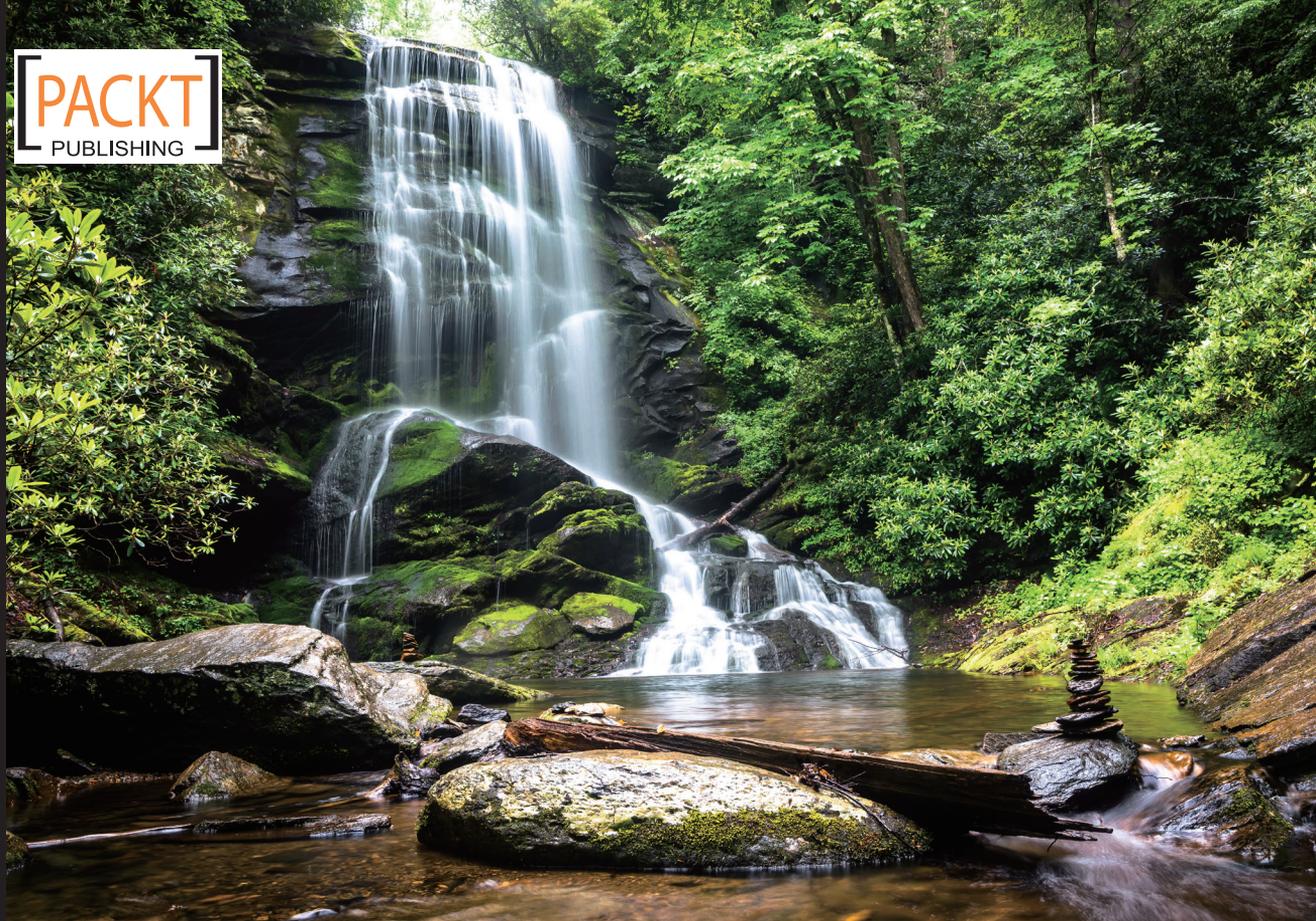


PACKT
PUBLISHING



通过学习Spring MVC框架达到设计真实Web应用的专业水准

精通Spring MVC 4

Mastering Spring MVC 4

[美] Geoffroy Warin 著
张卫滨 孙丽文 译

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS



精通Spring MVC 4

[美] Geoffroy Warin 著
张卫滨 孙丽文 译

人民邮电出版社

图书在版编目 (C I P) 数据

精通Spring MVC 4 / (美) 瓦林 (Geoffroy Warin)
著 ; 张卫滨, 孙丽文译. -- 北京 : 人民邮电出版社,
2017. 5
ISBN 978-7-115-44758-6

I. ①精… II. ①瓦… ②张… ③孙… III. ①JAVA语
言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第038777号

-
- ◆ 著 [美] Geoffroy Warin
 - 译 张卫滨 孙丽文
 - 责任编辑 陈冀康
 - 责任印制 焦志炜

 - ◆ 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>
 - 北京鑫正大印刷有限公司印刷

 - ◆ 开本: 800×1000 1/16
 - 印张: 18
 - 字数: 350千字 2017年5月第1版
 - 印数: 1-3000册 2017年5月北京第1次印刷
- 著作权合同登记号 图字: 01-2016-7045号
-

定价: 59.00元

读者服务热线: (010)81055410 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

内容提要

Spring MVC 属于 Spring Framework 的衍生产品，已经融合在 Spring Web Flow 里面。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。Spring MVC 4 是当前最新的版本，在众多特性上有了进一步的提升。

