

Testing Microservices with Mountebank

Mountebank

微服务测试

[美] 布兰登·贝耶斯(Brandon Byars) 著
王 超 译



MANNING

清华大学出版社

Mountebank 微服务测试

[美] 布兰登·贝耶斯(Brandon Byars) 著

王超 译

清华大学出版社

北 京

Brandon Byars

Testing Microservices with Mountebank

EISBN: 978-1-61729-477-8

Original English language edition published by Manning Publications, USA (c) 2019 by Manning Publications. Simplified Chinese-language edition copyright (c) 2019 by Tsinghua University Press Limited. All rights reserved.

北京市版权局著作权合同登记号 图字: 01-2019-4449

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Mountebank 微服务测试 / (美) 布兰登·贝耶斯 著; 王超 译. —北京: 清华大学出版社, 2020

书名原文: Testing Microservices with Mountebank

ISBN 978-7-302-54089-2

I. ①M… II. ①布… ②王… III. ①互联网络—网络服务器 IV. ①TP368.5

中国版本图书馆 CIP 数据核字(2019)第 241988 号

责任编辑: 王 军

封面设计: 孔祥峰

版式设计: 思创景点

责任校对: 牛艳敏

责任印制: 宋 林

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: 三河市春园印刷有限公司

经 销: 全国新华书店

开 本: 170mm×240mm 印 张: 14.5 字 数: 260 千字

版 次: 2020 年 1 月第 1 版 印 次: 2020 年 1 月第 1 次印刷

定 价: 79.80 元

产品编号: 084079-01

译者序

最近几年，随着现代应用程序的功能越来越复杂，软件的迭代更新速度越来越快，企业主流开发方式逐渐从传统的单体式产品交付转移到微服务架构，因此，软件测试工程师只有及时更新自己的测试工具和方法，才能有效地提高测试覆盖率，尽早发现潜在的漏洞，确保微服务满足企业对软件质量日益增长的严格要求。微服务的前身是 Peter Rodgers 博士 2005 年提出的微 Web 服务(Micro-Web-Service)，2014 年 Martin Fowler 与 James Lewis 共同给出了微服务的定义，即它是一种架构模式，将单一的应用程序分成了一系列微小服务，服务之间相互协作来为用户提供功能。每个服务在独立的进程中运行，服务与服务之间采用基于 HTTP 的 RESTful API 的轻量级通信机制。

与传统的测试方法相比，微服务架构使测试面临诸多挑战：如服务-模块-层次之间具有复杂的依赖项，不同的服务在不同的上下文中运行，端到端测试多个服务容易出错，测试结果与网络的稳定性有关，故障分析具有更高的复杂度，开发团队之间增加的沟通成本等。因此，测试应具有自动化、层次化和可视化特性。mountebank 作为一个开源的、跨平台的、支持多协议的功能强大的服务虚拟化工具，能够有效地帮助测试工程师解决以上问题。它易于安装，并且不依赖于任何平台。

本书主要介绍如何使用 mountebank 测试微服务，通过阅读本书，你还可以深入了解 mountebank 的全部功能以及服务虚拟化的适用范围。无论是微服务系统的开发工程师，还是进行 QA 或者性能测试的测试工程师，以及将客户需求转换为实际可执行项目的解决方案架构师，都适合阅读本书。

本书在翻译过程中得到了清华大学出版社编辑的帮助和支持，他们指出了译文中的不当之处，使我能够及时修改，以便更好地表达出原作者的意图，同时带给读者更流畅的阅读体验，在此对他们表示衷心的感谢！我还要感谢我的家人，他们在生活上为我提供了极大的支持，帮忙照看小孩，使我能够专注于本书的翻译工作，最终顺利及时完稿。在翻译的过程中还参考了一些专业论坛资料，在此一并表示感谢。尽管我对译稿进行了多次校对和修改，但难免存在疏漏之处，敬请读者批评指正。谢谢你们。

王 超
于南阳理工学院

序 言

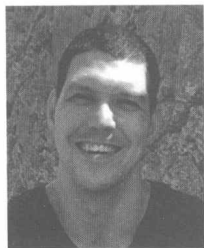
Pete Hodgson 曾经开玩笑说，构建自己的模拟框架是 ThoughtWorks 开发人员的一个惯例。那些日子已经过去了，不是因为 ThoughtWorks 不再关心测试(我们非常关心)，而是因为测试工具的功能更强大，并且我们现在需要关注更多有趣的问题。在 2014 年的冬天，我遇到了一个测试问题，结果发现无法测试会妨碍你专注于解决那些更有趣的问题。

