

TURING

图灵程序设计丛书

MANNING



Spring Boot in Action

Spring Boot 实战

【美】Craig Walls 著

丁雪丰 译

全面分析Spring Boot核心功能和特性
掌握高效Java开发利器，快速构建微服务



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Spring Boot 实战

Spring Boot in Action

[美] Craig Walls 著

丁雪丰 译

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

Spring Boot 实战 / (美) 克雷格·沃斯
(Craig Walls) 著 ; 丁雪丰译. -- 北京 : 人民邮电出版社, 2016.9
(图灵程序设计丛书)
ISBN 978-7-115-43314-5

I. ①S… II. ①克… ②丁… III. ①JAVA语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2016)第193197号

内 容 提 要

本书以 Spring 应用程序开发为中心, 全面讲解如何运用 Spring Boot 提高效率, 使应用程序的开发和管理更加轻松有趣。作者行文亲切流畅, 以大量示例讲解了 Spring Boot 在各类情境中的应用, 内容涵盖起步依赖、Spring Boot CLI、Groovy、Grails、Actuator。对于 Spring Boot 开发应用中较为繁琐的内容, 附录奉上整理完毕的表格, 一目了然, 方便读者查阅。

本书适合全体 Java 开发人员。

-
- ◆ 著 [美] Craig Walls
译 丁雪丰
责任编辑 朱 巍
执行编辑 张 憬
责任印制 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 14
字数: 331千字 2016年9月第1版
印数: 1-3 500册 2016年9月河北第1次印刷
著作权合同登记号 图字: 01-2016-4638号

定价: 59.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

版 权 声 明

Original English language edition, entitled *Spring Boot in Action* by Craig Walls, published by Manning Publications. 178 South Hill Drive, Westampton, NJ 08060 USA. Copyright © 2016 by Manning Publications.

Simplified Chinese language edition, copyright © 2016 by Posts & Telecom Press, All rights reserved.

本书中文简体字版由Manning Publications授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

时光回到2004年，Spring Framework 1.0正式发布，同年，Rod Johnson的*Expert one-on-one J2EE Development without EJB*一经出版就震撼了整个Java世界。不知不觉，12年就这么过去了，Spring已然成为Java应用开发的事实标准，影响着无数Java开发者。

刚才打开Spring的官网，已经能看到Spring Framework 5.0.0 SNAPSHOT的身影了，而Spring的家族也早就不再是Spring Framework一枝独秀，Spring Data、Spring Batch、Spring Security等一大堆名字让人看得眼花缭乱。其中最引人瞩目的无疑就是Spring Boot了，它正是本书的主角。

Spring Boot从无数知名企业的实践中吸取经验，总结并落实到框架中。如果说Spring Framework的目标是帮助开发者写出好的系统，那Spring Boot的目标就是帮助开发者用更少的代码，更快地写出好的生产系统。

Spring Boot为开发者带来了更好的开发体验，但写完代码只是万里长征路上的一小步，后续的运维工作才是让很多人真正感到无助的。Spring Boot在运维方面做了很多工作，部署、监控、度量，无一不在其涉猎范围之内，结合Spring Cloud后还可以轻松地实现服务发现、服务降级等功能。

2014年，Spring Source的Josh Long在向我介绍Spring Boot时，我不断重复一句话：“这个功能我们也做了。”的确，国内的百度、阿里、腾讯，国外的Amazon、Facebook、Twitter、Netflix等一票大公司都在框架和系统建设上有大量投入，为了提升性能和可用性，大家做了很多卓有成效的工作。现在，Spring Boot让人人都能享受业内顶级公司的“福利”，站在巨人的肩膀之上，想想都让人觉得兴奋。

说起为何想要翻译本书，那只能说是缘分使然。笔者当年在机缘巧合之下与Spring结缘，也因它结识了很多朋友。毫不夸张地说，是Spring开启了我的作译者生涯，先后参与了Spring官方文档、《Spring专业开发指南》和《Spring攻略》的翻译。