本书中从头开始构建了一个完整的 Web 应用。全书共 10 章，分别介绍了快速搭建 Spring Web 应用、精通 MVC 结构、处理表单和复杂的 URL 映射、文件上传与错误处理、创建 RESTful 应用、保护应用、单元测试与验收测试、优化请求、将 Web 应用部署到云等内容，循序渐进地讲解了 Spring MVC 4 的开发技巧。

本书最适合已经熟悉 Spring 编程基础知识并迫切希望扩展其 Web 技能的开发人员。通过阅读本书，读者将深度把握 Spring MVC 的各项特性及实用技巧。

译者序

Spring MVC 是 Spring 社区项目的重要组成部分，目前用到了无数规模各异的项目之中。Spring 以简化企业级应用开发为己任，十多年来的发展历史证明，他们确实做到了这一点。不论是 Web 应用开发、数据库访问还是应用集成、大数据处理，我们都能看到 Spring 的相关项目，借助 IoC 与 AOP 的核心理念，Spring 让开发人员的工作更加轻松愉悦。

Spring MVC 本身的发展也是如此，从最初的 XML 文件配置，到后来的注解和 Java 代码方式的配置，都是尽可能地让开发过程更加便利，同时，Spring MVC 还在不断完善其测试功能，推动优秀编码实践能够更容易地落地。

Spring Boot 项目的横空出世更是引起了空前的关注，这是一个改变游戏规则的项目，它能够极大地简化配置，高效地管理依赖，并且与当下流行的微服务架构模式契合良好，得到了广泛地应用。本书组合使用了 Spring Boot 和 Spring MVC，从项目的搭建过程入手，涉及页面开发、文件上传、应用安全、RESTful API 开发、测试等，涵盖了 Web 应用开发的方方面面，最后还介绍了如何将应用部署到云端，不管你关心哪个领域，希望通过本书都能获取到有用的知识。

从《Spring 实战》（第三版）开始，我参与翻译了多本 Spring 相关的书籍，结识了许多志同道合的好友，这也是一种特殊的缘分。技术的发展日新月异，作为从业者的我们自然不能停下前进的脚步，唯有不断学习、不断思考、不断实践，才能保证不被时代所淘汰，希望与诸位读者共勉。

