

全面深入自动化测试技术，包括接口自动化测试、App自动化测试、性能自动化测试、Web自动化测试

软件自动化 测试开发

Java和Python
测试开发实用指南

邹辉 编著

软件自动化 测试开发

—— 邹辉 编著 ——



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书全面深入地介绍了软件自动化测试开发领域方方面面的相关知识，包括 App 功能自动化测试的方案、环境、代码运行及报告（基于 Appium 工具和 Java 语言编写），API 接口自动化测试的方案、环境、代码运行及报告（基于 Zentao 工具和 Python 脚本语言编写），Selenium 的 Web 自动化测试（基于 Selenium 工具和 Java 语言编写），JMeter 接口和性能测试，LoadRunner 性能测试以及 Jenkins 持续集成。

本书适用于想要了解、学习和使用当前流行的自动化测试开发技术的广大开发和测试从业人员，以及产品测试开发 leader 等。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

软件自动化测试开发 / 邹辉编著. —北京：电子工业出版社，2017.1
ISBN 978-7-121-30453-8

I. ①软… II. ①邹… III. ①软件—测试—自动化②软件开发 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2016)第 283010 号

责任编辑：安 娜

印 刷：三河市鑫金马印装有限公司

装 订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：17 字数：258 千字

版 次：2017 年 1 月第 1 版

印 次：2017 年 1 月第 1 次印刷

定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819 faq@phei.com.cn。

前言

关于本书

互联网软件技术发展速度非常快，稍不留神就“out”了，然而测试思路、开发语言却是有历史承传的。只有多实践打好基础，知其然并知其所以然，方可以不变应万变。我们上学是从幼儿园、小学、中学……到博士后一级一级上，学习技术也不例外，因此具备快速学习的能力十分重要。人们对美好先进的东西永不会满足，追求永无止境，所以人生就是一个永远自强不息的成长过程。

本书用详细、完整的案例，完整的代码以及理论知识引领读者进入实际操作。当读者实践一遍之后，就能看到真实效果，并且印象深刻。就像拿到驾照后终身不用再考，而且能够驾驶准驾车型的各种车一样，二者的原理相同。这样我们就能花最小的代价学到最实实在在的自动化测试开发技术。

本书全面地深入讲解地自动化测试开发技术，包括接口自动化测试、App 自动化测试、Web 自动化测试和性能自动化测试，同时还讲解了最新、最流行的自动化技术：移动 Appium 功能自动化、API 后台接口自动化、LoadRunner 性能测试、Selenium 的 Web 自动化以及 JMeter 自动化技术等。

本书基于 Java 语言和 Python 语言编写，结合各种主流开源工具框架，能够真正地提升测试技术水平。掌握一门开发语言是极为必要的，读者可着重学习书中完整

的 Java 和 Python 源代码和解析，提升薄弱环节。源代码对于实战来说十分重要，因此书中代码部分都加上了注释和解析，以供读者理解和学习。写代码无疑是测试同行较为薄弱的部分，但是当把看代码、写代码当成一种习惯或一种乐趣时，学习代码自然就能得心应手。

本书实践与理论相结合，包括方案、环境、代码和运行报告。书中的源码在保留版权的情况下供读者使用，也就是说，读者使用源码时需要注明出自本书。

本书需要读者有较好的基础和耐心，以及领悟力。对初学者而言，可以照葫芦画瓢，在实践中激发兴趣和信心，对有基础的读者而言，可以更加深入地了解自动化测试，并直接应用到工作当中，本书的内容将起到一个实用指南的作用。有的人擅于自己摸索着学习，有的人喜欢在书本的引导下学习，也有的人需要通过培训来学习，这都没关系，俗话说：不管黑猫白猫，抓到老鼠就是好猫。

关于软件自动化测试开发

除手工测试外，其他都是自动化测试。因此，软件自动化测试开发指的是测试技术和开发技术相结合，用手动测试之外的测试技术，模拟手动用户场景测试的实现过程，简单来说，就是用写的代码来测试程序。

