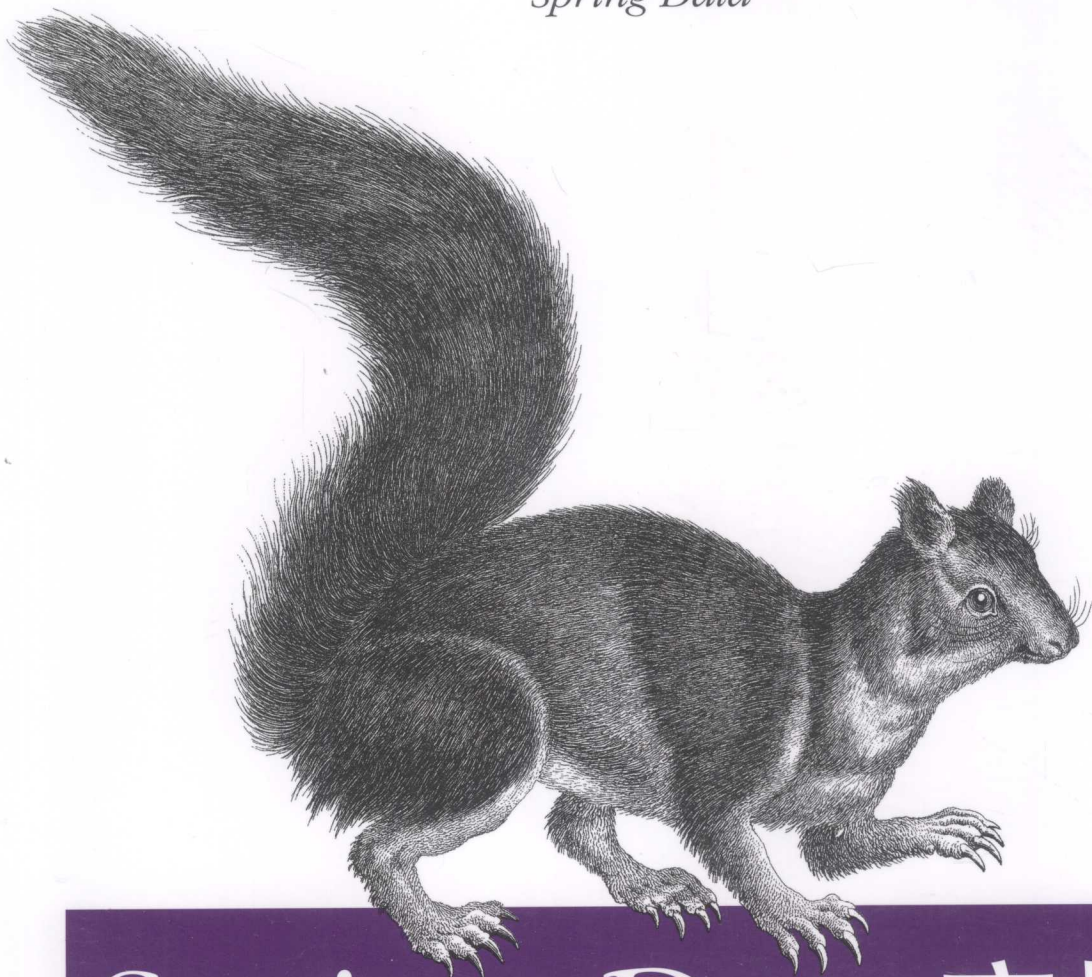


Spring Data



Spring Data 实战

[美] *Mark Pollack Oliver Gierke*
Thomas Risberg Jon Brisbin
Michael Hunger 著
张卫滨 文建国 译

