : 人民邮电出版社, 2020. 4
ISBN 978-7-115-52578-9

I. ①自… II. ①5… ②5… III. ①软件工具—自动检测 IV. ①TP311.5

中国版本图书馆CIP数据核字(2019)第250642号

内 容 提 要

本书共5章,分别讲解了开源的Web自动化测试工具Selenium,基于Java的压力和接口测试工具JMeter,单元测试中的JUnit测试框架和JMock工具,用于移动端的自动化测试工具Appium,以及Appium测试框架的搭建。

本书适合测试人员和开发人员阅读,也可供相关专业人士参考。

-
- ◆ 组 编 51Testing 软件测试网
编 著 51Testing 教研团队
责任编辑 谢晓芳
责任印制 王 郁 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
山东百润本色印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 13.5
字数: 247千字 2020年4月第1版
印数: 1-2400册 2020年4月山东第1次印刷
-

定价: 55.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

前 言

为什么写本书

当我们熟悉了手工测试之后，为了提升测试效率，就会需要使用自动化测试工具来代替人进行测试。因此，本书针对目前测试工作中较常见的 Web 测试和 APP 测试分别讲解了 Web 的自动化测试与性能测试，以及 APP 的自动化测试。

本书内容

本书共 5 章。每一章的主要内容如下。

第 1 章介绍了一款广泛使用的开源 Web 自动化测试工具——Selenium。同时，该章还详细讲解了 Selenium 的录制工具 Selenium IDE、WebDriver 实例开发和 JUnit 框架。

第 2 章介绍了基于 Java 的压力和接口测试工具——JMeter。JMeter 也是一款开源的工具，并且能够实现收费的性能测试工具 LoadRunner 95%以上的功能。因此，对于需要进行性能测试的中小型软件公司来说，JMeter 是一个十分理想的选择。

第 3 章讲述了单元测试中面向对象的概念，针对 JUnit 测试框架在 Selenium 的基础上又进一步给出了详细的实例。该章还讨论了 JMock 驱动和桩，并且通过实例来演示 Mock 对象的使用。

第 4 章讨论了一款针对 iOS 和 Android 平台的自动化测试工具——Appium。Appium 支持 Python 和 Java，非常实用，是移动端的主流自动化测试工具。该章是基于 Python 进行讲述的。该章详细介绍了 Appium 环境的搭建、元素的定位、Appium 常用操作和 yaml。

第 5 章阐述了 Appium 测试框架的搭建，主要内容包括框架整体说明、Logging 模块、PageObject 设计模式、框架的实现。

读者对象

本书适合软件测试人员和开发人员阅读，要求读者具有软件测试的基础知识并且能

够熟练编写手工测试用例。

学习成果

通过对本书的学习，读者不仅可以掌握 Web UI 自动化测试和性能测试的方法，还可以精通 APP 自动化测试脚本的开发和测试框架的使用。

作者简介

51Testing 软件测试网是专业的软件测试服务供应商，为上海博为峰软件技术股份有限公司旗下品牌，是国内人气非常高的软件测试门户网站。51Testing 软件测试网始终坚持以专业技术为核心，专注于软件测试领域，自主研发软件测试工具，为客户提供全球领先的软件测试整体解决方案，为行业培养优秀的软件测试人才，并提供开放式的公益软件测试交流平台。51Testing 软件测试网的微信公众号是“atstudy51”。

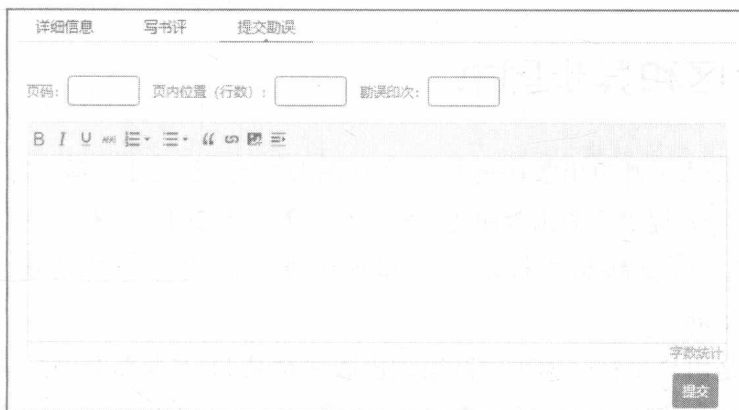
服务与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供后续服务。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，单击“提交勘误”按钮，输入勘误信息，单击“提交”按钮即可，如下图所示。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您所在的学校、培训机构或企业想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接通过邮件发送给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