在工作中，手动功能测试人员通常会感觉自己比其他如开发、产品等人员相对弱势，因为功能测试一般被认为技术含量不高，准入条件较低，甚至被认为工作就是单击页面，单击按钮。通常手动功能测试人员只需 3~6 个月左右的专业培训就足以胜任工作。

自动化测试开发工作相对测试而言，技术含量较高，相关人员的待遇也相对较高，他们与广大程序员的性质相同，是测试从业人员追求技术进步的表现。一般来说，自动化测试开发需要 1~3 年左右的专业培养方能胜任工作。

适合读者

绝大多数适用的读者

- 所有软件测试从业人员，包括测试团队 leader。
- 有一定开发语言基础的测试人员。

- 软件测试专业的学霸。

少部分适用的读者

- 软件测试专业的在校大学生。
- 软件技术、移动互联网相关人员，包括开发人员，研发团队 leader 等。
- 其他任何对软件自动化测试开发感兴趣的人。

作者简介

本书作者有丰富的软件测试从业经验，擅长开发和测试技术，担任测试主管，以电子商务、银行证券、移动互联网为主要业务领域，现在负责一上市企业 O2O 产品的软件自动化测试开发工作。

大纲内容

第 1 章到第 4 章：介绍 App 功能自动化测试方案、环境、代码、运行报告。

第 5 章到第 7 章：介绍 API 接口自动化测试方案、环境、代码、运行报告。

第 8 章：介绍 Selenium 的 Web 自动化测试。

第 9 章：介绍 JMeter 接口和性能测试。

第 10 章：介绍 LoadRunner 性能测试。

第 11 章：介绍 Jenkins 持续集成。

附录 A-D：小知识参考。

前言后记：“鸡汤”。

致谢

感谢自己业余时间咖啡厅辛勤地写完此书，感谢我的家人，感谢我的朋友们。

感谢之前相对比较默契的同事们，因为和大家共同的工作经历和成长才得来本书内容的酝酿——分别有测试经理蔡祥；走秀网 PM 吴盛幸，开发劳水生，测试经理

曾春刚；腾讯开发郑双明；华南城华盛电子商务 CTO 阳志军；充电网朱舜、邓明剑、刘东亮等。

最最感谢的是对本书感兴趣的你——读者朋友们。

关于勘误

虽然书中的每个技术点都曾在实际项目中实践和应用过，但也会因为我们个人技术、所测项目和视野的局限，以及本人因时间仓促和能力水平等种种原因，书中难免会有一些错误和纰漏，如果大家在阅读过程中发现了什么问题，恳请反馈给我，读者朋友们可即时在线交流，联系方式如下。

作者微信和 QQ 号：zouhui1003it, 7980068

测试博客：<http://www.cnblogs.com/finer>

读者实战 QQ 互动群：377029807

微信公众号：测试开发社区

目 录

第 1 章 App 自动化测试方案.....	1
1.1 概述.....	2
1.2 风险分析.....	2
1.3 软硬件需求.....	3
1.4 测试计划.....	3
1.5 Appium 移动自动化框架.....	4
1.6 测试框架.....	10
1.7 自动编译部署工具.....	11
第 2 章 Android 自动化环境搭建.....	14
2.1 Android 搭建的简要步骤.....	15
2.2 在 Windows 上搭建 Android 自动化环境.....	16
2.3 在 Mac 机器上搭建 Android 自动化环境.....	28
2.4 Android 自动化测试运行.....	29
第 3 章 iOS 自动化环境搭建.....	30
3.1 iOS 环境搭建的简要步骤.....	31
3.2 iOS 自动化环境搭建的详细步骤.....	31
3.3 iOS 自动化测试运行.....	38
3.4 iOS 的 App 自动化测试 demo 演示视频.....	38

