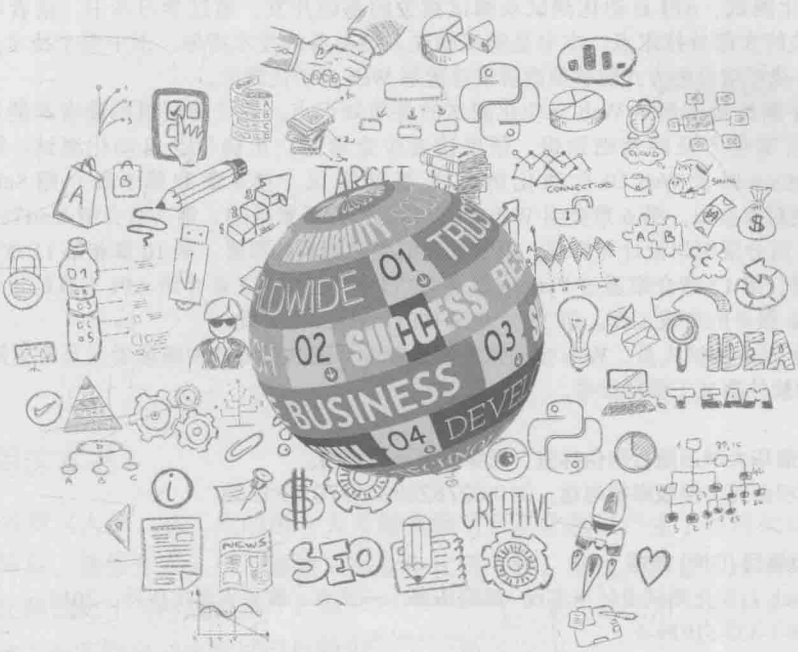
Python Web

自动化测试设计与实现

陈晓伍◎著

- 资深测试开发专家撰写，凝聚自己多年开发经验，系统且深入阐释利用Python进行Web自动化测试涉及的方法和实践
- 由浅入深剖析Web自动化测试开发过程中遇到的问题，涉及Web UI自动化测试、API自动化测试及测试相关的基础开发等

清華大學出版社



Python Web

自动化测试设计与实现

陈晓伍◎著

清華大學出版社
北京

内容简介

本书是资深测试开发专家的经验结晶，由浅入深地阐释了 Web 自动化测试的相关技术，包括 Web UI 自动化测试、API 自动化测试及测试相关的基础开发。通过学习本书，读者可以基本掌握 Web 测试相关的大部分技术点。本书是测试相关人员必备的技术指导。书中每个技术点都有示例代码，理论与实践相结合的方式能够使读者快速理解 Web 自动化测试。

本书循序渐进地讲解了 Web 自动化测试的各项知识点，使任何层级的读者都能从中受益。绪论部分介绍自动化方面的基础知识，帮助读者少走弯路，正确学会自动化测试。第 1~3 章介绍 Selenium、Python 以及 Web UI 自动化的相关基础知识。第 4 章和第 5 章介绍 Selenium IDE 和 Selenium 常规对象接口。第 6 章介绍 Web UI 自动化特殊场景处理。第 7 章介绍 UnitTest 单元测试框架。第 8 章介绍分层框架设计与实现。第 9 章介绍测试脚本的部署。第 10 章和第 11 章介绍 Web API 相关基础知识。第 12 章介绍通过 Python 发送 HTTP 请求。第 13 章介绍 API 工具的设计与实现。第 14 章介绍 Web 服务的集成工作。第 15 章介绍 HTTP Mock 的开发。

本书适合 Web 测试人员、Web 自动化人员、Web 开发人员等初中级读者以及希望使用 Python 作为编程语言的软件测试工程师参考。

图书在版编目 (CIP) 数据

Python Web 自动化测试设计与实现 / 陈晓伍著. —北京：清华大学出版社，2019
ISBN 978-7-302-51929-4

Ⅰ. ① P… Ⅱ. ①陈… Ⅲ. ①程序开发工具—程序设计 Ⅳ. ① TP311.561

中国版本图书馆 CIP 数据核字（2018）第 288545 号

责任编辑： 杨如林　薛　阳
封面设计： 杨玉兰
责任校对： 徐俊伟
责任印制： 李红英

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 装 者： 清华大学印刷厂
经　　销： 全国新华书店
开　　本： 186mm × 240mm　　**印　　张：** 23.25　　**字　　数：** 490 千字
版　　次： 2019 年 4 月第 1 版　　**印　　次：** 2019 年 4 月第 1 次印刷
印　　数： 1 ~ 2000
定　　价： 79.00 元