目 录

第 1 章 Selenium	1	1.5.1 JUnit 4	30
1.1 准备软件	1	1.5.2 在 Eclipse 中 JUnit 4 的 使用	31
1.2 Selenium 简介	1	1.5.3 通过模拟鼠标移动显示悬浮的 下拉窗体的实例	37
1.2.1 主要功能	2	1.5.4 模拟鼠标单击事件	39
1.2.2 各版本和系统之间的 关联	2	1.5.5 使用 javadoc 进行查找	40
1.3 Selenium IDE 的使用	3	第 2 章 JMeter	44
1.3.1 Selenium IDE 的安装	3	2.1 性能测试基础	44
1.3.2 Selenium IDE	4	2.1.1 性能的定义	44
1.3.3 Selenium IDE 入门实例	7	2.1.2 性能测试的概念	45
1.4 Selenium WebDriver	11	2.1.3 性能测试的分类	45
1.4.1 Selenium WebDriver 简介	11	2.1.4 性能指标	48
1.4.2 Selenium WebDriver 的安装 和配置	12	2.1.5 性能测试技术要求	49
1.4.3 Selenium WebDriver 入门 实例	16	2.2 JMeter 概述	49
1.4.4 javadoc 简介	19	2.3 搭建 JMeter 环境	51
1.4.5 WebDriver 元素的定位	20	2.4 JMeter 目录结构	53
1.4.6 get()方法的实例	22	2.5 JMeter 的测试计划及常用元件	53
1.4.7 关于浏览器的操作 方法	23	2.5.1 测试计划	53
1.4.8 弹出窗口的切换方法	24	2.5.2 线程（用户）	54
1.4.9 多个元素的选择	27	2.5.3 测试片段	55
1.4.10 单个元素的选择	28	2.5.4 控制器	56
1.5 JUnit 框架与 WebDriver	30	2.5.5 配置元件	57
		2.5.6 定时器	58
		2.5.7 前置处理器	58
		2.5.8 后置处理器	59

2.5.9 断言	59	2.16 实例 3: 使用 JMeter 创建 JDBC 测试计划	105
2.5.10 监听器	60	第 3 章 单元测试	111
2.6 脚本录制方法	61	3.1 面向对象编程	111
2.6.1 使用 Badboy 录制	61	3.1.1 什么是面向对象	111
2.6.2 使用 JMeter 内置的代理服务器 录制	64	3.1.2 类与实例	113
2.7 JMeter 中元件的作用域与 执行顺序	70	3.1.3 继承	115
2.8 JMeter 的参数化设置	72	3.1.4 接口	116
2.8.1 通过添加前置处理器 参数化	72	3.1.5 多态	118
2.8.2 通过 CSV Data Set Config 参数化	77	3.2 准备被测程序	120
2.8.3 借助函数助手随机 参数化	79	3.2.1 被测程序的功能	121
2.9 设置 JMeter 集合点	80	3.2.2 程序概要设计	121
2.10 设置 JMeter 检查点	82	3.2.3 程序代码实现	122
2.10.1 添加内容检查断言	82	3.2.4 开发测试代码	127
2.10.2 添加断言持续时间	84	3.3 JUnit 测试框架	129
2.10.3 设置断言结果大小	86	3.3.1 在 Eclipse 中配置 JUnit	130
2.11 设置 JMeter 关联	87	3.3.2 使用 JUnit 进行测试	131
2.12 JMeter 常用监听器	92	3.3.3 JUnit 断言机制	133
2.12.1 “图形结果”监听器	92	3.3.4 JUnit 各类注解	135
2.12.2 “查看结果树”监听器	93	3.3.5 JUnit 假设机制	137
2.12.3 “聚合报告”监听器	94	3.3.6 JUnit 参数化	138
2.12.4 Summary Report 监听器	94	3.3.7 JUnit 测试集	140
2.13 在非 GUI 模式下运行 JMeter	95	3.4 JMock 测试框架	141
2.14 实例 1: 使用 JMeter 创建 Web 测试计划	97	3.4.1 驱动和桩	141
2.15 实例 2: 使用 JMeter 创建 Web Service 测试计划	101	3.4.2 Mock 对象	142
		3.4.3 JMock 的特性	145
		3.4.4 使用 JMock 模拟 isNumber 方法	145
		3.4.5 使用 JMock 模拟类	147

