



引导测试人员攻克Python入门难点
提供可落地的HTTP协议接口自动化脚本

Python 接口 自动化测试

王浩然◎著



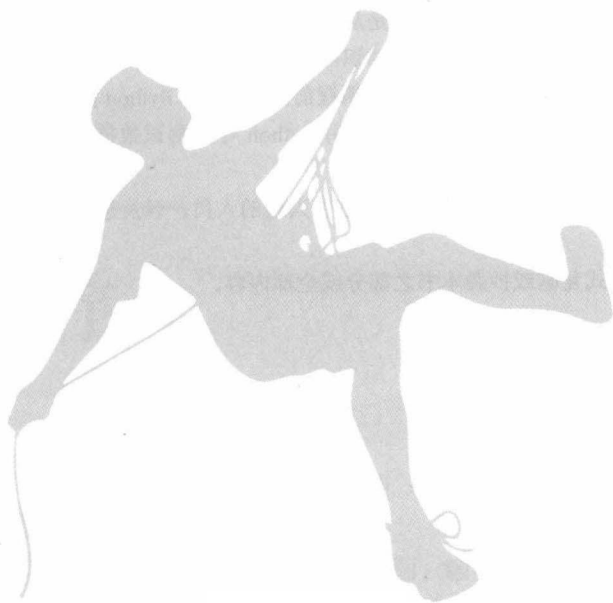
中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



测试馒头铺从0到1职业规划丛书



Python 接口 自动化测试

王浩然◎著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书没有采用传统的教科书写作模式，而是从要实现的目标着手，将“Python+MySQL 处理 HTTP 接口”过程拆分成一个个知识点，最后串联各个知识点。本书主要介绍了如何用 Python 实现接口自动化测试。全书主要内容包括接口基础、接口手工测试、编程前的准备、用 Python 操作 MySQL 数据库、用 Python 发送 HTTP 请求、用 Python 处理 HTTP 返回包、用 Python 导出测试数据、接口自动化及实际接口场景演示。

本书适合初、中级测试工程师，对 Python 语言感兴趣的人员，以及想要提升技术的人员。

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

Python 接口自动化测试 / 王浩然著. —北京：电子工业出版社，2019.5

（测试馒头铺从 0 到 1 职业规划丛书）

ISBN 978-7-121-35687-2

I. ①P… II. ①王… III. ①软件工具—自动检测IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2018)第 280896 号

策划编辑：王 静

责任编辑：牛 勇

印 刷：天津千鹤文化传播有限公司

装 订：天津千鹤文化传播有限公司

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：14.5 字数：230 千字

版 次：2019 年 5 月第 1 版

印 次：2019 年 5 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。



序

如今，软件测试岗位受到很大的挑战。一方面，敏捷开发模式、DevOps 实践等愈发流行，其强调开发与测试相融合，即测试人员能干开发的工作，开发人员能干测试的工作；另外一方面，测试行业开始认为自动化测试很重要，如果实现不了自动化测试，那么测试就无法敏捷起来（特别是在快速迭代、持续交付的环境下）。

说起自动化测试，根据公众号“软件质量报道”和相关机构最近的调查，目前的自动化测试（特别是面向 GUI 的自动化测试）效果还不够好，产出投入比不高，自动化测试做得好的公司或团队也不多。但是，基于 API 进行自动化测试（接口自动化测试）还是比较容易实施的，自动化率能达到 90% 以上，并且投入产出比高。另外，如今软件架构也慢慢转向 SOA 架构、微服务架构，基于 API 进行测试的需求越来越大，这给自动化测试提供了更多的机会。

本书正是帮助那些自动化测试基础比较弱，甚至是零基础的测试工程师转型做接口自动化测试，而且是基于现在如日中天的 Python 语言来开发自动化脚本，对渴望入门 Python 编程的朋友也有价值。本书循序渐进地引导读者完成接口自动化测试。

本书直接基于 Python 代码来实现接口自动化测试，不依赖其他测试工具，降低了学习门槛和使用成本。自动化测试所需的 Python 技能其实也很简单，读者也不用恐惧，用

一些资深人士的话说：1~2 天就可以了。即使学得慢一些，一周就能学会。在接口自动化测试过程中，一般建议以自动生成测试数据为主（即先基于自定义的业务数据模板来自动生成大部分测试数据，然后手工再补充一些特殊的测试数据）。未来可以结合人工智能算法来完善测试数据、自动分析与生成接口调用链等，虽然本书没有介绍这方面的内容，但有了本书作为的基础，读者就可以通过自学深入下去。