产品编号：072314-01

Preface 前　言

为什么要写这本书

作为一名测试人员，从工作的第一天开始我就对自动化测试产生了独特的兴趣。而最初的理由也很简单，就像开发人员不愿意只写业务代码一样，测试人员也不希望只局限于手动测试。自动化测试对于当时还是新手测试人员的我而言，完全可以用“高大上”来形容。自此，我便在学习和实践自动化测试的道路上越走越远。

而随着计算机技术及互联网的发展，如今作为一名测试人员，不仅要掌握针对于业务流程的手动测试方法和理论；还要具备一定的自动化、性能的测试能力。甚至于在找工作时会写脚本，会使用自动化工具进行测试已经成为测试人员的一种标配。本书总结了作者在项目实践中的多年工作经验，梳理了自动化测试需要掌握的一些基本技能和知识，帮助初级测试人员快速掌握目前常用的自动化测试手段和方法，提高自身的综合技能水平。

自动化测试对于测试新人而言，往往会理解为手动功能测试的自动化实现。比如：UI 自动化测试。但从广义概念来看，自动化测试还要包括：接口自动化、性能自动化、白盒自动化、安全自动化、自动化工具 / 框架 / 平台等一系列可以通过开发脚本来实现的测试。而本书所讲到的自动化测试内容包括：UI 自动化、自动化框架、接口自动化、自动化工具、自动化持续集成等相关知识。目的是给读者打开一个通向更加广泛的自动化测试之门。

此外，对于一些刚开始接触自动化测试的人员而言，自动化测试几乎等同于高效测试。其实现项目中并没有想象的那么美好，自动化测试需要根据不同的场景和需求来定制不同的自动化测试方案。本书最开始的部分就介绍了自动化测试的方法论和最佳实践，避免测试新人误入自动化测试的“陷阱”。

另外，本书也是一本 Python 的基础学习教程，作为 Python 的铁杆粉丝，自然也希望能够将 Python 语言最大程度地推广到自动化测试领域中来。正所谓“人生苦短，我用 Python！”

本书特色

1. 附带读书兴趣小组，方便学习沟通

为了便于读者相互沟通，提高学习效率，作者专门为本书建设了读书兴趣小组，读者可以通过登录 testqa.cn 并加入 seleniumbook 小组来学习和交流。另外本书中的源码包也会在这个小组中支持下载。

2. 涵盖多种自动化测试方法

本书涵盖自动化测试中使用到的多种测试方法，除了 UI 的自动化，还包括接口自动化，测试工具开发、CI 的使用。

3. 对 Selenium 工具的历史和原理进行了分析与说明

除了对于 Selenium 工具，提供相关接口的实例代码外，还介绍了 Selenium 的历史和基本原理。使得读者在学习的过程中，知其然也知其所以然。另外对 Selenium IDE 的操作和使用也做了较为详尽的说明，使得初学者也可以快速上手和使用 Selenium 进行自动化测试的实践。

4. 介绍详尽框架的开发

本书除了介绍 Selenium 的一些基本接口之外，还介绍了在基于 Selenium 的情况下，如何搭建可用性较高的测试基础框架。使用分层架构、数据驱动、业务解耦、功能封装等方式，让 UI 自动化测试不再是“可远观而不可亵玩”的技术。

5. 总结自动化最佳实践

本书的开头并没有一上来就开展技术的介绍，而是先从方法论和最佳实践开始。目的是让读者先理解“道”，再学习“术”。这样才能更好地学习和真正地利用自动化的相关测试技术。避免测试新人误入自动化的“陷阱”。