本以为在完成了30岁前每年翻译一本书的目标后，我应该不会再去翻译什么东西了，甚至在向图灵的编辑推荐本书时，我都没有想到最后会是自己来翻译这本书。不得不感叹一声，缘分就是如此妙不可言的东西。相信后续Spring Boot会有更好地发展，因为它牢牢抓住了开发者的需求。Craig的《Spring实战》已经到了第4版，本书应该也会有第2版，此时此刻正捧着本书的您会成为它的译者吗？至少让我们一起来为自己喜欢的技术贡献一份力量吧。

丁雪丰

2016年7月于上海

序

2014年春天，Netflix的交付工程团队开始着手实现一个伟大的目标——通过一个软件平台来实现端到端的全局持续交付，该平台有利于系统的可扩展性及弹性。为了满足Netflix的交付与部署需要，我的团队曾构建了两套不同的应用程序，但这两套应用程序都有演变成庞然大物的趋势，而且都没能满足灵活性和弹性的目标。更重要的是，这些庞大的应用程序最终还拖了我们的后腿，让我们跟不上合作伙伴的创新步伐。用户开始回避我们的工具，而不是使用它们。

很明显，如果想要向公司证明自己的真正价值并快速创新，我们需要把庞然大物分解成小的独立服务，这些服务要能随时发布。拥抱微服务架构给我们带来了希望，让我们能实现灵活性与弹性的双重目标。但是我们需要在一个可靠的基础上实现这一架构，它要能实现真正的并发、合理的监控、可靠易用的服务发现，运行时还要有极好的性能。

我们要在JVM上寻找一款框架，它要直接提供快速开发的能力和强大的运维能力。最终，我们找到了Spring Boot。

Spring Boot能用寥寥数行代码构建一套基于Spring并满足生产要求的服务，不费吹灰之力！实际上，一个简单的Spring Boot Hello World应用程序能放进一条推文里，这在短短几年之前还是完全不可能的事情。它还自带了不少非功能性的特性，比如安全、度量、健康检查、内嵌服务器和外置配置，这些都让选择Spring Boot成为了一件顺理成章的事情。

然而，踏上Spring Boot之旅后，我们却发现手头没有好的文档。要搞明白怎么利用好框架的特性，只能依靠源码，这可不是个让人愉快的办法。

Manning那本著名的《Spring实战》的作者再度接受挑战，将Spring Boot的核心用法写成了另一本好书，对此我一点都不吃惊。毫无疑问，Craig和Manning的团队又做成了一件了不起的大事！正如我们所料，《Spring Boot实战》是一本通俗易懂的好书。

从第1章引人入胜的介绍以及富有传奇色彩的90字符推文应用程序，一直到第7章对Spring Boot的Actuator（提供了很多生产应用程序所需的神奇的运维特性）的深度分析，《Spring Boot实战》做到了知无不言，言无不尽。实际上，对我而言，第7章对Actuator的深度分析解答了不少问题，这些问题自一年多以前我开始使用Spring Boot后，就一直萦绕在我的脑海里。第8章对部署选项的透彻研究让我大开眼界，了解到Cloud Foundry在云部署方面是如此简便。第4章是我最喜欢的章节之一，Craig揭示了很多强大的选项，它们能很方便地测试Spring Boot应用程序。从一开始我就惊喜于Spring的测试特性，而Spring Boot将它们发挥得淋漓尽致。

正如上文中我所说的那样，Spring Boot正是十几年来Java社区所探寻的那种框架。它那简单

易用的开发特性和开箱即用的运维能力，让Java开发再度趣味横生。我欣然向大家宣布，Spring和Spring Boot已经成为了Netflix新持续交付平台的基础。而且，Netflix的其他团队也参考了我们的做法，因为他们也看到了Spring Boot的巨大益处。

我怀着兴奋与激动的心情，向大家强烈推荐Craig的书。作为Spring Boot的文档，本书可谓通俗易懂、趣味横生，是Spring Boot征服Java社区后，大家翘首以盼的佳作。Craig浅显易懂的写作风格，对Spring Boot核心特性与功能的全面分析，一定能让读者对Spring Boot有个彻底的认识(而且在满心欢喜的同时还肃然起敬)。