读者在阅读本书的过程中，如果发现错误或问题，请不吝指正，您可以通过 levinzhang1981@126.com 或微博@张卫滨 1895 联系我。

希望本书能够为您的工作和学习带来帮助。

张卫滨

2017 年 2 月于大连

前言

作为 Web 开发人员，我愿意创建新的东西，将它们快速上线，然后将注意力转移到下一个新的想法上。

如今，所有的应用都互相连接在了一起，我们需要与社交媒体进行交互，促进产品和复杂系统的发展，为用户提供更大的价值。

直到现在，这些对于 Java 开发人员来说都非常遥远和复杂。随着 Spring Boot 的诞生和云平台的平民化，我们可以在有限的时间内创建精彩的应用并让所有的人来访问，而这个过程不需要花一分钱。

在本书中，我们将会从头开始构建一个有用的 Web 应用。这个应用有很多很棒的特性，如国际化、表单校验、分布式会话与缓存、社交登录、多线程编程等。

同时，我们还会对其进行完整的测试。

在本书结束之前，我们还会将这个小应用部署到云端，使它能够通过 Web 进行访问。

如果你觉得这挺有意思的话，那么就别浪费时间了，马上开始着手编码吧！

本书所涵盖的内容

第 1 章“快速搭建 Spring Web 应用”能够让我们非常快速地开始使用 Spring Boot。本章介绍了让我们更具生产效益的工具，如 Spring Tool Suite 和 Git，本章还会帮助我们搭建应用的主体框架，并见识 Spring Boot 背后的魔力。

第 2 章“精通 MVC 架构”指导我们创建一个小的 Twitter 搜索引擎，同时，本章还涵

盖了 Spring MVC 和 Web 架构的基础知识。

第 3 章“处理表单和复杂的 URL 映射”帮助你理解如何创建用户基本信息表单，本章介绍如何在服务端和客户端校验数据，并且让我们的应用支持多语言访问。

第 4 章“文件上传与错误处理”将会指导你为基本信息表单添加文件上传功能，它阐述了如何在 Spring MVC 中恰当地处理错误并展示自定义的错误页面。

第 5 章“创建 RESTful 应用”阐述了 RESTful 架构的理念，它还帮助我们创建了一个可以通过 HTTP 调用的用户管理 API，这个过程中会看到帮助我们设计 API 的工具，并且会讨论如何很简便地实现文档化。

第 6 章“保护应用”将会指导我们如何保护应用，包括如何使用基本 HTTP 认证保护 RESTful API，以及如何保护登录页之后的 Web 页面，它阐述了如何通过 Twitter 进行登录以及如何将会话保存在 Redis 中，从而允许我们的应用进行扩展。

第 7 章“单元测试与验收测试”帮助我们对应用进行测试。它讨论了测试与 TDD，介绍了如何对控制器进行单元测试，如何使用现代的库设计端到端的测试。最后，介绍了 Groovy 如何提升测试的生产效率和可读性。

第 8 章“优化请求”对应用进行了优化。它包括缓存控制和 Gzip，本章将教会我们如何把 Twitter 搜索结果缓存到内存和 Redis 中，以及如何对搜索实现多线程执行。除此之外，还会介绍如何实现 Etag 和使用 WebSocket。

第 9 章“将 Web 应用部署到云中”会指导我们对应用进行部署，通过对比，阐述了不同 PaaS 解决方案的差异。然后，介绍了如何将应用部署到 Cloud Foundry 和 Heroku 中。

第 10 章“超越 Spring Web”在整体上讨论了 Spring 生态系统，介绍了现代 Web 应用的组成部分以及后续的发展方向。

阅读本书所需的前提条件

尽管我们将要构建的是一个很高级的应用，但是并不需要你安装很多的东西。

我们将要构建的应用需要 Java 8。

我们并不强制你使用 Git，不过你绝对应该使用 Git 来对自己的应用进行版本控制。如果你希望将应用部署到 Heroku 上，那么会需要用到它。另外，借助 Git 可以非常容易地回顾你的工作，通过查看代码的差异和历史来了解其演进过程。在第 1 章中包含了很多开始使用 Git 的资源。