第 4 章 Appium 开发	150	4.3 Appium 常用操作	181
4.1 搭建 Appium 环境	150	4.3.1 等待元素出现	181
4.1.1 环境准备	150	4.3.2 toast 元素的定位	181
4.1.2 安装 JDK	150	4.3.3 Appium 屏幕截图	182
4.1.3 下载与安装		4.3.4 WebView 定位	183
Android SDK	153	4.3.5 swipe 方法	186
4.1.4 添加 Android SDK 环境		4.3.6 手势定位	189
变量	155	4.4 yaml	191
4.1.5 连接夜神模拟器	156	4.4.1 yaml 支持的数据类型	191
4.1.6 安装 Node.js	157	4.4.2 读取 yaml 数据	192
4.1.7 安装 Python	158	4.4.3 配置 yaml	192
4.1.8 安装 Appium-desktop	159	第 5 章 搭建 Appium 测试框架	196
4.1.9 安装 .NET Framework	160	5.1 准备软件	196
4.1.10 检查 Appium 环境		5.2 框架整体说明	196
设置	161	5.2.1 Appium 框架的组成	196
4.1.11 安装 Appium-Python-		5.2.2 框架实现说明	197
Client	161	5.3 Logging 模块	197
4.1.12 第一个脚本	162	5.3.1 日志的级别	197
4.1.13 Desired Capabilities	168	5.3.2 Logging 模块的组成	197
4.2 定位元素	172	5.3.3 使用 Logging 模块过滤	
4.2.1 使用 Appium Inspector 定位		输出日志	198
元素	172	5.4 PageObject 设计模式	198
4.2.2 使用 UI Automator Viewer		5.4.1 PageObject 设计模式存在的	
定位元素	173	问题及解决方案	198
4.2.3 使用 id 定位元素	176	5.4.2 基于 PageObject 设计模式	
4.2.4 使用 Appium Inspector 中的		封装架构	198
xpath 定位元素	176	5.5 实现框架	199
4.2.5 使用 id 和 text 定位		5.5.1 建立项目文件夹	199
元素	177	5.5.2 在 base_view 下封装常用	
4.2.6 使用 List 定位元素	178	方法	199

5.5.3 封装常用元素和业务逻辑	200	5.5.6 批量生成报告	204
5.5.4 对测试数据进行封装	203	5.5.7 以批处理方式执行测试	205
5.5.5 对测试用例进行封装（以登录功能为例）	203	5.5.8 持续集成（以 Jenkins 为例）	205

第 1 章 Selenium

Selenium 是 ThoughtWorks 专门为 Web 应用程序编写的一个验收测试工具。Selenium 支持的浏览器包括 IE、Firefox、Safari 等。框架底层使用 JavaScript 模拟真实用户对浏览器进行操作。

1.1 准备软件

本节介绍要准备的软件。

Selenium IDE 包括以下内容：

- firebug-1.12.8-fx.xpi;
- Firefox_Setup_25.0.1_chs.exe;
- selenium-ide-2.6.0.xpi。

WebDriver 包括以下内容：

- chromedriver.exe;
- geckodriver.exe;
- eclipse.zip;
- jdk-8u121-windows-x64.exe.exe;
- selenium-server-standalone-3.3.1.jar（可从 Selenium 官网下载最新版本）。

帮助文档是 selenium_javadoc。

被测环境是 WAMP+Discuz_X2_SC_UTF8 论坛（可从 WAMP 官网和 Discuz 官网下载）。

1.2 Selenium 简介

下面对 Selenium 的主要功能及版本进行介绍。

1.2.1 主要功能

Selenium 的主要功能如下。

- 测试与浏览器的兼容性——测试应用程序是否能够在不同浏览器和操作系统上正常运行。
- 测试系统功能——创建回归测试以检验软件功能和用户需求，支持自动录制动作和自动生成.NET、Java、Perl 等语言的测试脚本。