Craig加油！Manning出版社加油！那些开发出Spring Boot的天才开发者们加油！请你们一定坚持下去！正是你们确保了JVM的光明未来。

Andrew Glover
Netflix交付工程团队经理

前 言

在1964年的纽约世界博览会上，沃特·迪士尼向世界介绍了三件有开创意义的东西：“小小世界”（it's a small world）、“与林肯先生共度的伟大时刻”（Great Moments with Mr. Lincoln）以及“文明演进之旋转木马”（Carousel of Progress）。^①所有这三样东西随后都搬进了迪士尼乐园和迪士尼世界，你今天仍能看见它们。

其中，我最喜欢的是“文明演进之旋转木马”，这大约也是沃特·迪士尼的最爱之一。这既是骑行，又是舞台表演，座位区域围绕着中心区域旋转，上演四场表演，讲述了一个家庭在20世纪不同时代（分别是20世纪初、20世纪20年代、20世纪40年代和近年）的故事，突出了不同年代技术的进步。从手摇洗衣机，到电灯和收音机，到自动洗碗机和电视，再到电脑和声控家电，无一不在述说着创新的故事。^②

在每幕表演中，父亲（也是演出的叙述者）都会讲述最新的发明，并带上一句“这玩意儿不能更好了”，到头来却发现随着技术的进步，它的确变得更好了。

比起这场舞台演出，Spring的历史要短得多。但是对于Spring，我的感受和“演进老爹”（Progress Dad）对20世纪的体会相似。似乎每个Spring应用程序都让开发者的生活更上一个台阶，仅从Spring组件的声明和织入方式就能看出端倪。让我们来看看Spring历史中的一些演化历程。

- Spring 1.0的出现彻底改变了我们开发企业级Java应用程序的方式。Spring的依赖注入与声明式事务意味着组件之间再也不存在紧耦合，再也不用重量级的EJB了。这玩意儿不能更好了。
- 到了Spring 2.0，我们可以在配置里使用自定义的XML命名空间，更小、更简单易懂的配置文件让Spring本身更便于使用。这玩意儿不能更好了。
- Spring 2.5让我们有了更优雅的面向注解的依赖注入模型（即@Component和@Autowired注解），以及面向注解的Spring MVC编程模型。不用再去显式地声明应用程序组件了，也不再需要去继承某个基础的控制类了。这玩意儿不能更好了。
- 到了Spring 3.0，我们有了一套基于Java的全新配置，它能够取代XML。在Spring 3.1里，一系列以@Enable开头的注解进一步完善了这一特性。终于，我们第一次可以写出一个没有任何XML配置的Spring应用程序了。这玩意儿不能更好了。

① 关于这届世博会里迪士尼相关的信息，详见<http://www.dwz.cn/2Hrvyh>中的Disney influence部分。——译者注

② 关于这个游乐设施，详见<http://www.yesterland.com/progress.html>的介绍。——译者注

□ Spring 4.0对条件化配置提供了支持，根据应用程序的Classpath、环境和其他因素，运行时决策将决定使用哪些配置，忽略哪些配置。那些决策不需要在构建时通过编写脚本确定了；以前会把选好的配置放在部署的包里，现在情况不同了。这玩意儿不能更好了。

现在轮到Spring Boot了。虽然Spring的每个版本都让我们觉得一切都不能更好了，但Spring Boot还是向我们证明了Spring仍然有巨大的潜力。事实上，我相信Spring Boot是长久以来Java开发历程里最意义深刻、激动人心的东西。

以历代Spring Framework的进步为基础，Spring Boot实现了自动配置，这让Spring能够智能探测正在构建何种应用程序，自动配置必要的组件以满足应用程序的需要。对于那些常见的配置场景，不再需要显式地编写配置了，Spring会替你料理好一切。

选择在构建时和运行时要包含在应用程序里的库，往往要花费不少工夫，而Spring Boot的起步依赖（starter dependency）将常用依赖聚合在一起，借此简化一切。它不仅简化了你的构建说明，还让你不必苦思冥想特定库和版本。