第 4 章 App 自动化测试源代码	39
4.1 基于 Java 的 App 自动化源代码解析	40
4.2 源代码结合 Ant 持续集成到 Jenkins	71
4.3 Android 和 iOS 自动化测试结果展示	73
第 5 章 API 接口自动化测试方案	75
5.1 概述	76
5.2 所用技术点	78
5.3 主要功能	78
5.4 测试计划	79
第 6 章 API 接口自动化环境搭建	80
6.1 Python 环境准备	81
6.2 Zentao (禅道) 项目管理工具	83
6.3 MySQL 数据库	84
6.4 Fiddler 接口抓包工具	86
6.5 Postman 接口测试工具	93
第 7 章 API 接口自动化源代码	96
7.1 基于 Python 的接口自动化脚本解析	97
7.2 Python 接口测试数据展示	147
7.3 脚本持续集成到 Jenkins	151
7.4 接口自动化测试报告	151
第 8 章 Selenium 的 Web 自动化测试	154
8.1 Selenium 自动化测试准备	155
8.2 Selenium 自动化源码解析	156
8.3 持续集成到 Jenkins	173
8.4 Web 自动化测试结果展示	174
第 9 章 JMeter 接口测试和性能测试	176
9.1 安装和介绍	177
9.2 Jmeter 接口测试示例	189

9.3	结合 Ant 持续集成到 Jenkins.....	195
9.4	接口测试结果	198
9.5	JMeter 性能测试示例	199
第 10 章	LoadRunner 性能测试	203
10.1	小概念	204
10.2	安装和破解	213
10.3	脚本调试	217
10.4	运行场景	221
10.5	性能监控	223
10.6	问题分析和调优	224
10.7	性能压力测试报告样例	226
第 11 章	Jenkins 持续集成.....	232
11.1	介绍	233
11.2	系统配置	233
11.3	项目配置	236
11.4	多机器节点配置.....	240
11.5	结果展示视图.....	243
附录 A	自动化管理平台和产品自动化系统.....	245
附录 B	Java 和 Python 开发语言学习历程	247
附录 C	常见错误和问题解答.....	252
附录 D	常用软件安装包链接.....	255
后记	259

第 1 章

App 自动化测试方案

1.1 概述

什么是 App 自动化？为什么要做 App 自动化？

App 自动化是指给 Android 或 iOS 上的软件应用程序做的自动化测试。

手工测试和自动化测试的对比如下。

手工测试优势：不可替代、能发现更多 bug、包含了人的想象力与理解力。

自动化测试优势：可重复、效率高，能增加对软件质量的信任度。

注意，不是所有功能都需要自动化，只需把重复执行的以及主要的交给自动化。

App 自动化测试的特点如下：

- 执行自动化测试只能发现一小部分 bug。
- 执行自动化冒烟测试或回归测试是用来验证系统状态，而不是找出更多 bug。
- 执行自动化测试可以让测试同事有更多的精力来关注复杂场景，做更多更深层次的测试。
- 编写自动化测试过程中会发现一部分 bug，发现后要及时记录。

1.2 风险分析

自动化测试的主要风险分析如下：

(1) 测试用例覆盖率（覆盖率决定了测试效率，因此要选择合适的用例，应约占功能用例集的 20%~50%）。

(2) 测试结果准确度（准确度决定了测试有效性，因此应尽可能减少误报）。

(3) 自动化代码维护（维护影响成本，本书写的是关键字驱动自动化框架，自动化框架代码应尽可能优化，所测的功能改动而代码不需要改动时才是强大的框架，维护成本才足够低）。

(4) 版本开发和测试时间进度（当项目需求和功能较为稳定时，建议用自动化测试）。

(5) 开发对控件元素增修改的程度（需开发人员尽可能地用 `name` 元素，并且和 UI 设计一致。当修改变动量较小时，测试人员可根据提供的元素提前介入，开发自动化脚本）。

App 源码权限控制，在 iOS 上测试时需要用到源码，我们测试人员可能只需要 SVN 下载权限，不需要上传权限，因此应尽可能地避免改动 SVN 上开发人员的源码。

1.3 软硬件需求

自动化测试的软硬件需求如下。

硬件：

- Mac 电脑、iPhone 手机。
- Windows 电脑、Android 手机。

软件：

- Appium 测试框架：运行 App 驱动的自动化平台，通过识别的控件元素，模拟用户的手工操作，支持 iOS 和 Android 系统。
- Jenkins：持续集成自动构建和执行任务。
- TestNG、Ant、SVN：测试插件初始化、测试、断言、清理。
- JDK、Eclipse、Java 语言开发编写环境。
- AdbWireless：安卓手机和电脑间的无线连接。