6. 提供基础的 Python 教程

除了介绍自动化相关的测试技术，本书还涵盖了书中其他地方需要用到的 Python 编程基础知识。为的是让读者只需一本书就可以开始步入自动化测试的行列。

7. 提供完善的技术支持和售后服务

本书提供了专门的技术支持邮箱：five3@163.com。读者在阅读本书过程中有任何疑问都可以通过该邮箱获得帮助。

读者对象

- ❑ 希望学习自动化测试技术的测试人员；
- ❑ 希望提升自身技术的测试人员；
- ❑ 希望了解自动化测试技术的开发人员；
- ❑ 其他希望利用自动化技术的相关人员。

本书主要内容

本书分为三大部分。

第一部分为方法论，主要介绍入门自动化测试之前需要了解的相关方法论和最佳实践。

第二部分为 Selenium 介绍，着重讲解 Selenium 的历史、原理、IDE 和接口的使用，同时还介绍了基于 Selenium 的自动化框架搭建。

第三部分为工具开发介绍，通过一步步深入的介绍带领读者进行接口测试工具、mock 测试工具的开发，同时集成到 Web 服务中。

除了这三个主要部分之外，还会有一些其他的自动化相关知识，各自分散在不同的章节中。比如：CI 持续集成的使用，基础环境的搭建，Python 语言的学习等。如果你是一名初学者，建议从第 1 章开始学习。

阅读本书的建议

- 测试新手读者，建议从第 1 章顺次阅读。
- 有一定 Python 基础的读者，可以根据实际情况有重点地选择阅读各个技术要点。
- 在学习框架之前，需要保证对 Selenium 和 Python 语法章节有了一定的掌握；先通读一遍有个大概印象，然后将每个知识点的示例代码都在开发环境中操作一遍，加深对知识点的印象。
- 结合 github 中的完整代码来实际操作，这样理解起来就更加容易，也会更加深刻。

进一步学习建议

当您阅读完本书后，相信已经掌握了 Python Web 自动化测试的基本知识。但如果还要更进一步深入下去，还必须要进一步地掌握 Python 的开发技术，以及加深对自动化测试的理解。掌握了扎实的技术能力之后，针对项目中需要提炼的流程和事务，进行分析并有针对性地优化。做好自动化项目最重要的一点就是：结合实际业务需求，否则可能就成为“空中楼阁”。

此外还需要学习性能、白盒、安全等相关测试技术，结合自动化来提升这些测试过程中的效率。比如：测试数据的准备、mock 系统的开发、代理监听、信息采集等。

如果希望在 Python 编程方面有更多的提升，推荐去阅读 Python 核心编程方面的书籍，而对于项目中的效率提升则需要自己更多地去实践、学习和思考。

勘误和支持

由于作者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，特意创建一个在线支持与应急方案的二级站点 http://www.testqa.cn/seleniumbook。你可以将书中的错误发布在 Bug 勘误表页面中，同时如果你遇到任何问题，

也可以访问 seleniumbook 小组页面，我将尽量在线上为读者提供最满意的解答。书中的全部源文件除可以从 github（http://github.com/five3）下载外，还可以从 testqa 站点下载，我也会及时更新相应的功能。如果你有更多的宝贵意见，也欢迎通过清华大学出版社网站（www.tup.com.cn）与我们联系，期待能够得到你们的真挚反馈。

致谢

首先要感谢我的爱人、岳母，是她们的辛苦付出和支持才让我有时间来进行本书的写作。

感谢出版社的编辑老师，在这一年多的时间中始终支持我的写作，你的鼓励和帮助引导我能顺利完成全部书稿。

还要感谢职业生涯中给予过我帮助的同事们，没有你们的信任和无私的帮助就没有这本书。

最后感谢我的爸爸、妈妈、哥哥、姐姐，感谢你们将我培养成人，并给予我的一切！

谨以此书献给我最亲爱的儿子，希望他一直快乐成长！

陈晓伍

Contents 目　录

绪论

自动化测试与其他技术工作相比有着自己的独特性，它不同于纯开发或者测试，在有工作量投入的前提下就必定有对应的产出，所以在正式学习这项技术之前，有必要对其进行一番真切的认识，让我们不仅能够学习这门技术，更能把这门技术运用到正确的项目中。