我们采用了微服务体系结构，但受到一些摒弃的遗留服务代码的限制。服务虚拟化使用测试来模拟下游网络依赖性的想法对我们来说并不新鲜，即使该术语在开源世界中并不常见。似乎每个新的开发人员加入团队时，他们都建议使用 VCR(Ruby 工具)或 WireMock(Java 工具)来解决问题。这些工具都很出色，很多人都在使用它们。ThoughtWorks 已经为 mix 提供了更多优质工具(stubby4j 和 Moco)，类似 Hoverfly 的工具很快就会出现。如果需要虚拟化 HTTP 服务，那么选择其中任何一个都是可以的。

遗憾的是，我们的下游服务不是 HTTP。新团队成员不断建议使用相同类型工具的方式提供了很好的证据证明该方法有效。事实上，没有这样的工具，我们就无法获得在测试中所需要的信心，而测试需要 mountebank 的支持。

虽然本书介绍的是 mountebank，但也涉及了测试和持续交付，以及微服务和体系结构。你可以在不使用 mountebank 的情况下完成工作，但如果遇到问题，服务虚拟化可能会有助于解决更有趣的问题。

作者简介



Brandon Byars 是 ThoughtWorks 公司的首席顾问，也是 mountebank 的创建者和维护者。他在 IT 领域具有 20 年的经验，曾担任开发人员、DBA、架构师和客户经理。当他不再热衷于测试自动化时，他专注于将系统思维应用到大规模开发中，并在我们已经打开的潘多拉技术盒的世界中找到重新发现人类意义的方法。

致 谢

在 ThoughtWorks 这样的公司工作的好处是我认识的很多人都撰写过技术书籍。但是他们中的每一个人都告诉我写一本书是非常具有挑战性的(感谢 Martin Fowler、Neal Ford 和 Mike Mason)。

幸运的是, Pete Hodgson 不是其中之一。不过, 他写了几篇文章, 其中一篇是关于在 Martin Fowler 的 bliki 上测试 JavaScript。我第一次读了 10 遍也没有真正理解它, 因为自己不是 JavaScript 开发人员, 我尝试基于对 promise 的简单解释来实现一个同步的 promise 库。几周后, 当我绞尽脑汁并意识到写自己的 promise 库是一个糟糕的想法时, 我向 Pete 求助。他首次使用 mountebank 向我展示了如何实际测试 JavaScript。我觉得这是一件好事, 因为我正在写一个测试工具。谢谢 Pete。

Paul Hammant 是另一个从不告诉我写一本书有多难的人。然而, 他也从未有意告诉我管理一个流行的开源项目有多困难。作为一个长期的开源者(他启动了一种控制框架的早期版本 Selenium, 并做出了一系列其他举措), 他很可能认为每个人都有同样的愿望, 像他一样每天晚上把时间留给编码和博客, 以及管理社区工作。Paul 还是一个强有力的 mountebank 推广者和非常优秀的导师。

当然, 如果没有第一团队的支持, 这一切都是不可能的。第一团队的名字 SilverHawks 是根据一部卡通画命名的。我要感谢 Shyam Chary、Bear Claw、Gonzo、Andrew Hadley、Sarah Hutchins、Nagesh Kumar、Stan Guillory 和许多其他人。mountebank 社区是从那些微不足道的开始发展起来的, 我要感谢所有为改进产品而投入空闲时间的人们。

Manning 出版社建议我写一本书时, 我在俄克拉荷马州。这真的很难。开发编辑 Elesha Hyde 很出色, 即使我花费很多时间进行写作, 他在生活上给了我强大的支持。我在俄克拉荷马、达拉斯、多伦多、温哥华、旧金山、圣安东尼奥、休斯敦写过这本书。是的, 我写这本书的时候, 同时在海滩上喝莫吉托酒(第 4 章是这样)。

这让我想到 Mona。你让我在周末和假期写作。你让我在家庭活动中写作, 尤其是当 Patriots 在 Super Bowl 中玩耍时(或者不管他们怎么称呼最后一场比赛, 我都不再关注棒球了)。你让我在水疗中心和游泳池写作, 同时照看孩子不让他们溺水。谢谢你。