针对使用Groovy来开发Spring应用程序，Spring Boot的命令行界面提供了一个令人瞩目的选项，它将Java应用程序开发过程中的噪声降到最低，开发方式平易近人。有了Spring Boot CLI，就不再需要访问方法了，不再需要诸如public与private之类的访问修饰符，也不再需要分号或者return关键字。在许多场景中，import语句都可以去掉。因为你是正在命令行里以脚本方式运行应用程序，所以连构建说明都能免了。

Spring Boot的Actuator让你能一窥应用程序运行时的内部工作细节，看看Spring应用程序上下文里都有哪些Bean，Spring MVC控制器是怎么与路径映射的，应用程序都能取到哪些配置属性，诸如此类。

Spring Boot为我们带来了这么多奇妙的特性，这玩意儿当然不能更好了！

本书中你将看到，Spring Boot着实让Spring比以前更好了。我们将一同去了解自动配置、Spring Boot起步依赖、Spring Boot CLI和Actuator。我们还会去摆弄一下Grails的最新版本，它就是基于Spring Boot的。临近末尾，你也许会觉得Spring不可能更好了。

如果说迪士尼的“文明演进之旋转木马”告诉了我们什么事情，那就是当我们觉得什么东西不可能更好了的时候，它一定会变得更好。Spring Boot的进步正在带来越来越大的益处。真的难以想象Spring还能变得更好，但它肯定会更好。毫无疑问，Spring的前景总是美好的。

关于本书

Spring Boot旨在简化Spring的开发，就这点而论，Spring Boot涉及了Spring的方方面面。用一本书讲清楚Spring Boot的所有用法是不可能的，因为这必须涵盖Spring本身所支持的各种技术。所以《Spring Boot实战》把Spring Boot大致分为4个主题：自动配置、起步依赖、命令行界面和Actuator。书中还会讲到一些必要的Spring特性，但重点还是在Spring Boot上。

《Spring Boot实战》面向的是全体Java开发者。虽然读者需要有一些Spring背景，但Spring Boot让那些新接触Spring的人也更容易上手。然而，因为本书的重点是Spring Boot，不会深入Spring本身，所以手边再准备一本Spring读物也许效果会更好，比如说《Spring实战（第4版）》。

章节安排

《Spring Boot实战》全书分为8章。

- 第1章会对Spring Boot进行概述，内容涵盖最基本的自动配置、起步依赖、命令行界面和Actuator。
- 第2章会进一步深入Spring Boot，重点介绍自动配置和起步依赖。在这一章里，你将用很少的显式配置来构建一个完整的Spring应用程序。
- 第3章是对第2章的补充，演示了如何通过设置应用程序属性来改变自动配置，或者在自动配置无法满足需要时彻底覆盖它。
- 在第4章里我们会看到如何为Spring Boot应用程序编写自动化集成测试。
- 在第5章里你将看到一种有别于传统Java开发方式的做法，Spring Boot CLI能让你通过命令行来运行应用程序，这个应用程序完全是由Groovy脚本构成的。
- 讲到Groovy，第6章会介绍Grails 3，这是Grails框架的最新版本，它基于Spring Boot。
- 在第7章里你将看到如何通过Spring Boot的Actuator了解运行中的应用程序，以及它是如何工作的。你还会看到如何使用Actuator的Web端点、远程shell和JMX MBean对应用程序一窥究竟。
- 第8章讨论了各种部署Spring Boot应用程序的方法，包括传统的应用程序服务器部署和云部署。

编码规范及代码下载

书中包含了很多代码示例，这些代码使用了等宽字体，如DispatcherServlet。正文中出现的所有类名、方法名或者是XML片段也会用这种字体。不少Spring的类和包的名字都特别长（但是一目了然），因此在需要时会使用续行符（`\`）。书中的代码并非都是完整的，通常我只会就某个特定主题摘出类中的一两个方法。

你可以在Manning出版社的网站上下载书中应用程序的完整代码，地址是www.manning.com/books/spring-boot-in-action。

作者在线

购买本书的读者还能免费访问Manning出版社的私有Web论坛，在那里你能就本书发表评论，询问技术问题，向作者以及其他用户寻求帮助。如需访问并订阅该论坛，请打开浏览器访问www.manning.com/books/spring-boot-in-action。该页面提供了详细的信息，告诉你在注册后如何访问论坛，论坛里都能提供哪些帮助，以及论坛的管理规则。