最后需要提醒读者：想要做好测试，深刻理解用户、产品和业务是非常重要的。任何商业软件最终都是为了解决业务问题和满足用户的需求，而测试正是对这种质量的保障。所以，在学习自动化测试的过程中，一定要重视业务需求、测试思维和测试方法等。只有具备良好的测试素质，才能让自动化测试发挥其价值，才能真正做到事半功倍。

朱少民

国内知名测试专家



前言

本书整体设计思想

自动化测试的前景

软件测试,在大多数的公司中还是处于相对弱势的地位,原因主要在软件测试本身:相比于软件开发,软件测试无论进入门槛还是编程能力,要求都低一些;而且大多数公司的软件测试还局限于手工测试。这就造成了业界对软件测试的偏见——软件测试只是随便按一按鼠标,技术含量低,比不得架构设计和模块化编程等。

诚然,软件测试离不开手工测试,但不能因手工测试而忽视有编程能力。技术性测试的方法有很多,常见的有自动化测试、性能测试、白盒测试、安全性测试等,这些测试方法都需要测试人员有比较强的编程能力。其中,自动化测试的进入门槛较低,但效果最明显,所以,自动化测试可以作为广大测试人员进入技术性测试的切入点。

现阶段,薪资高一些的测试岗位,普遍要求从业人员具有自动化基础及实际操作能力。所以,从就业角度来说,自动化测试是突破测试行业薪资瓶颈的一条捷径;从测试人员的职业发展来说,掌握测试编程技术,有助于建立技术思维及行业内部的沟通,便

于将来继续走测试技术路线，或是转到其他岗位。

本书写作目的

相信很多测试人员和我的经历很像，从最开始的手工测试开始积累经验，在这个过程中肯定想过做自动化测试、做技术。我最开始是用 QTP（Quick Test Professional，一种自动测试工具）录制了登录功能，看着浏览器自动打开、自动输入网址、自动登录，那种愉悦的心情是不言而喻的。但是，真正华丽转型（或者说能有一技之长）的测试人员还是偏少的，原因就在于难以突破关键技术点。

各种编程语言本身都有相同点，只要掌握了一门，其他语言学起来也就没那么难了。所以，如何掌握一门语言，如何跨过第一道门槛，成为制约测试人员技术能力提升的关键点，这也是本书编写的目的——引导测试人员突破 Python 的入门难点。

为什么要选择 Python？最主要的原因是——合适。正所谓“鞋合不合适，只有脚知道”。Python 简单易学的特点，恰好符合测试人员的要求。很多时候，我都在怀疑 Python 是不是专为测试人员量身打造的。关于 Python 的更多优势，读者可以在正文中看到。

本书特点

作者在构思本书的时候，也翻阅了市面上很多同类技术书籍，发现大多数都有一个通病——采用教科书的写作模式。所谓“教科书模式”，即按照“语言的历史→语法→章节练习→案例讲解”的模式来展开。这种模式很经典，内容也很翔实，但是太过于死板和理论化。这样的书虽然是结构明晰、循序渐进，仔细阅读后确实能收获很多，但是不利于快速上手。

读教科书模式的书时，学习者很容易半途而废，包括我自己，经常是看了不到三分之一的内容就放弃了，所有的新鲜劲都在语法和练习中被消磨殆尽了。我真正开始写脚本并不是从某本书开始的，而是从实际的项目拆分入手的，遇到问题再去翻书，纯粹将

它当作工具书来使用。

所以，本书不是按照传统的教科书模式编写的，而是从实际要实现的目标着手，一步步将目标拆分成知识点，再对知识点进行细分，将每个点拆分成一个个小的突破点——类似于“拆书帮”（某个学习社区）的形式，将“Python+MySQL 处理 HTTP 协议接口的过程”拆解到一个个章节中。读者在每个章节中都能学到完整的一个知识点，最后串联各个知识点，实现最终的学习目标。关于如何分解、如何逐个突破，读者可以在正文中看到。

读者通过实现每个章节的功能，逐步加深对 Python 的理解，通过小篇幅的功能实现来提升成就感，激发自己继续往下看、继续往下学的信心和勇气。本书所介绍的方法是作者在实际项目中实践过的，并且也被很多业内同行所采用。读者不仅可以将其用在 Python 学习中，也可以用在其他语言甚至生活中。