关于封面插图

《Mountebank 微服务测试》封面插图的标题是“一个来自斯洛文尼亚的人”。这幅插图是 Balthasar Hacquet 的 *Images and Descriptions of Southwestern and Eastern Wends, Illyrians, and Slavs* 的最新再版，它由克罗地亚位于 Split 的 Ethnographic 博物馆于 2008 年出版。Hacquet(1739—1815)是奥地利的一名医生和科学家，他花费很多年的时间研究了奥地利帝国许多地方的植物学、地质学和人种学，以及威尼托、朱利安阿尔卑斯山和西巴尔干半岛，这些地方过去居住着伊利里亚部落的人民。Hacquet 出版的许多科学论文和书籍都附有手绘插图。

Hacquet 出版物中丰富多样的图画生动地说明了两百年前东阿尔卑斯山和巴尔干西北部地区的独特性和个性。当时，相隔几英里的两个村庄的着装风格表明，人们是独一无二的，属于彼此独立的个体，可以很容易地通过穿着来区分社会阶层或行业的成员。从那以后，着装风格发生了变化，各地区当时如此丰富的着装多样性逐渐消失。现在仅靠着装已经很难区分来自不同大陆的居民。而如今，斯洛文尼亚的阿尔卑斯山或巴尔干沿海城镇中风景如画的城镇和村庄的居民与欧洲其他地区的居民也不易区分。

Manning 出版社采用展现两个世纪前丰富多元的各地生活的图书封面，以此来颂扬计算机行业的创造性、主动性和乐趣。

前 言

我用 mountebank 编写了测试微服务的程序，展示了服务虚拟化如何帮助你测试微服务，以及 mountebank 如何成为一个强大的服务虚拟化工具。这就需要对 mountebank 有一个深入的了解。本书的中间部分专门讨论这个主题，但是许多经验都适用于任何服务虚拟化工具。

本书读者对象

mountebank 是一种对开发人员友好的工具，它使开发人员成为使用 mountebank 测试微服务的主要受众。希望读者对测试自动化有一些了解，但是我避免在本书中使用任何高级语言特性来重点关注工具和方法。自动化友好的 QA 测试人员也会发现本书的价值，那些专门从事性能测试的人员也是如此。最后，服务虚拟化越来越成为一个体系结构问题，在这些页面中，我希望为解决方案架构师提供正确决策所需的论据。

本书内容安排

本书分为三部分 10 章。

- 第 I 部分介绍了分布式系统的总体测试原理。
 - 第 1 章简要介绍了微服务，并对传统的端到端测试进行了评论。它有助于解释服务虚拟化如何适应微服务的世界，并为 mountebank 提供一个心理模型。
 - 第 2 章建立了一个示例体系结构，我们将在本书中反复讨论，并展示如何使用 mountebank 来自动执行确定性测试，尽管它是分布式体系结构。
- 第 II 部分深入介绍 mountebank，让你全面了解它的功能。
 - 第 3 章提供了了解 HTTP 和 HTTPS 环境中基本 mountebank 响应的基础知识。它还描述了通过配置文件管理测试数据的基本方法。
 - 第 4 章探讨了谓词——mountebank 对不同类型请求的不同响应方式。还介绍 mountebank 关于匹配 XML 和 JSON 的功能。
 - 第 5 章介绍 mountebank 的记录和重放功能。mountebank 使用真实系统的

代理来捕获真实的测试数据。

第 6 章展示如何通过使用一个称为注入的特性在 JavaScript 中编写你自己的谓词和响应来对 mountebank 本身编程。我们将研究注入如何帮助解决 CORS 和 OAuth 握手中的一些棘手问题,包括虚拟化 GitHub 的公共 API。第 7 章通过研究应用于响应的行为,对 mountebank 引擎的核心功能进行了详细介绍。行为允许用户添加延迟,从外部源查找数据以及执行大量其他转换步骤。

第 8 章展示第 3~7 章中的所有概念如何延伸到 HTTPS。mountebank 的引擎是协议不可知的,我们给出了基于 TCP 的示例,包括扩展的 .NET Remoting 场景。

- 第 III 部分后退一步,将服务虚拟化放在更广泛的上下文中。

第 9 章探讨微服务的一个示例测试管道,从单元测试到手动探索测试,并展示了服务虚拟化的适用和不适用之处。

第 10 章说明服务虚拟化如何帮助性能测试。它包括一个虚拟化公共可用 API 的完整例子。

关于代码

本书使用了一些代码示例来帮助说明这些概念。其中一些是假设的(见第 4 章),一些是基于虚拟化的真实公共 API(见第 6 章和第 10 章),还有一些是非常笼统的(见第 8 章)。我尽力让这些例子在服务虚拟化可以解决的各种问题中通俗易懂,这不是件容易的事。有些问题很容易理解,但有些,比如虚拟化返回二进制数据的 .NET Remoting 服务,则不好理解。我希望保持足够的幽默感,让你对容易出现的问题保持兴趣,对于复杂的行为,给你足够的感觉,让你有能力自己创新。

