

Java 微服务测试

基于Arquillian、Hoverfly、
AssertJ、JUnit、Selenium与Mockito

Testing Java Microservices:
Using Arquillian, Hoverfly,
AssertJ, JUnit, Selenium, and Mockito

Alex Soto Bueno
[美] Andy Gumbrecht 著
Jason Porter

刘梦馨 译



Java 微服务测试

基于Arquillian、Hoverfly、
AssertJ、JUnit、Selenium与Mockito

Testing Java Microservices:
Using Arquillian, Hoverfly,
AssertJ, JUnit, Selenium, and Mockito



Alex Soto Bueno
[美] Andy Gumbrecht 著
Jason Porter

刘梦馨 译

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书从实战出发,介绍微服务架构所带来的测试方面的挑战,以及如何利用新的技术来应对这些挑战。通过本书,读者可以学会如何编写微服务架构下的单元测试、组件测试、集成测试以及契约测试。在此过程中还会用到 Arquillian、ShrinkWrap、Pact、Selenium、Docker、Hoverfly 等多个帮助测试的工具和框架。书中涵盖大量的代码和样例,可以帮助读者快速上手,并在自己的实际工作中应用这些技术。

本书适合有一定 Java 基础的开发和测试人员,对使用其他编程语言的开发者也会有一定的帮助。

Original English Language edition published by Manning Publications, USA. Copyright © 2018 by Manning Publications. Simplified Chinese-language edition copyright © 2019 by Publishing House of Electronics Industry. All rights reserved.

本书简体中文版专有出版权由 Manning Publications 授予电子工业出版社。未经许可,不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字:01-2018-8267

图书在版编目(CIP)数据

Java微服务测试:基于Arquillian、Hoverfly、AssertJ、JUnit、Selenium与Mockito/(美)亚历克斯·索托·布埃诺(Alex Soto Bueno),(美)安迪·巩布莱希(Andy Gumbrecht),(美)詹森·波特(Jason Porter)著;刘梦馨译. —北京:电子工业出版社,2019.7

书名原文:Testing Java Microservices: Using Arquillian, Hoverfly, AssertJ, JUnit, Selenium, and Mockito
ISBN 978-7-121-36710-6

I. ①J… II. ①亚… ②安… ③詹… ④刘… III. ①JAVA语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第106667号

责任编辑:张春雨

文字编辑:李云静

印 刷:山东华立印务有限公司

装 订:山东华立印务有限公司

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:19 字数:350千字

版 次:2019年7月第1版

印 次:2019年7月第1次印刷

定 价:89.00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 zltts@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。

译者序

微服务架构的理念日渐深入人心，然而隐藏在其架构优势下的还有对传统软件开发模式的强烈冲击。如何在测试环境中部署种类繁多的微服务？如何对复杂的交互、多样的场景进行完整覆盖？单元测试、组件测试、集成测试以及端到端测试又有怎样的不同？有没有专门针对微服务架构的测试框架和工具？本书就将告诉你上述问题的答案。

本书介绍了大量 Java 微服务测试领域最为前沿的技术。本书从微服务架构讲起，介绍了测试在这种新型架构下所面临的问题和挑战，之后逐一介绍了单元测试、组件测试、集成测试、契约测试和端到端测试领域针对这些问题的应对方法。本书还提供了一个完整的微服务应用作为样例，从代码层面介绍如何编写不同领域的测试，给作者提供了上手实践的方法。本书的几位作者都是在测试领域内耕耘多年的老兵，相信他们宝贵的经验能给你带来收获。

本书主要介绍 Java 微服务测试，但是其中的方法和理念对于使用任何一门编程语言的开发和测试人员都会有很大的帮助。希望本书能够帮助你更好地改善工作中开发和测试的流程，提升整体的代码质量。

愿你的程序永远不出 bug。

致我的父母：感谢你们给我买的 ZX Spectrum（早期的 8 位个人计算机）。

——Alex Soto Bueno

致我的孩子：Antony 和 Toriann。你们拥有我，但是我拥有这本书！

——Andy Gumbrecht

致伟大的软件工程师社区：我们在一起做令人惊叹的事！

致我的家人（尤其是我的妻子 Tessie）：感谢在这“疯狂”的人生旅程中能与你们一路同行。

——Jason Porter

序言

在编程的早期是没有任何框架的，测试散落在代码的各个角落，以确保重要的软件功能的确是经过测试验证的。在那个时候，存储空间是十分宝贵且有限的。

后来，单元测试从一个流行概念演化成了软件测试事实上的标准。存储空间的问题也得到了很大的缓解，再也不能成为不编写测试代码的借口。如今，几乎所有的开发者都已经早早开始学习并使用单元测试的方法，并且单元测试已经成为软件开发成功的基础。