1.2.2 各版本和系统之间的关联

Selenium 1.0 是一套完整的 Web 应用程序测试系统，可用于测试的录制（Selenium IDE 负责）、编写与运行（Selenium Remote Control 负责），以及测试的并行处理（Selenium Grid 负责）。Selenium 的核心——Selenium Core 基于 JsUnit，完全由 JavaScript 编写，因此可运行于任何支持 JavaScript 的浏览器上。Selenium 1.0 的原理如图 1-1 所示。

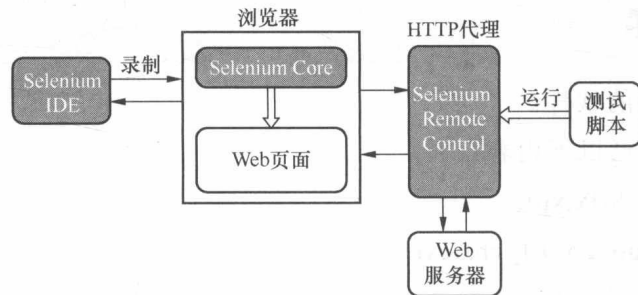


图 1-1 Selenium 1.0 的原理

Selenium 1.0 中组件的功能如下。

- Selenium IDE: Firefox 的附加组件，结合 Firefox 不但可以录制测试脚本，回放脚本，而且可以生成一些基于 Selenium Remote Control 模式的简单代码。
- Selenium Core: 整个测试机制的核心部分，即有断言（assertion）机制的测试套件运行器（test suite runner）。它由一些纯 JavaScript 代码组成，可以运行在 Windows/Linux 系统下的不同浏览器中。
- Selenium Remote Control: 一个代理与控制端，可代替 Selenium Core/ Selenium IDE 的客户端（相当于通过编程来实现一切），支持多种语言。

和 Selenium 一起出现的还有 WebDriver。WebDriver 和 Selenium 本是两个独立的项目，实现机制也是不同的，但是 Selenium 团队在 Selenium 2.0 中将两者合并，将其命名为 WebDriver。Selenium 2.0 的主要新功能是集成了 Selenium 1.0 及 WebDriver。也就是

说，Selenium 2.0 兼容 Selenium 1.0，它既支持 Selenium API，也支持 WebDriver API。Selenium 2.0 的原理如图 1-2 所示。

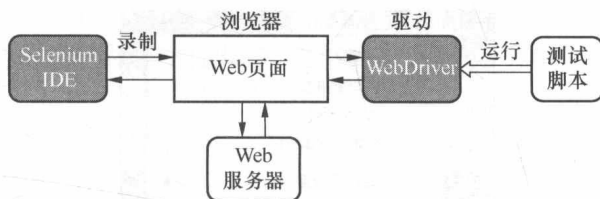


图 1-2 Selenium 2.0 的原理

1.3 Selenium IDE 的使用

Selenium IDE 是 Selenium 的图形化录制、回放工具，虽然易于使用，但是由于只能在 Firefox 浏览器上使用，局限性较大，因此近几年来用得越来越少，逐步被 WebDriver 替代。这里只对其做简单介绍。

1.3.1 Selenium IDE 的安装

Selenium IDE 的安装步骤如下。

- (1) 下载 Firefox 浏览器，并安装。
- (2) 在 Firefox 的附加组件中，搜索 Selenium IDE，并安装。

如没有找到相关的组件，则可以从 Selenium 或 Firefox 官网下载。从 Firefox 官网下载 Selenium IDE 插件的方法如图 1-3 所示。