O'REILLY®

人民邮电出版社
POSTS & TELECOM PRESS

014036013

TP312JA
1572

O'REILLY

Spring Data 实战

【美】Mark Pollack, Oliver Gierke, Thomas Risberg,
Jon Brisbin, Michael Hunger 著

张卫滨 文建国 译



TP312JA
1572

人民邮电出版社



北航 C1715623

01020010

图书在版编目 (C I P) 数据

Spring Data 实战 / (美) 波拉克 (Pollack, M.) 等著; 张卫滨, 文建国译. — 北京: 人民邮电出版社, 2014. 4

ISBN 978-7-115-34370-3

I. ①S… II. ①波… ②张… ③文… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第000746号

版权声明

Copyright © 2013 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2014. Authorized translation of the English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体字版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

-
- ◆ 著 [美] Mark Pollack Oliver Gierke Thomas Risberg
Jon Brisbin Michael Hunger
 - 译 张卫滨 文建国
 - 责任编辑 杜洁
 - 责任印制 程彦红 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>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1000 1/16
印张: 17
字数: 338千字 2014年4月第1版
印数: 1-3000册 2014年4月河北第1次印刷
- 著作权合同登记号 图字: 01-2013-3679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: 京崇工商广字第0021号

内容提要

数据访问领域正在发生一场变革，关系型数据库无法解决的问题需要新的数据访问技术来解决。Spring Data 项目就是一种简化 Java 应用构建的数据访问技术，它可以帮助开发人员高效地使用最新的数据处理和管理工具，同时还能够以最新的方式使用传统的数据库。

本书从 Spring Data 背景知识、关系型数据库、NoSQL、快速应用开发、大数据、数据网格 6 个方面深度解析了数据访问技术，介绍的内容都是最流行和前沿的，其中文档数据库、图数据库、键 / 值存储、Hadoop 以及 GemFire 数据结构等是最重要的内容。本书介绍了 Spring Data、Repository、Querydsl 的基础理念，然后阐述了借助 Spring Data 如何简化 NoSQL 和大数据的访问，并且涵盖了使用 Spring Roo 和 Spring Data REST 导出功能如何实现应用的快速开发，除此之外，书中还涉及与其他 Spring 子项目的协同工作，如 Spring Integration 和 Spring Batch。

本书面向实战、结构清晰，示例丰富，适用于各类 Java 开发人员和数据库开发人员，也可以作为各大、中专院校相关专业师生的参考用书和相关培训机构的培训教材。

感谢我的妻子 Daniela、我的儿子 Gabriel 和 Alexandre，因为他们的宽容才使得我能够挤出时间来撰写这本书。

—Mark Pollack

我要感谢我的家人、朋友、音乐家以及迄今为止与我共同奋斗的每位同事；还有带来这段精彩旅程的整个 Spring Data 和 SpringSource 团队；最后，事实上首先要感谢的是 Sabine，她给了我无尽的关爱和支持。

—Oliver Gierke

致我的妻子 Carol 和我的儿子 Alex，感谢你们让我的人生多姿多彩，并给予我无限的支持和鼓励。

—Thomas Risberg

致我的妻子 Tisha，我的儿子 Jack、Ben 和 Daniel，我的女儿 Morgan 和 Hannah，感谢你们对我的关爱、支持和容忍。如果没有你们，所有的这一切都没有意义。

—Jon Brisbin

特别感谢 Rod 和 Emil 启动了 Spring Data 项目，特别感谢 Oliver，他使得这个项目变得意义非凡。感谢我的家人，你们总是非常支持我疯狂工作；有如此善解人意的女人在我身边，我心怀感激。

—Michael Hunger

我要感谢我的妻子 Nanette 和我的孩子们，他们给予我无限的支持、宽容和理解。同样要感谢 Rod 和 Spring Data 团队的同事们，他们使得这一切成为可能。

—David Turanski

序

我们生活在一个很有意思的时代，新的业务流程驱动着新的需求。我们以前认为理所应当的事情正面临着被颠覆的威胁——其中有一条就是关系型数据库是实现持久化的默认可选方案。尽管这个理念还被广泛认可，但是对于如何更高效地适应新世界，还远远没有明晰。