1.4 测试计划

时间计划

对于有良好代码基础的熟手，可用一周时间做出演示 demo。如果是从零开始的小白，则可用 3 到 6 个月的时间做出演示 demo。

对于有良好代码基础的熟手，可用一个月时间试运行冒烟测试用例。如果是从零开始的小白，则可用半年到一年的时间试运行测试冒烟测试用例。

目前 App 自动化框架计划

采用自动化关键字数据驱动模式设计，即表格驱动测试或者基于动作的测试。关键字驱动框架的基本工作是将测试用例分成四个部分：一是测试步骤，二是测试步骤中的对象，三是测试对象执行的动作（Action），四是测试对象需要的数据（Test Data）。

后期 App 自动化框架计划

把测试用例、控件元素等放入数据库或页面进行展示操作。做到写自动化测试用例完全不用增修改代码，而是由用例自动生成代码，自由管理大量用例和测试数据，最终做成自动化平台。这些内容需要一个测试开发团队来实现，可参见本书附录 A。

1.5 Appium 移动自动化框架

1. 需要掌握的技能（第 1 章到第 4 章都会用到）

- (1) Appium API、WebDriver 基础知识和环境搭建（见第 1.5 节）。
- (2) TestNG 等测试框架（见第 1.6 节）。
- (3) Android/iOS 开发测试基础以及环境搭建（见第 2 章和第 3 章）。
- (4) 开发移动自动化项目的 Java 语言或 Python 语言等（见第 4 章）。

需要说明的是，如果想用 Python 语言编写 App 自动化测试框架，那么读者可以参考本书的 Java 和 TestNG 的代码逻辑，使用 Python 和 unittest 编写测试脚本。Appium 自动化测试框架的功能概括如下：

2. Appium 框架的功能

- (1) 支持 iOS、Android，可在多台机器上并行 App 自动化，测试机型适配。

(2) 代码实现关键字驱动:

- 测试集: 关联 Excel 测试用例和脚本配置。
- 测试数据: Excel 存储输入数据、控件元素、测试结果。
- 测试脚本: 由 Java 和 TestNG 编写, 分层结构有 case、log、config、report 以及 data 等。

(3) 自动测试用例执行:

- 从功能测试用例中抽取需重复执行的、主要的功能进行用例覆盖。
- 支持用例 failed (失败) 时自动截屏。
- failed (失败) 用例自动重复执行数遍。

(4) 持续集成环境 Jenkins, 定时自动构建和执行测试任务。

- 测试结果报告展示, 自动邮件展示。

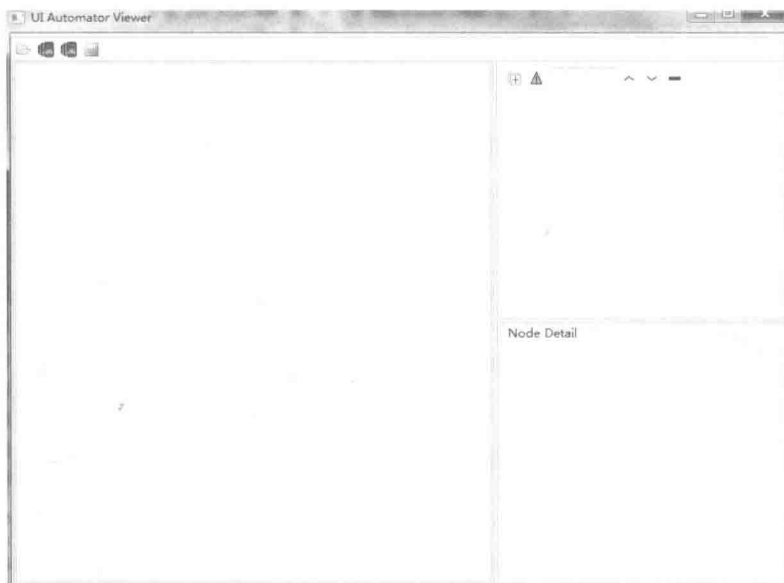
Appium 自动化测试一个 App 的基本过程如下:

3. 测试 App 的基本过程

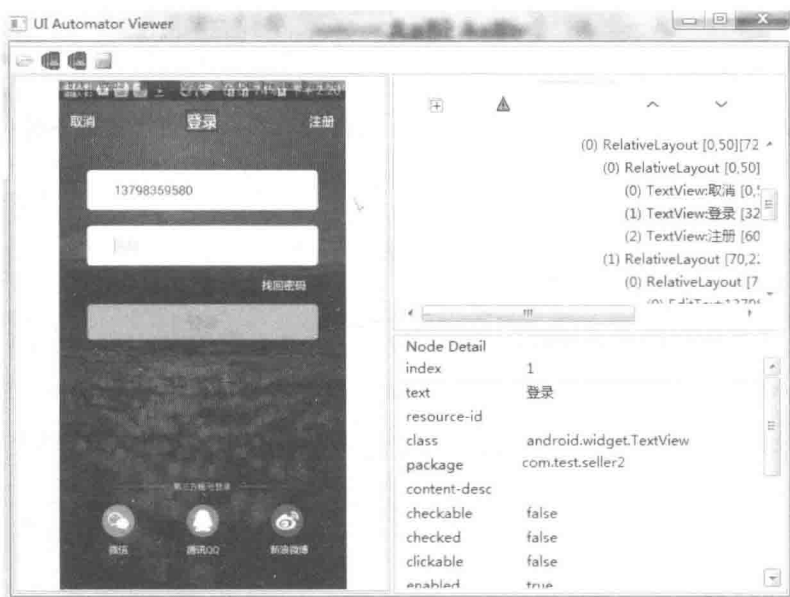
基于 Appium 自动化测试框架, 我们要进行的是连接电脑、连接手机、解锁、安装 App、卸载 App、启动 App、元素定位、元素的操作、屏幕的操作、页面等待、异常处理截图、数据校验、日志、报告等一系列自动化测试执行的详细过程。

Appium 自动化框架元素控件的捕获, 根据捕获到的元素控件进行相应的操作。

Appium 元素控件有多种定位方法, 最常用的是元素的 ID (即 By.id) 和元素的值 (即 By.name)。还可以通过元素类型 tagName、元素的位置 XPath、手机设备的坐标等进行定位操作。安卓的元素控件可以通过 SDK 中的 uiautomatorviewer.bat 文件进行录制和捕获定位, 如图 1.1 至图 1.3 所示。



▲图 1.1



▲图 1.2



▲图 1.3

如图 1.3 所示，Node Detail 下面的 resource-id `com.test.sellerid/phone_edit1` 对应 Excel 和代码中的定位方法 `By.id`，控件元素数据 `text 13798359580` 对应 Excel 和代码中的操作方法 `sendKeys()`，控件元素赋值数据为 `13798359580`。

可以这样理解：首先找到这个文本框，接着给这个文本框输入数据。即通过 ID 属性值 `com.test.sellerid/phone_edit1`，找到此用户名文本框的控件元素，然后通过 `sendKeys()` 方法输入用户名数据 `13798359580` 到此用户名文本。其他自动化测试步骤的定位方法、控件元素以及操作方法也都与此类似。实际上，自动化测试就是通过程序代码来实现模拟手动测试去操作一遍的过程。

上面介绍了用户名文本框输入用 `sendKeys()` 方法，那么其他元素的操作方法有哪些呢？元素操作方法大致有单击 (`click`)、输入 (`sendKeys`)、元素滑动、页面滑动、长按、下拉、弹出、屏幕放大缩小等，最常用的就是单击和输入。代码解析详见第 4 章中的源码。

数据校验。其实元素本身就是数据校验，当程序找不到元素时，用例就会失败。另外，测试用例时可以加入一个或多个断言进行验证数据，还可设置步骤等待延迟时间，详细内容见第 4 章讲解中的源码。

测试结果。测试用例中记录了运行后的测试结果，如 `pass`、`failed` 或是 `skip`，详细内容见第 4 章关联的 Excel 测试用例。