读者对象

- 对 Python 感兴趣的人员；
- 想在项目中实现 HTTP 协议接口自动化测试的人员；
- 想要提升技术的人员；
- 初、中级测试工程师。

代码下载

本书配套代码的下载地址：<https://www.broadview.com.cn/35687>。

有关于任何问题、建议和疑问，欢迎发邮件到：smallprocess@yeah.net。

致谢

感谢 Guido van Rossum 于 1989 年发明了 Python。

感谢电子工业出版社出版此书，以及为本书能够快速出版而做的审校、加工等辛苦工作。

感谢何飞在本书构思和出版中所做的指导和帮助。没有他的指引就没有本书的出现。

感谢我的家人，感谢你们对我工作的理解和支持，有你们一直的付出才有今天的这本书。

作 者



目录

1	本书整体设计思想	1
1.1	为什么要做懂技术的测试人员.....	2
1.2	为什么选择这本书.....	4
1.3	为什么选择 Python	5
1.4	本书能给你带来什么.....	6
1.5	自动化代码的设计思路.....	6
1.5.1	由手工测试分析出哪些步骤可自动化处理	8
1.5.2	以可重复步骤为契机，梳理自动化测试的步骤	9
1.5.3	抽象自动化步骤到功能点	10
1.6	补充知识点	10
1.6.1	什么是面向对象编程中的对象	10
1.6.2	什么是面向对象编程中的类	11
1.6.3	什么是编程语言中的实例	11
1.6.4	自动化测试是不是比手工测试覆盖率高	12

1.6.5	什么是自动化测试.....	13
1.6.6	什么是分层自动化测试.....	14
2	接口基础.....	18
2.1	什么是接口.....	19
2.2	接口的分类.....	23
2.3	HTTP 接口.....	24
2.3.1	HTTP 发送请求的方式.....	28
2.3.2	GET 方式和 POST 方式的区别.....	29
2.4	接口测试.....	30
2.4.1	什么是接口测试.....	30
2.4.2	为什么要做接口测试.....	30
2.4.3	如何开展接口测试.....	31
2.4.4	前/后端交互的“契约—接口”文档.....	32
2.5	接口实例.....	34
2.5.1	前端页面.....	34
2.5.2	数据流图.....	35
2.5.3	逻辑代码.....	39
2.6	补充知识点.....	40
2.6.1	名词解释.....	40
2.6.2	答疑.....	41
3	接口手工测试.....	43
3.1	HTTP 接口工具.....	44
3.2	Fiddler 工具的使用.....	47
3.2.1	Fiddler 工具介绍.....	47
3.2.2	手工调用 HTTP 接口.....	48

3.2.3	获取 PC 端的网络数据包	51
3.2.4	获取手机端的网络数据包	52
3.2.5	截包与改包	54
3.2.6	Fiddler 工具的其他功能	58
3.3	接口手工测试的用例设计	59
3.3.1	接口测试用例设计——总纲	59
3.3.2	接口测试用例设计——参数校验	59
3.3.3	参数校验——SQL 注入	62
3.3.4	接口测试用例设计——逻辑校验	63
3.3.5	接口测试用例设计——用例模板	64
3.4	补充知识点	65
4	编程前的准备	67
4.1	Python 环境准备	68
4.1.1	选择 Python 2 还是 Python 3	68
4.1.2	在 Windows 下安装 Python 3	69
4.1.3	Python 2 和 Python 3 共存之道	70
4.2	准备本地 MySQL 服务	71
4.3	补充知识点	74
4.3.1	Python 2 与 Python 3 的语法区别	74
4.3.2	Python 解释器	75
4.3.3	Python 的函数	75
5	用 Python 操作 MySQL 数据库	77
5.1	提前工作	78
5.2	操作 MySQL 数据库	80
5.2.1	用 Python 操作 MySQL 数据库的流程	80

5.2.2	用 Python 操作 MySQL 代码.....	81
5.3	本章所涉及的 Python 语法.....	95
5.3.1	模块与包.....	95
5.3.2	类.....	99
5.3.3	条件判断.....	104
5.3.4	异常处理.....	106
5.3.5	Python 3 代码注释.....	107
5.4	PyCharm 使用基础.....	109
5.4.1	为什么选择 PyCharm.....	109
5.4.2	PyCharm 使用基础.....	110
5.5	补充知识点.....	113
5.5.1	MySQLdb 与 PyMySQL.....	113
5.5.2	Python 命名规则.....	113
5.5.3	self.cur.scroll 源码分析.....	113
5.5.4	主流数据库的分类.....	115
5.5.5	MySQL 的基本语法.....	117

6 用 Python 发送 HTTP 请求 120

6.1	准备工作.....	121
6.2	发送 HTTP 请求实例.....	123
6.2.1	用 Python 发送 HTTP 请求的流程.....	123
6.2.2	用 Python 操作 HTTP 请求的代码.....	125
6.3	本章所涉及的 Python 语法.....	135
6.3.1	数据类型.....	135
6.3.2	方法与函数.....	137
6.3.3	切片.....	140
6.3.4	日志模块 logging.....	141