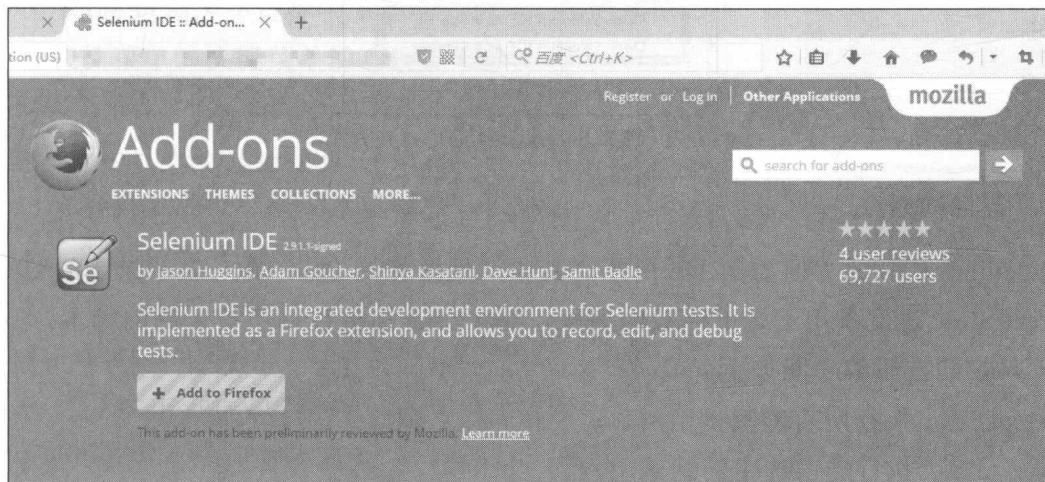


图 1-3 从 Firefox 官网下载 Selenium IDE 插件的方法

(3) 重启 Firefox 浏览器，完成安装。在 Firefox 浏览器的“工具”菜单中可以找到 Selenium IDE，如图 1-4 所示。

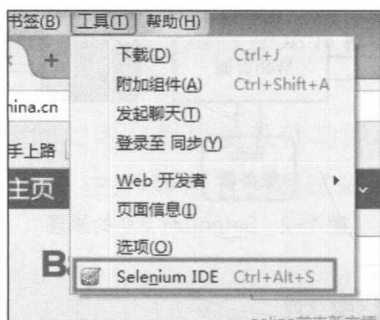


图 1-4 选择“工具”→Selenium IDE

1.3.2 Selenium IDE

选择 Firefox 浏览器的“工具”菜单中的 Selenium IDE，打开 Selenium IDE。如果安装了 Selenium IDE Button 插件，则还能选择以边栏或弹出窗口来打开 Selenium IDE，如图 1-5 所示。

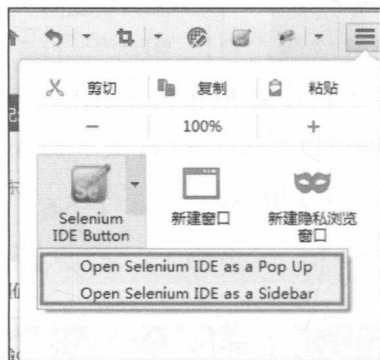








图 1-5 选择 Selenium IDE 的打开方式

1. 界面的组成

Selenium IDE 的界面如图 1-6 所示。其中，地址栏的 Base URL 表示录制的 URL 地址；Fast/Slow 进度条用于调节回放的速度。Test Case 面板包括所有录制的脚本。

工具栏的按钮的功能如下。

- : 执行所有的测试用例。
- : 执行当前选中的测试用例。
- : 暂停/恢复。

- : 单步执行选中用例的命令，常用于命令调试中。
- : 允许一连串的 Selenium 命令组合成一个动作。
- : “录制”按钮，Selenium IDE 打开后默认处于录制状态，单击该按钮则取消录制。单击浏览器中的链接，在 Table 下就会产生一条命令。