Manning向读者承诺，为读者与读者之间以及读者与作者之间的沟通建立桥梁。但Manning并不保证作者在论坛中的参与程度，他们在论坛上投入多少精力是全凭自愿的（并且是无偿的）。我们强烈建议你向作者问些有挑战性的问题，让他有兴趣留在论坛里。

只要本书仍在销售，你就能在出版商的网站上查找作者在线论坛及其讨论归档。

关于封面图

本书封面上的插画题为“喀山鞑靼民族服饰”（Habit of a Tartar in Kasan），喀山是俄罗斯联邦鞑靼斯坦共和国首府。这幅图选自Thomas Jefferys的《各国古代和现代服饰集》（*A Collection of the Dresses of Different Nations, Ancient and Modern*，共四卷，1757—1772年间出版于伦敦），该书扉页中谈到，这些插画都是手工上色、铜版雕刻，还用了阿拉伯树胶。Thomas Jefferys（1719—1771）被誉为“乔治三世国王的御用地理学家”（Geographer to King George III）。他是一名英国地图制图师，是当时地图行业的领导者。他为政府和其他官方机构雕刻并印刷地图，还制作了各种不同的商用地图和地图集，尤其是北美洲地图。地图制图师的工作引发了他调研当地民族服饰的兴趣，这一兴趣在这套服饰集里体现得淋漓尽致。

着迷于远方的大陆，为了消遣而去旅行，这在18世纪晚期还是相对新鲜的现象，而像这套服饰集这样的合集在当时非常流行，向观光客和足不出户的“游客”介绍其他国家的居民。Jefferys著作中异彩纷呈的图画生动地描绘了200年前世界各国的特色。自那以后，服饰文化发生了变化，各个国家与地区之间一度非常丰富的多样性已逐渐消失。现在，不同大洲的居民往往很难通过服饰来分辨了。也许，我们该乐观一点儿，我们用文化和视觉上的多样性换来了更多样的人生，或者说是更多样、更有趣、更智能的科技人生。

在很难从外观上分辨不同计算机读物的年代里，Manning出版社脱颖而出，在图书封面上采用了两个世纪以前各地居民丰富多样的形象，以此体现了计算机行业别出心裁、独具创新特性。这些都得归功于Jeffreys的绘画。

电子书

扫描如下二维码，即可购买本书电子版。



致 谢

本书将告诉你Spring Boot如何自动处理应用程序幕后的各种杂事，让你专注于做那些使应用程序独特的工作。从很多方面来说，这和本书的诞生经历非常类似。很多人帮我操心了不少事情，让我能专心撰写本书的内容。我要感谢Manning出版社的Cynthia Kane、Robert Casazza、Andy Carroll、Corbin Collins、Kevin Sullivan、Mary Piergies、Janet Vail、Ozren Harlovic以及Candace Gillhoolley，他们做了很多幕后工作。

编写测试能让你知道自己的软件是否实现了目标。同样，很多人在本书撰写过程中就审稿并提供了反馈意见，他们让我确信本书没有偏离方向。为此，我要感谢Aykut Acikel、Bachir Chihani、Eric Kramer、Francesco Persico、Furkan Kamaci、Gregor Zurowski、Mario Arias、Michael A. Angelo、Mykel Alvis、Norbert Kuchenmeister、Phil Whiles、Raphael Villela、Sam Kreter、Travis Nelson、Wilfredo R. Ronsini Jr.以及William Fly。还要特别感谢John Guthrie在原稿即将付印前的最终技术审校，也感谢Andrew Glover为本书作序。

当然，如果没有Spring团队中各位天才成员的杰出工作，本书就不可能也不必问世。你们太棒了！能成为改变软件开发方式的团队的成员，我十分激动。

我还要感谢所有参加了No Fluff/Just Stuff活动^①的人，无论是演讲嘉宾还是出席的听众。我们之间的对话某种程度上也促成了本书。

没有那些组成文字的字母，像这样的书也不可能出现。因此，就和我之前的书一样，我想借此机会感谢发明第一个字母表的腓尼基人。

最后，我要隆重感谢我的挚爱，我美丽的妻子Raymie，还有我了不起的女儿Maisy和Madi。你们又一次忍受我从事一个写作项目。现在，书写完了，我们该去迪士尼世界了，你们说呢？