本书的源代码可在 <https://github.com/bbyars/mountebank-in-action> 上下载,也可扫描封底二维码获取。

图书论坛

购买《Mountebank 微服务测试》包括免费访问 Manning 出版社运行的私人 Web 论坛,你可以在该论坛上对本书发表评论、提出技术问题以及获得作者和其他用户的帮助。要访问论坛,请访问 www.manning.com/books/testing-microservices-with-mountebank。还可以在 <https://forums.manning.com/forums/about> 了解有关 Manning 论坛和行为准则的更多信息。

Manning 对读者的承诺是提供使读者之间以及读者和作者之间进行有意义的对话的场所。作者对论坛的贡献仍然是自愿的，但这不代表作者承诺任何具体的参与数量。我们建议你问作者一些有挑战性的问题，来引起他的兴趣。

目 录

第 I 部分 起 步

第 1 章 测试微服务	3
1.1 微服务刷新器	4
1.1.1 微服务路径	5
1.1.2 微服务和组织结构	7
1.2 端到端测试的问题	9
1.3 了解服务虚拟化	10
1.3.1 使用 API 逐个设置 测试	12
1.3.2 使用持久数据存储	12
1.3.3 记录和重放	13
1.4 mountebank 介绍	14
1.5 服务虚拟化工具生态 系统	18
1.6 本章小结	19
第 2 章 体验 mountebank	21
2.1 设置示例	22
2.2 HTTP 和 mountebank: 入门	23
2.3 虚拟化产品目录服务	26
2.4 第一个测试	31
2.5 本章小结	38

第 II 部分 使用 mountebank

第 3 章 使用屏蔽响应进行测试	43
3.1 屏蔽响应的基础	44
3.1.1 默认响应	45
3.1.2 了解默认响应的工作 方式	48
3.1.3 更改默认响应	49
3.1.4 循环响应	50
3.2 HTTPS imposter	52
3.2.1 设置可信的 HTTPS imposter	55
3.2.2 使用相互身份验证	57
3.3 在配置文件中保存响应	58
3.4 本章小结	62
第 4 章 使用谓词发送不同的 响应	65
4.1 谓词基础	66
4.1.1 谓词的类型	68
4.1.2 匹配对象请求字段	73
4.1.3 deepEquals 谓词	74
4.1.4 匹配多值字段	75
4.1.5 exists 谓词	77
4.1.6 连接点	78

4.1.7	谓词类型的完整列表	79	6.5	本章小结	129
4.2	参数化谓词	80	第 7 章	添加行为	131
4.3	在 JSON 值中使用谓词	80	7.1	理解行为	131
4.3.1	使用直接 JSON 谓词	81	7.2	修饰一个响应	132
4.3.2	使用 JSONPath 选择 JSON 值	81	7.2.1	使用 decorate 函数	133
4.4	选择 XML 值	83	7.2.2	为保存的代理响应添加 修饰	134
4.5	本章小结	86	7.2.3	通过 shellTransform 添加 中间件	137
第 5 章	添加记录/重放行为	87	7.3	为响应添加延迟	139
5.1	设置代理	87	7.4	多次重复响应	140
5.2	生成正确的谓词	91	7.5	替换响应中的内容	141
5.2.1	使用 predicateGenerators 创建谓词	91	7.5.1	将请求数据复制到 响应	141
5.2.2	添加谓词参数	93	7.5.2	从外部数据源查找 数据	148
5.3	为同一请求捕获多个响应	96	7.6	完整的行为列表	152
5.4	重放代理的方法	100	7.7	本章小结	152
5.5	配置代理	101	第 8 章	协议	153
5.5.1	使用相互身份验证	102	8.1	协议在 mountebank 中的 工作方式	154
5.5.2	添加自定义标题	103	8.2	TCP 入门	155
5.6	代理用例	104	8.3	存根基于文本的 TCP RPC	156
5.6.1	使用代理作为回退	104	8.3.1	创建基本的 TCP imposter	157
5.6.2	将 HTTPS 转换为 HTTP	106	8.3.2	创建 TCP 代理	159
5.7	本章小结	107	8.3.3	匹配和操作 XML 载荷	160
第 6 章	mountebank 编程	109	8.4	二进制支持	161
6.1	创建自己的谓词	110	8.4.1	使用二进制模式进行 Base64 编码	162
6.2	创建自己的动态响应	115			
6.2.1	添加状态	116			
6.2.2	添加异步	118			
6.2.3	确定响应与谓词注入之间 的关系	127			
6.3	注意：安全很重要	127			
6.4	调试提示	128			