我还推荐你使用一个好的 IDE。我们会看到如何使用 Spring Tool Suite（免费）和 IntelliJ Idea（一个月的免费试用）实现快速起步。

如果你使用 Mac 的话，应该了解一下 Homebrew（<http://brew.sh>）。通过使用这个包管理器，你可以安装本书中提到的所有工具。

本书为谁而作

本书最适合已经熟悉 Spring 编程基础知识并迫切希望扩展其 Web 技能的开发人员。建议你事先掌握一些 Spring 框架的知识。

约定

在本书中，你会看到多种样式的文本，它们用来区分不同类型的信息。如下是这些风格的样例以及对它们的描述。

文本中的代码、数据库表名、文件夹的名称、文件名、文件扩展名、路径名称、伪 URL（dummy URL）、用户输入以及 Twitter 如下所示。

“你会在目录 `build/libs` 中找到 JAR 文件”。

代码片段如下所示：

```
public class ProfileForm {
    private String twitterHandle;
    private String email;
    private LocalDate birthDate;
    private List<String> tastes = new ArrayList<>();

    // getters and setters
}
```

如果你希望你关注特定的代码片段，那么相关的行或条目会粗体显示：

```
public class ProfileForm {
    private String twitterHandle;
    private String email;
    private LocalDate birthDate;
    private List<String> tastes = new ArrayList<>();
}
```

```
    // getters and setters  
}
```

命令行的输入和输出会按照如下所示进行编写：

```
$ curl https://start.spring.io
```

新术语和关键字会粗体显示。在屏幕、菜单或对话框中看到的内容将会按照如下的样例显示：“转到新建项目菜单并选择 **Spring Initializr** 项目类型”。



警告或关键的提示按照这种方式显示。



小技巧或窍门按照这种方式显示。

读者反馈

我们欢迎读者的反馈。请告诉我们你觉得这本书怎么样——你喜欢什么，不喜欢什么。读者的反馈是非常重要的，这将有助于我们开发出读者得到真正想要的图书。

如果是一般的反馈，只需发送邮件到 feedback@packtpub.com 即可，并在邮件的主题中包含本书的名称。

如果对于某个话题，你有专业的知识，并且愿意编写一本书或者为某本书做出贡献，那么可以参考我们的读者指南：www.packtpub.com/authors。

客户支持

如果你购买了 Packt 图书，那么我们会从很多方面为你提供帮助，尽可能最大化你的购买所带来的价值。

示例代码下载

你可以通过账号，在 <http://www.packtpub.com> 下载所有你所购买的 Packt 图书的示例代码文件。如果你是在其他途径购买的本书，那么可以访问 <http://www.packtpub.com/support>

并进行注册，这些文件会直接邮件发送给你。

对于本书的样例代码，你也可以通过 <https://github.com/Mastering-Spring-MVC-4/mastering-spring-mvc4> 进行下载。

答疑

如果你对本书有任何的问题，可以通过 questions@packtpub.com 联系我们，我们将尽全力为你解答。

作者简介

Geoffroy Warin 从 10 岁就开始编程了，是软件匠艺（Software Craftsmanship）运动的坚定信奉者和开源的倡导者，他跟随自己的内心选择成为一名开发人员并对其坚定不移。在职业生涯中，他一直致力于使用 Java 和 JavaScript 语言开发企业级的 Web 应用。

Geoffroy 在前端和后端方面都游刃有余，他非常关注代码整洁和可测试性。他深信开发人员应该尽其所能编写出可读性更强的代码，这样的代码能够持续地为用户交付价值。

在推动测试驱动开发和创建伟大的软件设计方面，他的核心工具就是结对编程和导师制。

他还讲授 Java Web 技术栈的课程，是 Groovy 和 Spring 的狂热支持者。

最近，他还担任《Learning Spring Boot》和《Spring Boot Cookbook》等书的审阅者，这两本书都是由 Packt 出版社所出版，涵盖了 Spring 生态系统中新增的主要内容。