开发的直接产出、最终产出都是代码。测试的直接产出、最终产出都是测试覆盖的程序。自动化测试的直接产出是脚本，而最终产出却是效率。所以如果自动化测试没有产出效率则表示没有最终产出，此时的直接产出就没有意义了。这正是自动化测试的特殊之处。

因此，本书在一开始就针对如何更好地运用自动化测试技术展开了一番论述，把以前踩过的坑、趟过的河以及行业内的共同认知进行了整理归纳；希望能让新人在以后的项目中尽量避免这些误区，少走些弯路，这样才能让自动化技术真正地发挥它本来的作用。

如何学习自动化

近些年来作为一名软件测试人员，掌握一门过硬的自动化技术是必不可少的。无论是 UI 自动化、接口自动化还是性能自动化，至少需要掌握一种技术。对于此前以手工测试为主或者应届大学生而言，怎样才能学好自动化技术则是首要问题。由于本书主要介绍 UI 和 API 的自动化，所以接下来所讲的方法也是针对这些方面的。

首先，在学习自动化之前要有足够的意愿和信心，也就是说为了提升自己而主动学习，而非外力压迫去学习。这样在学习的过程中即使有困难和不顺你都可以坚持和跨越过去，否则可能一个小小的挫折就让你放弃甚至厌恶学习自动化。学习自动化技术是一件需要坚持的事情，只有不忘初心，才能方得始终！

其次，需要掌握一些编程基础，例如一门编程语言，如 Python、Java、C#、Ruby 等。具体需要掌握哪种语言主要取决于选择的自动化技术。例如，如果使用 QTP 则需要掌握 VBS，使用 Watir 则需要掌握 Ruby，而使用 Selenium 则上面提到的几种语言都可以。此外，可能还需要掌握数据库的基本使用方法，知道如何通过自己熟悉的语言去操作各种不同的数据库。还需要掌握一些基本的操作系统知识、基本算法等。

再次，需要了解所测试的对象使用的一些技术，例如，要对 Web 项目进行自动化测试，那么需要掌握 HTTP、HTML、JavaScript、CSS 等 Web 开发的相关知识，理解它的运行机制和原理，这样才能在进行自动化学习的过程中平稳而顺利地前进；否则，可能遇到很多莫名其妙、不得其解的现象或问题。

最后，如果满足了上述基本要求，那么恭喜你已经可以开始自己的自动化测试之旅了。可以从最开始的环境搭建，到 demo 样例，再到常用函数的学习、简单场景的实现、多个场景的实现、批量运行、框架设计等，一步步学习。这期间每一次的运行成功都会提升你的自信心，而每一次的执行失败则考验着你能否最终进阶到自动化测试工程师。在这里作者希望读者在学习时不要害怕、厌恶过程中出现的错误和问题，要积极主动地找到问题的原因并最终解决问题。每当解决一个问题之后，离成功就更进一步了。另外，如果你身边有技术大牛，可以向他请教学习，但切记不要一遇到问题就去寻找帮助，需要给自己一个思考的机会，也需要适度地消费技术大牛们的耐心和时间。

自动化项目选型

尽管软件测试人员对自动化技术趋之若鹜，但自动化技术本身不是万能的，并不是所有的项目都适合自动化测试；所以在进行自动化项目选项的时候就需要进行一下条件筛选，看下是否满足进行自动化的条件，这样不仅能让我们的自动化技术有施展的空间，也能让自动化测试技术带来实实在在的效益。下面就列出一些符合自动化测试技术应用的项目特点。

周期长且需求稳定

即项目本身是一个长期规划的，而不是短期的或者是新项目，因为开发测试脚本也是需要时间的，这个投入就需要在后期反复回归测试时补回来，如果项目周期不是足够长的话，可能脚本没有开发完项目就结束了；需求稳定指的是项目在长期的进行过程中不会大量或者频繁地修改需求，因为一旦需求改变了，意味着之前的测试脚本都将不可用，也就无法完成测试脚本的积累，最终也可能导致项目结束但脚本却没写完，或者真正在执行测试的脚本少之又少。