8.4.2	在二进制模式下使用谓词.....	163	
8.5	虚拟化.NET 远程服务.....	164	
8.5.1	创建简单的.NET Remoting 客户端.....	164	
8.5.2	虚拟化.NET Remoting 服务器.....	167	
8.5.3	如何告诉 mountebank 消息结束的位置.....	171	
8.6	本章小结.....	175	
 第Ⅲ部分 关闭循环			
 第 9 章 微服务的安全.....			179
9.1	一个持续的交付刷新器.....	179	
9.1.1	基于微服务的 CD 测试策略.....	181	
9.1.2	将测试策略映射到部署管道.....	183	
9.2	创建测试管道.....	184	
9.2.1	创建单元测试.....	186	
9.2.2	创建服务测试.....	189	
9.2.3	平衡服务虚拟化与合约测试.....	193	
9.2.4	探索性测试.....	196	
9.3	本章小结.....	198	
 第 10 章 mountebank 性能测试.....			199
10.1	为什么服务虚拟化支持性能测试.....	200	
10.2	定义你的场景.....	202	
10.3	捕获测试数据.....	203	
10.3.1	捕获响应.....	204	
10.3.2	捕捉实际延迟.....	206	
10.3.3	模拟随机的延迟波动.....	207	
10.4	运行性能测试.....	207	
10.5	扩展 mountebank.....	210	
10.6	本章小结.....	212	

第 I 部分

起 步

“欢迎你，朋友”。

我家门前的迎宾垫上也写着这句话，当访问 mountebank 网站(<https://www.mbstest.org>)时，你第一眼看到的也是这些词语。我非常希望你在本书中第一次读到的也是这些词语，因为我特别欢迎你来到服务虚拟化的精彩世界，尤其是 mountebank 的精彩世界。

第 I 部分旨在提供引言相关的背景知识，主要介绍将 mountebank 作为测试和持续交付堆栈的一部分。服务虚拟化背后的一个主要驱动因素是计算日益增长的分布式特性，第 1 章首先对微服务进行概述，重点介绍它改变测试软件的方式。然后介绍如何将服务虚拟化置于上下文中，并对 mountebank 的主要组件进行了详述。

第 2 章展示了 mountebank 的作用。当使用服务虚拟化编写第一个测试，让它提供一个简单的启动点用来研究第 2 章中 mountebank 的全部功能时，你可能会感到有些力不从心。

测试微服务

本章主要内容：

- 微服务的简要背景
- 测试微服务面临的挑战
- 服务虚拟化如何简化测试
- mountebank 简介

有时候，需要以假乱真。

当 Web 开始与企业组织中的桌面应用程序竞争时，我就已经开始开发软件了。虽然基于浏览器的应用程序带来了极大的部署优势，但我们还是使用几乎相同的方式来对其进行测试。我们编写了一个单片机应用程序，将其连接到数据库，像用户测试应用程序一样来测试它。我们测试了一个真实的应用程序。

测试驱动的开发教会我们，良好的面向对象设计能够在更细粒度的层级上进行测试。我们可以单独测试类和方法，并在快速迭代中获得反馈。依赖注入——传入类的依赖性，而不是按需实例化它们——使我们的代码更加灵活并且更具可测试性。只要传入了与实际测试接口相同的测试依赖性，就可以完全隔离想要测试的代码位。通过向代码中注入伪依赖项，我们对编写的代码有了更多信心。

不久，聪明的开发人员开发了开放源码库，使创建这些伪依赖项变得更容易，从而使我们能够专注于讨论更重要的事情，例如如何调用它们。我们根据自己的测试风格形成了小团体：模拟主义者陶醉于使用模拟的纯洁性；古典主义者则骄傲地坚持自己对存根的完全依赖^[1]。但是双方都没有对伪依赖项测试的基本价值

[1] 你可能将时间花费在比阅读古典主义者和模拟主义者之间的差异更有趣的事情上，但如果你自己无法弄明白，可以访问 <http://martinfowler.com/articles/mocksArentStubs.html> 来获取更多内容。