如果想学习最新的 Spring 和 JavaScript 编程技巧，请查阅 Geoffroy 的博客 (<http://geowarin.github.io>) 和他的 Twitter 账号 (<https://twitter.com/geowarin>)。

技术审校者简介

Raymundo Armendariz 是有超过 10 年经验的软件开发人员，他之前主要致力于为 Java 和 .NET 平台构建软件，但是现在他投身于 JavaScript。

他还是一本 JavaScript 微框架图书的作者。

在他的职业生涯中，大部分的工作都是与汽车行业相关，他曾经就职的公司包括 Autozone、Alldata、TRW 和 1A Auto。

他是《Getting Started with Backbone Marionette》（Packt 出版社）一书的作者，这本书可以在 <https://www.packtpub.com/web-development/getting-started-backbone-marionette> 找到。

我要感谢朋友们对我的支持和帮助。

—Raymundo

Abu Kamruzzaman 是纽约城市大学的 Web 程序员和数据库分析师。在过去的 10 多年以来，他所开发和维护的 Web 应用作为班级教学和注册课程，用于高校的教学。从 2014 年 11 月开始，他担任 CUNY 总部的 PeopleSoft 开发专家，目前，他所从事的项目是与商业智能团队（Business Intelligence）一起使用 OBIEE 为 CUNY 构建数据仓库。在加入总部之前，自 2001 年以来，他曾经在 CUNY 的各个校区工作过，还讲授研究生和本科生的 IT 课程，指导学生们使用他所开发的应用。从 2001 年开始，他所讲授的课程包括 J2EE、DBMS、数据仓库、面向对象编程、Web 设计以及 Web 编程。他是柏鲁克学院（Baruch College）杰克林商学院（Zicklin School of Business）计算机信息系统的教员之一。他非常热心于教育事业，对开源项目也有着极大的热情，如 Hadoop、Hive、Pig、NoSQL 数据库、Java、云

计算以及移动应用开发。他从布鲁克林学院（Brooklyn College）/CUNY 获得了硕士学位，从宾厄姆顿大学（Binghamton University）/SUNY 计算机专业获得了学士学位。他的 Web 站点地址：<http://faculty.baruch.cuny.edu/akamruzzaman/>。

我要感谢美丽的妻子 Nicole Woods，感谢她对我所从事事业的耐心、支持以及鼓励。感谢我的父母，感谢他们的祝福和祈祷。感谢本书的作者和 Packt 出版团队能够给我参与本书的机会。

— Abu

Jean-Pol Landrain 是软件工程专业的学士，从 1998 年开始就主要从事面向网络、实时和分布式计算的工作。他逐渐成为一名软件架构师，具有超过 17 年的面向对象编程经验，尤其擅长 C++、Java/JEE、各种应用服务器、操作系统（Windows 和 Linux）以及相关的技术。

他目前就职于 Agile Partner，这是一家位于卢森堡的 IT 咨询公司，从 2006 年以来一直致力于敏捷方法论的推广、教育和应用。在过去的 5 年中，他参与了针对欧洲议会（European Parliament）开发团队的工具与技术方案的选择与验证。

他与 Packt 出版社协作，审校了《HornetQ Messaging Developer's Guide》，与 Manning 出版社合作，审校了《Docker in Action》《Git in Practice》《ActiveMq in Action》以及《Spring in Action》（第一版）。

首先，我要感谢我的妻子 Marie Smets 以及 9 岁的女儿 Phoebe，感谢他们理解我对技术的热爱以及在这上面所花费的时间。我还要感谢在 Agile Partner 的朋友和同事，如果一个人单枪匹马地投身于技术会非常乏味的，感谢他们所增添的乐趣。