今天的企业级应用已经不可能仅依赖简单的单元测试来保证功能的完备。如今的客户也变得更加苛刻，要求也越来越高。如果企业想要充分满足客户的要求，就需要在开发的过程中使用多种测试策略。

本书一方面介绍如何解决当下企业测试的问题，另一方面介绍如何应对未来将微服务引入应用架构所带来的测试挑战，这将会给读者带来更大的价值。

我们花了很长时间来编写本书：最初它只是一个关于某个框架的小册子，现如今它已经成为一个介绍大量测试策略供读者选择的具有 10 章且内容丰富的图书。随着技术的不断演进，我们在这个过程中收获颇多。为了能给读者提供尽可能多的选

择，我们尽可能地聚焦于策略、方法论和解决方案，而不是简洁的代码和看起来令人兴奋然而最终却不可用的应用。如果通过本书你可以对如何测试自己的应用有新的想法，那么我们的目的就达到了。

我们十分感谢你愿意花时间来读这样一本书，希望你能够喜欢我们的写作风格。

致谢

本书由 3 位独立开发者完成，在这里让我们感谢彼此的理解和支持，感谢每个人的辛苦工作和反馈。

十分感谢 Arquillian 项目的每一位参与者，尤其是 Aslak Knutsen、Dan Allen、Bartosz Majsak 和 Matous Jobanek。在你读完这本书的时候，就会了解他们在这个伟大的项目上投入了多少精力，以及这个项目会如何显著地帮助你改善软件测试的质量。

开源软件（OSS）社区提供了很多极其有用的工具来帮助每个人更有效地进行测试。这些项目中的大部分都是由那些不知疲倦且无人知晓的开发者在他们的空闲时间完成的，在这里我们要表达对他们这份难能可贵工作的敬意。我们同时希望作为读者的你，也能够每个合适的机会对这些“不可思议”的开发者表达谢意。

感谢 Daniel Bryant 和 Marcin Grzejszczak 花费时间对契约测试部分进行讨论。

在我们徘徊不前之际，Cynthia Kane 和 Tiffany Taylor 两位编辑给我们提供了前进的动力，帮助我们完成了这个工作。利用空闲时间写书是一项很有挑战性的任务，十分感谢 Cynthia 和 Tiffany 帮助我们完成了这项任务。同时也要感谢帮助这本书从可能变为现实的 Manning 出版社的每一位工作人员：出版者 Marjan Bace 以及相关编辑和产品团队。

Joshua White 提供了大量技术上的论证，感谢他解决了本书质量上的问题。他对我们的测试进行了测试！

感谢每一位花费时间阅读并提供反馈的读者，他们帮助我们完成了最终的图书产品。这些读者包括我们的技术同行评审员：Aleksandar Dragosavljević、Alex Jacinto、Anshuman Purohit、Boris Vasile、Conor Redmond、Eddú Meléndez Gonzales、Ethan A. Rivett、Fabrizio Cucci、Gualtiero Testa、Henrik Løvborg、Jan Paul Buchwald、Jonathan Thoms、José Díaz、Kiran Anantha、Leo van den Berg、Mari Machado、Nilesh Thali、Piotr Gliźniewicz、Robert Walsh、Yagiz Erkan 和 Zorodzayi Mukuya。

最后，要感谢家人帮助我们度过了写书过程中一个个漫长的周末以及交稿前的忙乱和写作过程中无数起起伏伏的时光。没有你们的支持，我们将无法完成这项任务！

关于本书

如今，测试已经成为每个开发者在软件开发过程中必不可少的一部分。然而原来并非如此，测试框架是在软件开发出现很长时间之后才出现的。由于现在已经有了足够多关于为什么需要进行软件测试的理论，因此本书并不会关注这一部分。对我们来说，如何进行软件测试以及如何将这个理念带给其他人是更为重要的话题。本书主要关注如何测试这一部分，并提供了应用代码的示例以帮助你快速上手。

本书中的很多内容都可以在互联网上找到，然而在人们真正需要去寻找这些资料的时候，却常常并不一定能联网查阅，即使联网也不一定能及时搜集或找到所需的内容（或者特意选择不联网）。如果使所有这些人们需要的信息汇聚到一起供人直接查阅使用，那将是非常方便的事情。我们希望本书就可以作为这样一种资源。