功能模块有回归需求

编写自动化脚本目的是提高测试效率，只有测试脚本被反复使用时，测试效率才能最大化地提升；试想如果测试脚本只被使用几次，即使效率提高了也是很有限的，但是开发脚本的成本却是很高的，所以被测试项目对功能模块回归的需求很大程度上决定了实施自动化测试的意义。

操作场景易于自动化

这里主要指的是某些特殊场景可能在人为的情况下很难实现，或不易完成的场景，例如，快频次的反复操作、大量的文本输入、精确的单击、大量的数字计算操作等。这些操作场景有些人为做不到，有些人为易出错，但是使用自动化技术则可以很容易实现这样的需求。如果项目中有大量这样的场景，那么自动化测试技术则是不二选择。

自动化的正确打开方式

上面只是从一个宏观的角度来审视一个项目是否适合自动化。接下来就从细节上来阐述下如何才能更好地实施自动化，在具体的自动化执行过程中需要关注哪些点，从而避免误入自动化测试的陷阱里。这里总结了 10 条参考建议。

考虑成本效益

成本效益即通常所说的 RIO（Return On Investment，投资回报率），在自动化测试中这个概率必须要牢记，因为自动化测试的初衷是为了提高效率和节约成本，如果最终都没能达成则表示自动化是失败的。因此在考虑是否要进行自动化测试的时候，需要优先核算 RIO 而不能误入为了自动化而自动化的陷阱。

自动化测试的投资主要为测试脚本开发、维护的人力成本，而自动化测试的回报为每次执行脚本所节约的人力成本；做一个简单的计算就是脚本开发和维护的总成本不应大于自动化测试所能节约的总成本。可以简单地理解为如下公式：

自动化测试工程师总人天数 < 自动化单次平均节约人天数 × 执行次数

从公式中可以得出如下结论。

- 执行次数越多越好。
- 有一个收回成本的临界点。

❑ 执行次数至少大于这个临界点才能节约成本。

从这里可以得出为什么项目周期需要足够长，且项目功能需要稳定；因为项目周期越长、可回归的次数越多，则自动化脚本执行的次数就越多，自动化测试得到的回报就会越高，自动化测试才能真正地发挥价值。那么问题来了，如果你的老板或上司需要你开展自动化测试，你应该怎么跟他保证呢？

提示 这里需要辩证地对待这个问题，既不能打包票也不能一味儿推脱，因为 UI 自动化测试的风险还是有的，并不是任何一个项目都是适合进行 UI 自动化，所以一旦遇到类似的情况，我们需要询问在下面的几个场景中，作为自动化测试的效果来看，最低可以接受的选项是哪个？

❑ 提高测试工作的效率。

❑ 增加测试工作的覆盖率。

❑ 节约测试工作的总成本。

❑ 以上三者。

针对提示中提到的选项，相信大多数人的期望都是第 4 项，而实际的测试项目中能达到第 4 项的项目并不多，但是这并不意味着达不到这个效果就直接放弃掉 UI 自动化测试。因为有时候我们是需要付出成本来换取时间，例如，为了缩短项目的回归周期；有时候是需要付出成本来提高产品测试覆盖率，例如，银行系统的准确性。使用不同的需求来裁定自动化测试是否有效，才是正确的投资回报的体现。

选择合适的工具

正如前面所提到的广义的自动化测试包括很多：功能、性能和安全等，不同的自动化类型需要选择不同的测试工具。即便是本书中重点讲解的功能自动化也是如此，针对不同的项目类型需要进行最佳工具的选择。

功能自动化测试工具有很多，从是否收费来分可以分为：商业、开源工具；从被测对象来分可以分为：Windows、Web 工具；从测试阶段来分可以分为：白盒、接口、GUI 工具。因此，项目的诉求不同、类型不同、阶段不同，所选取的工具是不一样的。

如果是公司没有购买工具的预算，则开源工具是首选；如果是 Windows 的程序，则可能会考虑 QTP、Rational、UIAutomation 等；如果是 Web 程序，则大多会选择 Selenium、Watir 等；如果是接口测试，则可以选择 SoapUI；正常情况下，都是可以选择到一款合适的测试工具的。