^① 这是一个面向JVM软件开发者的活动，详见<https://nofluffjuststuff.com>。——译者注

目 录

| | | | |
|--|----|---------------------------------|----|
| 第 1 章 入门 | 1 | 3.2 通过属性文件外置配置 | 49 |
| 1.1 Spring 风云再起 | 1 | 3.2.1 自动配置微调 | 50 |
| 1.1.1 重新认识 Spring | 2 | 3.2.2 应用程序 Bean 的配置外置 | 55 |
| 1.1.2 Spring Boot 精要 | 3 | 3.2.3 使用 Profile 进行配置 | 59 |
| 1.1.3 Spring Boot 不是什么 | 6 | 3.3 定制应用程序错误页面 | 62 |
| 1.2 Spring Boot 入门 | 6 | 3.4 小结 | 64 |
| 1.2.1 安装 Spring Boot CLI | 7 | 第 4 章 测试 | 66 |
| 1.2.2 使用 Spring Initializr 初始化 Spring Boot 项目 | 10 | 4.1 集成测试自动配置 | 66 |
| 1.3 小结 | 18 | 4.2 测试 Web 应用程序 | 68 |
| 第 2 章 开发第一个应用程序 | 19 | 4.2.1 模拟 Spring MVC | 69 |
| 2.1 运用 Spring Boot | 19 | 4.2.2 测试 Web 安全 | 72 |
| 2.1.1 查看初始化的 Spring Boot 新项目 | 21 | 4.3 测试运行中的应用程序 | 74 |
| 2.1.2 Spring Boot 项目构建过程 解析 | 24 | 4.3.1 用随机端口启动服务器 | 75 |
| 2.2 使用起步依赖 | 27 | 4.3.2 使用 Selenium 测试 HTML 页面 | 76 |
| 2.2.1 指定基于功能的依赖 | 28 | 4.4 小结 | 78 |
| 2.2.2 覆盖起步依赖引入的传递依赖 | 29 | 第 5 章 Groovy 与 Spring Boot CLI | 80 |
| 2.3 使用自动配置 | 30 | 5.1 开发 Spring Boot CLI 应用程序 | 80 |
| 2.3.1 专注于应用程序功能 | 31 | 5.1.1 设置 CLI 项目 | 81 |
| 2.3.2 运行应用程序 | 36 | 5.1.2 通过 Groovy 消除代码噪声 | 81 |
| 2.3.3 刚刚发生了什么 | 38 | 5.1.3 发生了什么 | 85 |
| 2.4 小结 | 41 | 5.2 获取依赖 | 86 |
| 第 3 章 自定义配置 | 42 | 5.2.1 覆盖默认依赖版本 | 87 |
| 3.1 覆盖 Spring Boot 自动配置 | 42 | 5.2.2 添加依赖仓库 | 88 |
| 3.1.1 保护应用程序 | 43 | 5.3 用 CLI 运行测试 | 89 |
| 3.1.2 创建自定义的安全配置 | 44 | 5.4 创建可部署的产物 | 91 |
| 3.1.3 掀开自动配置的神秘面纱 | 48 | 5.5 小结 | 91 |
| 3.2 通过属性文件外置配置 | 49 | 第 6 章 在 Spring Boot 中使用 Grails | 93 |
| 3.2.1 自动配置微调 | 50 | 6.1 使用 GORM 进行数据持久化 | 93 |
| 3.2.2 应用程序 Bean 的配置外置 | 55 | | |
| 3.2.3 使用 Profile 进行配置 | 59 | | |
| 3.3 定制应用程序错误页面 | 62 | | |
| 3.4 小结 | 64 | | |