测试是一个相当通用的术语，但同时测试也是一系列技术的组合，如果想发挥它的最大威力，你需要对相应的技术进行很好的组合。我们收集了大量的热门主题和组件，并且将其以一种最为合适的方式进行组织并展示给读者。

谁应该阅读本书

我们很想说“所有人”都应该阅读本书，然而实际并非如此。我们的目标用户是企业级的 Java（Jave EE 和 Spring）应用开发者。如果你就是这样的开发者，那么

本书将会帮助你更深入地进行单元测试。

如果你并不是一个专门的 Java 开发者，本书依然可能包含你感兴趣的内容。本书提供的大部分信息都是和编程语言无关的，并可以应用于多个不同的领域。

正如书名所示，我们的关注点是正在不断兴起的以微服务为基础的架构。但是我们也提供了大量和更通用的企业级测试相关的信息，因此不必担心本书遗漏了这些内容——书中只是更深入地介绍了微服务相关的内容。

路线图

本书分为 10 章，下面是所包含内容的速览：

- 第 1 章介绍与微服务相关的术语以及为什么要使用微服务。
- 第 2 章介绍阅读本书预先需要掌握的知识以及如何构建代码，帮助你在阅读本书之后的过程中节省时间。
- 第 3 章会带你快速浏览通用的单元测试技术、方法论以及最佳实践。
- 第 4 章深入介绍 Arquillian 测试框架。
- 第 5 章展示如何创建一个相互依赖的微服务应用的集成测试。
- 第 6 章讨论消费者驱动的契约测试。
- 第 7 章探索端到端测试技术和工具。
- 第 8 章介绍利用 Docker 创建可复用的测试环境。
- 第 9 章探索服务虚拟化的概念和实现。
- 第 10 章讨论持续交付和使用 Jenkins 构建流水线。

我们以一种循序渐进地介绍各个基本主题的方式来安排章节的顺序。每一章都会承接上一章的内容，当然你也可以按照自己的兴趣来安排阅读的顺序。本书选择 Java 作为实现示例的编程语言，但是本书所讨论的原则可以被应用到其他语言和框架中。

代码规范和下载

本书中包含了大量的代码示例，这种代码部分以等宽字体（fixed-width font）展示，以便与正文的其他内容进行区分。

本书中的很多源码都已经过“格式化”，包括加入换行、更改缩进，以更好地适应图书版面。在某些情况下，还包含了续行符（`\`）。此外，如果代码已经在正文中进行了解释，代码中的注释便会被移除。与代码清单相关的注释中突出显示了相关的重要概念。

本书的源码可以通过 www.manning.com/books/testing-java-microservices 获得¹。

¹ 也可通过<http://www.broadview.com.cn/36710>下载。——译者注

关于作者

Alex Soto Bueno 是一位 Java 大师，也是一位在 Red Hat 工作的软件工程师，主要研发提升测试体验的工具。他十分喜爱 Java、软件自动化，并且深信开源软件。Alex 是 NoSQLUnit 项目的创建者以及 JSR374（JSON 处理的 Java API）专家组的一员，并且是一位国际讲师。

Alex 从 ZX Spectrum（使用 POKE 命令的遥远年代）就开启了编程之路，之后使用过多台不同的计算机，例如 80286（感谢他的父母 Mili 和 Ramon 买下了它）。从 La Salle 大学计算机工程专业毕业后，他加入了 Aventia，开始了自己的职业生涯，在此期间开发了一个生成并校验电子信号的平台。之后他加入了 Grifols，并开发了一种医疗诊断设备，严格的测试成为其软件生涯中重要的一部分。之后他加入了 Everis 的银行部门，其间开发了电子投票系统及 CloudBees。

工作之余，Alex 会陪伴他的妻子 Jessica 以及两个女儿 Ada 和 Alexandra，Ada 和 Alexandra 是他的掌上明珠。

Andy Gumbrecht 是 Tomitribe 的高级软件工程师。从 12 岁起他就对任何与计算机相关的事情感兴趣，并在那时幸运地拥有了一台高达 1KB 内存的计算机 Sinclair ZX81。那时候许多早期的程序示例都是表单上的二进制码，需要手动输入到计算机中。这非常容易出错，因此 Andy 很快就找到他的兄弟 John，让他作为自

己的测试者来保证输入的正确性。这也是他第一次意识到测试代码的重要性。

接触机器码和 BASIC 使 Andy 在代码优化方面的能力持续提升。在成为英国军队的皇家工程师之后不久，他回到德国的大学继续深造。之后他进入 PROVOX Sytemplanung GmbH 实习并在那里工作了很多年，主要开发政府软件。