6.4	补充知识点	142
6.4.1	Python 的循环机制	142
6.4.2	logging	143
7	用 Python 处理 HTTP 返回包	144
7.1	提前工作	145
7.2	处理 HTTP 返回包实例	145
7.2.1	用 Python 处理 HTTP 返回包的基础	145
7.2.2	用 Python 处理 HTTP 返回包的流程	148
7.2.3	用 Python 处理 HTTP 返回包的代码	150
7.3	本章所涉及的 Python 语法	161
7.3.1	json 方法	161
7.3.2	字典的两个方法	162
7.3.3	eval()与 instance()方法	163
7.3.4	set()与 issubset()方法	163
7.4	补充知识点	164
7.4.1	Python 的垃圾回收机制	164
7.4.2	字符串的 startswith()和 endswith()方法	166
8	用 Python 导出测试数据	168
8.1	提前工作	169
8.2	用 Python 导出测试数据	170
8.2.1	导出测试数据的基础知识	170
8.2.2	导出测试数据实例	171
8.3	整体业务流程图	176
8.4	补充知识点	178

8.4.1	Python 时间戳.....	178
8.4.2	Excel 表格的操作.....	178
9	接口自动化起航.....	179
9.1	提前工作.....	180
9.2	代码之外.....	180
9.2.1	初始化数据.....	180
9.2.2	代码结构图.....	181
9.3	接口自动化起航代码.....	182
9.3.1	业务逻辑梳理.....	182
9.3.2	代码实例.....	183
9.4	代码操作步骤.....	188
9.5	补充知识点.....	190
9.5.1	用 print 格式化输出.....	190
9.5.2	数据驱动和关键字驱动.....	191
10	实际接口场景演示.....	192
10.1	提前工作.....	193
10.2	接口举例.....	193
10.3	准备与执行.....	193
10.3.1	设计接口测试用例.....	193
10.3.2	按照接口用例设计准备测试数据.....	193
10.3.3	在 config_total 表中增加执行与导出配置项.....	193
10.3.4	执行 main.py.....	195
附录 A	本书用到的 Python 代码清单.....	196

1

本书整体设计思想

本章讲什么：

- (1) 为什么要做懂技术的测试人员；
- (2) 为什么选择这本书，为什么选择 Python；
- (3) 本书能给你带来什么；
- (4) 自动化代码的设计思路。

本章主要介绍如何构思和实现接口自动化测试。传统的教学模式固然有其突出的一面，但如何能让读者快速入门、快速上手、快速产出才是本书的出发点。所以，本书的总体思想是“学以致用”，以目的反推要做的每一步，逐步实现每一步，需要什么就学什么，目的性很明确，最后再将“已实现的每一步”串联，达到最终的效果。

1.1 为什么要做懂技术的测试人员

物竞天择，适者生存。

相当一部分人选择软件测试（以下简称测试）行业的原因是，测试相对开发来说，其技术门槛更低。在 IT 行业，测试岗位对编程能力的要求相对较低。直白地说，只要会操作电脑就可以做测试。最重要的是，测试人员的待遇相对其他岗位来说还是比较高的，工作也是“白领”模式。所以，大批的新鲜血液源源不断地汇入测试行业。这也造成了在测试行业中，技术含量比较低、纯手工的重复性工作偏多。在笔者之前所在的公司中，测试部门居然不属于技术部门，可见大环境给测试工作贴的“标签”就是“技术含量较低”。

图 1-1-1 所示的是 51Testing 发布的 2010—2016 年软件测试行业人员年龄分布图。很明显，“90 后”已经成为测试行业的主力军，而“80 后”拿什么和这些年轻人竞争呢？

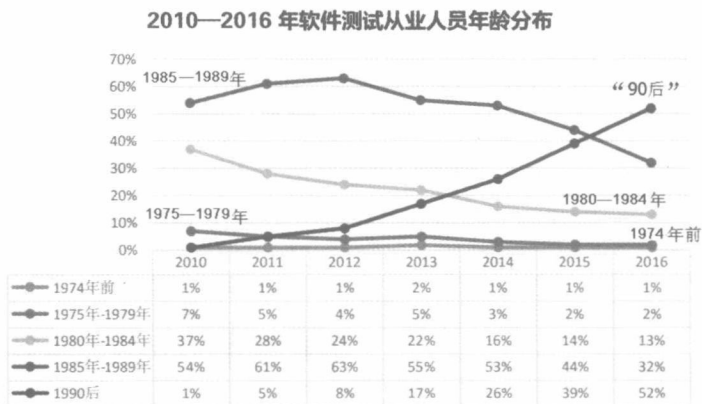


图 1-1-1 2010—2016 年软件测试行业人员年龄分布图