| | | | |
|-------------------------------------|-----|---------------------------|-----|
| 6.2 使用 Groovy Server Pages 定义视图 | 98 | 7.4.3 添加自定义度量信息 | 129 |
| 6.3 结合 Spring Boot 与 Grails 3 | 100 | 7.4.4 创建自定义跟踪仓库 | 132 |
| 6.3.1 创建新的 Grails 项目 | 100 | 7.4.5 插入自定义健康指示器 | 134 |
| 6.3.2 定义领域模型 | 103 | 7.5 保护 Actuator 端点 | 136 |
| 6.3.3 开发 Grails 控制器 | 104 | 7.6 小结 | 138 |
| 6.3.4 创建视图 | 105 | 第 8 章 部署 Spring Boot 应用程序 | 139 |
| 6.4 小结 | 107 | 8.1 衡量多种部署方式 | 139 |
| 第 7 章 深入 Actuator | 108 | 8.2 部署到应用服务器 | 140 |
| 7.1 揭秘 Actuator 的端点 | 108 | 8.2.1 构建 WAR 文件 | 141 |
| 7.1.1 查看配置明细 | 109 | 8.2.2 创建生产 Profile | 142 |
| 7.1.2 运行时度量 | 115 | 8.2.3 开启数据库迁移 | 145 |
| 7.1.3 关闭应用程序 | 121 | 8.3 推上云端 | 150 |
| 7.1.4 获取应用信息 | 121 | 8.3.1 部署到 Cloud Foundry | 150 |
| 7.2 连接 Actuator 的远程 shell | 122 | 8.3.2 部署到 Heroku | 153 |
| 7.2.1 查看 <code>autoconfig</code> 报告 | 123 | 8.4 小结 | 155 |
| 7.2.2 列出应用程序的 Bean | 124 | 附录 A Spring Boot 开发者工具 | 157 |
| 7.2.3 查看应用程序的度量信息 | 124 | 附录 B Spring Boot 起步依赖 | 163 |
| 7.2.4 调用 Actuator 端点 | 125 | 附录 C 配置属性 | 169 |
| 7.3 通过 JMX 监控应用程序 | 126 | 附录 D Spring Boot 依赖 | 202 |
| 7.4 定制 Actuator | 128 | | |
| 7.4.1 修改端点 ID | 128 | | |
| 7.4.2 启用和禁用端点 | 129 | | |

本章内容

- Spring Boot简化Spring应用程序开发
- Spring Boot的基本特性
- Spring Boot工作区的设置

Spring Framework已有十余年的历史了，已成为Java应用程序开发框架的事实标准。在如此悠久的历史背景下，有人可能会认为Spring放慢了脚步，躺在了自己的荣誉簿上，再也做不出什么新鲜的东西，或者是让人激动的东西。甚至有人说，Spring是遗留项目，是时候去看看其他创新的东西了。

这些人说得不对。

Spring的生态圈里正在出现很多让人激动的新鲜事物，涉及的领域涵盖云计算、大数据、无模式的数据持久化、响应式编程以及客户端应用程序开发。

在过去的一年多时间里，最让人兴奋、回头率最高、最能改变游戏规则的东西，大概就是Spring Boot了。Spring Boot提供了一种新的编程范式，能在最小的阻力下开发Spring应用程序。有了它，你可以更加敏捷地开发Spring应用程序，专注于应用程序的功能，不用在Spring的配置上多花功夫，甚至完全不用配置。实际上，Spring Boot的一项重要工作就是让Spring不再成为你成功路上的绊脚石。

本书将探索Spring Boot开发的诸多方面，但在开始前，我们先大概了解一下Spring Boot的功能。

1.1 Spring 风云再起

Spring诞生时是Java企业版（Java Enterprise Edition，JEE，也称J2EE）的轻量级代替品。无需开发重量级的Enterprise JavaBean（EJB），Spring为企业级Java开发提供了一种相对简单的方法，通过依赖注入和面向切面编程，用简单的Java对象（Plain Old Java Object，POJO）实现了EJB的功能。

虽然Spring的组件代码是轻量级的，但它的配置却是重量级的。一开始，Spring用XML配置，而且是很多XML配置。Spring 2.5引入了基于注解的组件扫描，这消除了大量针对应用程序自身