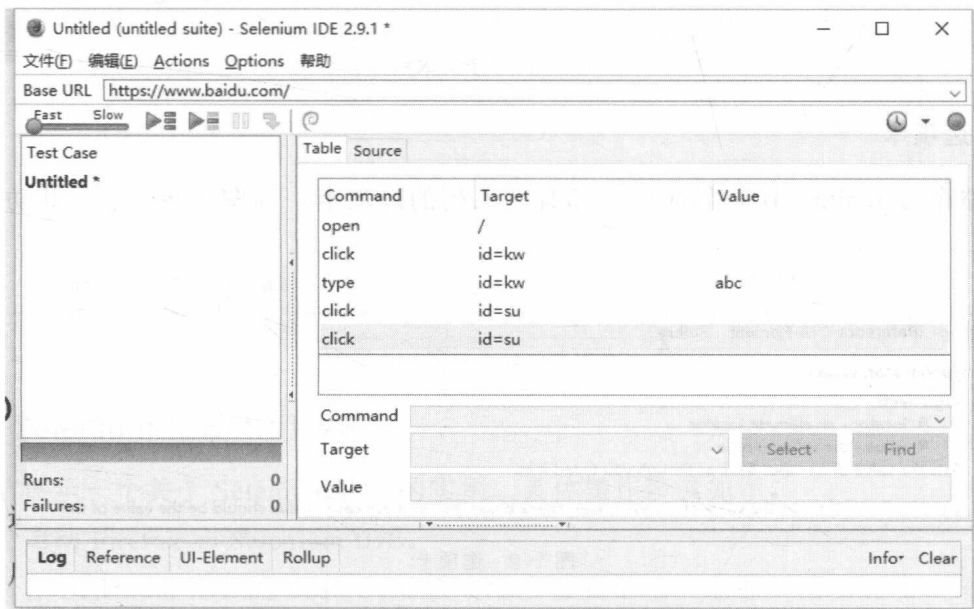


图 1-6 Selenium IDE 的界面

2. 快捷菜单

在 Table 选项卡中选择一条命令并右击，会出现图 1-7 所示的快捷菜单。快捷菜单的功能如下。

- **Insert New Command:** 插入命令。
- **Insert New Comment:** 插入注释（移动命令或注释只要用鼠标拖动就可以了）。
- **Toggle Breakpoint:** 设置断点，调试。要取消断点，再次选择该项即可。
- **Set/Clear Start Point:** 设置/清空起点，调试。
- **Execute this command:** 执行命令。

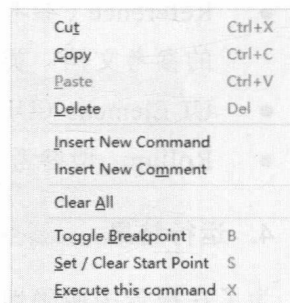


图 1-7 快捷菜单

Table 选项卡和 Source 选项卡可以切换，使用 Source 选项卡可以查看 XML 源文件。Table 选项卡中有 3 个文本框，选中 Table 选项卡中的命令后，可以对命令进行编辑。

如图 1-8 所示，在 Value 文本框中输入了 abc。选中 type 命令，则 Command 文本框中显示 type，操作的对象是一个 id 为 kw 的文本框，Value 为 abc。



图 1-8 3 个文本框

3. 选项卡

在整个 Selenium IDE 界面的下部有一系列的选项卡，如图 1-9 所示。其功能描述如下。

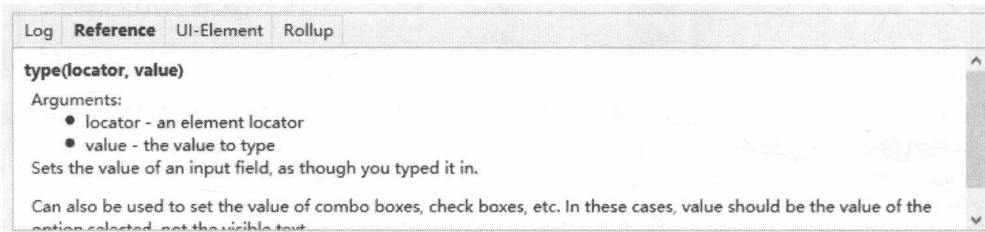


图 1-9 选项卡

- Log (日志): 当运行测试时，错误和消息将会显示在这里。右边有 Info 按钮和 Clear 按钮，单击 Info 按钮可以选择显示日志的级别，单击 Clear 按钮可以清除日志。
- Reference (参考): 当在表格中输入和编辑 selenese 命令时，面板中会显示相应的参考文档，类似于快速帮助的内容。
- UI-Element (UI 元素): 参考“帮助”菜单下的 UI-Element Documentation。
- Rollup: 请参考“帮助”菜单。

4. 运行结果

为了在报告中明确地展示结果，Selenium IDE 会使用颜色来标记执行结果。执行过后，测试用例将会被标注为“红色”或“绿色”。

- 红色表示测试用例运行失败。
- 绿色表示测试用例成功运行。

运行结果还显示了所有的执行测试用例数量与失败的测试用例的数量（左下角）。