很不幸的是，在这本书制作期间，我失去了我的祖父 André Landrain 和祖母 Hédène Guffens，我非常悲痛。因为这些私人的事件导致了一些延迟，非常感谢 Packt 出版社编辑团队的耐心，尤其要感谢该项目的协调者 Nidhi Joshi 和本书的作者 Geoffroy Warin。他们做得非常棒，我完全信任本书的品质。在 Spring MVC 方面，这是市面上很好的一本书。

— Jean-Pol

Wayne Lund 是 Pivotal 的 PaaS 和现场工程师。在企业级软件开发和分布式环境方面具有超过 25 年的经验，主要的方向是 Spring、企业级 Java、Groovy 和 Grails，并将这些技术扩展至使用 Smalltalk 和 C++ 的系统，他非常热衷于个性化和新兴的技术。他的目标是继续享受下一代的技术，这就是 PaaS 以及新的 Spring 工具集，这包括构建在 Spring Boot、Spring

Cloud 和 Spring XD 技术上的原生云应用，这样的技术方案能够启用快数据（Fast Data）、大数据、社交以及移动等特性。

目前，他就职于 Pivotal，关注云、数据以及敏捷的结合，之前曾经在一家财富 500 强的医疗保健公司和一家全球性的大型咨询公司工作多年。

他还参与出版了 Packt 出版社的《Learning Spring Application Development》一书。

目录

第 1 章 快速搭建 Spring Web 应用	1
1.1 Spring Tool Suite 简介	2
1.2 IntelliJ 简介	7
1.3 start.Spring.io 简介	8
1.4 命令行方式简介	8
1.5 那就正式开始吧	9
1.5.1 Gradle 构建	11
1.5.2 让我们看一下代码	15
1.6 幕后的 Spring Boot	17
1.6.1 分发器和 multipart 配置	17
1.6.2 视图解析器、静态资源以及区域配置	22
1.7 错误与转码配置	24
1.8 嵌入式 Servlet 容器 (Tomcat) 的配置	27
1.8.1 HTTP 端口	29
1.8.2 SSL 配置	29
1.8.3 其他配置	29
1.9 小结	30
第 2 章 精通 MVC 架构	32
2.1 MVC 架构	32
2.2 对 MVC 的质疑及其最佳实践	33
2.2.1 贫血的领域模型	33
2.2.2 从源码中学习	35
2.3 Spring MVC 1-0-1	35
2.4 使用 Thymeleaf	36
2.5 Spring MVC 架构	40
2.5.1 DispatcherServlet	40
2.5.2 将数据传递给视图	41
2.6 Spring 表达式语言	42
2.7 结束 Hello World, 开始获取 Tweet	44
2.7.1 注册应用	44
2.7.2 搭建 Spring Social Twitter	46
2.7.3 访问 Twitter	46
2.8 Java 8 的流和 lambda 表达式	48
2.9 使用 WebJars 实现质感设计	49
2.9.1 使用布局	52
2.9.2 导航	54
2.10 检查点	59
2.11 小结	59
第 3 章 处理表单和复杂的 URL 映射	60
3.1 基本信息页——表单	60
3.2 校验	68
3.2.1 自定义校验信息	70