Andy 从 2007 年开始进行开源软件的研发并在 2009 年加入了 Apache OpenEJB 和 Apache TomEE Application Server 项目，现在他已经成为项目管理委员会的一员。

Jason Porter 从 12 岁起就开始不断提升自己的软件开发技术。在那之前的几年间，他通过查看一个在古老的 80286 上用 BASIC 编写的游戏开启了自己的计算机和编程之旅。在编程方面的兴趣带领他之后又接触了 Java 和 C/C++。他在 Netscape Navigator 和 IE 浏览器时期成为一名早期的 Web 开发者，和 DHTML 以及层级元素的“斗争”占据了他的大量时间。15 岁时，Jason 在当地的一家 Web 开发公司找到了一份工作，开始开发网站并利用 Perl 来编写 CGI 脚本（然而，他很想忘记那段岁月）。在那之后，他在多个领域工作过，使用过 Java、PHP、Ruby、C# 以及 JavaScript。他认为自己主要是一名后端开发者，但是其实他拥有应对任何编程领域问题的能力。

在 Red Hat，Jason 参与过多个不同的框架、网站和集成工作。他在美国和世界其他地方做过的演讲可以帮助人们更好地理解编程的概念或新的技术。Jason 现在和他的妻子以及 5 个孩子居住在犹他州，对于 Jason 来说，带孩子可比从事计算机编程工作要困难多了。

关于封面

本书的封面图片标题为“Visitor to the Tuileries Gardens”，它是 Eugène Lami（1800—1890）的手工上色木制作品。该图片出现在 20 世纪 40 年代早期 Louis Curmer 在巴黎出版的一部巨作 *Les Français peints par eux-mêmes: Encyclopédie morale du dix-neuvième siècle*（第 3 卷）的一篇短文（*The French painted by themselves: moral encyclopedia of the nineteenth century*）中。这部作品通过一个个代表性的人物（尤其是一些特定类型的人物和某些少见的职业）展示了法国社会当时的流行趋势。这部作品有 5 卷介绍巴黎的情况，3 卷介绍法国其他省份以及殖民地的情况。

这部作品中丰富多彩的图片向我们生动地展示了 200 年前世界上各个村庄和地区的独特风采。在那个时代，仅相隔数里的人就可能有着完全不同的着装风格，大家可以通过着装对人群进行准确的区分。该系列图片向我们展示了那个时期以及历史上除了我们现在这个快速发展时期外，其他时期中生命的隔离感和距离感。

着装风格在那之后发生了巨大的变化，那个时期如此丰富的多样性逐渐消失了。现在我们已经很难通过着装来区分来自两个大陆的群体。也许我们牺牲了文化和视觉上的多样性，以追求个人生活、智慧创作以及技术生命的多样性。

在 Manning，我们通过从 200 年前丰富多样的地区生活中选择封面图片来纪念具有创造力、主动性以及颇具乐趣的计算机事业。

目录

第1章 微服务概述.....	1
1.1 什么是微服务，为什么要使用微服务.....	2
1.1.1 为什么要使用微服务.....	3
1.1.2 什么是微服务.....	5
1.1.3 持续集成、持续部署和 Docker.....	6
1.2 微服务网络及其功能.....	7
1.2.1 微服务网络.....	7
1.2.2 微服务特性.....	8
1.3 微服务架构.....	9
1.3.1 资源组件.....	10
1.3.2 业务领域组件.....	10
1.3.3 远程资源组件.....	11
1.3.4 持久化组件.....	11
1.4 微服务单元测试.....	12
1.4.1 孤立型单元测试.....	12

1.4.2 联合型单元测试.....	12
总结.....	13
第2章 测试下的应用	14
2.1 准备开始.....	15
2.2 准备工作.....	16
2.2.1 Java Development Kit.....	16
2.2.2 构建工具.....	17
2.2.3 环境变量.....	17
2.2.4 集成开发环境 (IDE).....	18
2.3 架构.....	18
2.3.1 游戏服务.....	19
2.3.2 评论服务.....	22
2.3.3 视频服务.....	24
2.3.4 聚合服务.....	26
2.3.5 整体架构.....	27
2.4 应用设计模式.....	28
2.4.1 解构.....	28
2.4.2 ECB 模式.....	38
2.4.3 其他模式.....	40
2.5 设计决策.....	42
总结.....	42
第3章 微服务单元测试	44
3.1 单元测试技术.....	44
3.1.1 联合型单元测试.....	45
3.1.2 测试替身.....	46
3.1.3 孤立型单元测试.....	48
3.1.4 微服务的单元测试.....	49