3.2.2 用于校验的自定义注解	73	5.6.2 浏览器中的 RESTful 客户端	130
3.3 国际化	74	5.6.3 httpie	131
3.3.1 修改地域	76	5.7 自定义 JSON 输出	131
3.3.2 翻译应用的文本	79	5.8 用户管理 API	136
3.3.3 表单中的列表	81	5.9 状态码与异常处理	140
3.4 客户端校验	84	5.9.1 带有状态码的 ResponseEntity	140
3.5 检查点	86	5.9.2 使用异常来处理状态码	142
3.6 小结	87	5.10 通过 Swagger 实现文档化	146
第 4 章 文件上传与错误处理	88	5.11 生成 XML	148
4.1 上传文件	88	5.12 检查点	149
4.1.1 将图片写入到响应中	93	5.13 小结	150
4.1.2 管理上传属性	94	第 6 章 保护应用	151
4.1.3 展现上传的图片	97	6.1 基本认证	151
4.1.4 处理文件上传的错误	99	6.1.1 用户授权	152
4.2 转换错误信息	102	6.1.2 URL 授权	155
4.3 将基本信息放到会话中	103	6.1.3 Thymeleaf 安全标签	156
4.4 自定义错误页面	107	6.2 登录表单	158
4.5 使用矩阵变量进行 URL 映射	108	6.3 Twitter 认证	163
4.6 将其组合起来	114	6.3.1 搭建社交认证环境	164
4.7 检查点	121	6.3.2 详解	167
4.8 小结	122	6.4 分布式会话	169
第 5 章 创建 RESTful 应用	123	6.5 SSL	171
5.1 什么是 REST	123	6.5.1 生成自签名的证书	172
5.2 Richardson 的成熟度模型	124	6.5.2 单一模式	173
5.2.1 第 0 级——HTTP	124	6.5.3 双通道模式	173
5.2.2 第 1 级——资源	124	6.5.4 置于安全的服务器之后	174
5.2.3 第 2 级——HTTP 动作	124	6.6 检查点	175
5.2.4 第 3 级——超媒体控制	126	6.7 小结	175
5.3 API 版本化	127	第 7 章 单元测试与验收测试	176
5.4 有用的 HTTP 代码	127	7.1 为什么要测试我的代码	176
5.5 客户端为王	128	7.2 该如何测试自己的代码	177
5.6 调试 RESTful API	130		
5.6.1 JSON 格式化扩展	130		

7.3 测试驱动开发	178	8.8 检查点	244
7.4 单元测试	179	8.9 小结	245
7.5 验收测试	180	第 9 章 将 Web 应用部署到云中	246
7.6 第一个单元测试	180	9.1 选择主机	246
7.7 Mock 与 Stub	184	9.1.1 Cloud Foundry	246
7.7.1 使用 Mockito 进行 mock	184	9.1.2 OpenShift	247
7.7.2 在测试时 Stub bean	186	9.1.3 Heroku	248
7.7.3 该使用 Mock 还是 Stub	189	9.2 将 Web 应用部署到 Pivotal Web Services 中	248
7.8 对 REST 控制器进行单元测试	189	9.2.1 安装 Cloud Foundry CLI 工具	248
7.9 测试认证	196	9.2.2 装配应用	249
7.10 编写验收测试	198	9.2.3 激活 Redis	252
7.10.1 Gradle 配置	198	9.3 将 Web 应用部署到 Heroku 中	253
7.10.2 第一个 FluentLenium 测试	200	9.3.1 安装工具	254
7.10.3 使用 FluentLenium 创建页面对象	206	9.3.2 搭建应用	255
7.10.4 用 Groovy 实现测试	209	9.3.3 Heroku profile	256
7.10.5 使用 Spock 进行单元测试	210	9.3.4 运行应用	257
7.10.6 使用 Geb 进行集成测试	213	9.3.5 激活 Redis	258
7.10.7 在 Geb 中使用页面对象	215	9.4 改善应用的功能	260
7.11 检查点	218	9.5 小结	261
7.12 小结	220	第 10 章 超越 Spring Web	262
第 8 章 优化请求	221	10.1 Spring 生态系统	262
8.1 生产环境的 profile	221	10.1.1 核心	263
8.2 Gzip	222	10.1.2 执行	263
8.3 缓存控制	222	10.1.3 数据	263
8.4 应用缓存	224	10.1.4 其他值得关注的项目	264
8.4.1 缓存失效	229	10.2 部署	264
8.4.2 分布式缓存	230	10.3 单页面应用	265
8.5 异步方法	231	10.3.1 参与者	265
8.6 ETag	237	10.3.2 未来的前景	266
8.7 WebSocket	241	10.3.3 实现无状态	267
		10.4 小结	267