众多可选的数据存储方案导致了碎片化。Java 开发人员以前会将应用中的数据存储到关系型数据库之中，相对于此，新的存储形式需要开发者付出更多的努力才行。本书将会帮助你解决这个现实问题。它能够帮助你全面了解当前的硬件条件下存储领域的现状，并阐述对于现代的业务问题，NoSQL 为什么那么重要。

由于保守的企业级市场对语言的认同度较低（也可能是因为 Java 对象-关系映射（ORM）解决方案的复杂性），Java 开发人员传统上往往不会涉及到 NoSQL 领域。还好，现在状况发生了变化，进而产生了这本重要和及时的图书。Spring Data 是一个重要的项目，它能够帮助开发人员征服新的挑战。

Spring 带来了众多价值，因此其成为企业级 Java 开发人员首选的平台，在持久化方案碎片化的现状下它能够带来特殊的收益。Spring 所提供的一部分价值就在于能够以一致的方式集成不同的技术（并不会将所有人拉低到一个最低的通用标准）。独特的“Spring 方式（Spring way）”有助于降低开发人员的学习曲线并简化代码的维护。如果你已经熟悉 Spring，就会发现 Spring Data 能够帮助你探索和采用并不是非常熟悉的数据存储形式。如果你对 Spring 还不熟悉的话，这是一个很好的机会去了解 Spring 如何简化代码并使其保持更加一致。

本书作者是最有资格阐述 Spring Data 的人，因为他们就是项目的领导者。他们对 Spring 有深刻的了解，同时将多种现代数据存储紧密结合了起来。他们很好地阐述了 Spring Data 背后的驱动力以及如何延续 Spring 一贯的数据访问方式。另外，本书涵盖了 Spring Data 如何与其他 Spring 组成部分协同工作这部分内容，如 Spring Integration 和 Spring Batch。书中还提供了除 Spring 之外其他很有意义的内容，如 Repository 的理念、类型安全查询的优点以及为什么 Java 持久化 API（Java Persistence API，JPA）并不适合作为通用的数据访问解决方案。

尽管本书主要讨论的是数据访问而不是 NoSQL 的使用，但是你会发现很有价值的 NoSQL 资料，因为它所介绍的概念和代码可能是你并不熟悉的。所有的内容都是最新的，重要的内容包括文档数据库、图数据库、键 / 值存储、Hadoop 以及

Gemfire 数据结构。

我们这些编程人员通常追求实用，并且亲手实践时能够学得更好。因此，本书充分考虑到了实用性。在开始的时候，作者首先介绍了如何在业界领先的集成开发环境（Integrated Development Environment, IDE）中运行示例代码，甚至包含了方便学习的屏幕截图，还阐述了数据库驱动方面的需求以及数据库的基本安装过程。他们将代码托管在了 GitHub 上，这一点我很赞赏，这样每个人都可以对其进行访问和浏览。本书涵盖了众多的主题，所以要设计良好的样例，以便将所有内容尽量关联起来。

本书有一章介绍了 Spring Roo，这也能够证明他们是非常重视实际开发的，Spring Roo 是 Spring 提供的快速应用开发（Rapid Application Development, RAD）解决方案。大多数 Roo 的用户已经熟悉如何将其用于传统的 JPA 架构之中，作者展示了 Roo 所带来的生产率提升不仅限于关系型数据库。

读完本书之后，你会深刻理解现代的数据访问为何更加专门化和碎片化、NoSQL 数据存储的主要分类、Spring Data 如何帮助 Java 开发人员新的环境下高效工作以及对于感兴趣的话题要到什么地方去寻找深入学习的资料。但最为重要的是，这是探索代码的绝佳起点。

Rod Johnson

Spring 框架创始人

前言

数据访问领域现状概述

数据访问领域在过去的 7 年间发生了重要的变化。过去 30 年间一直占据企业级数据存储和处理核心位置的关系型数据库已经不能再独领风骚了。在过去的 7 年间诞生了很多可选的数据存储形式，当然也有的面临着消亡，它们被使用到了带有关键任务的企业级应用程序之中。这些新的数据存储形式是为了解决特定的数据访问问题而设计的，使用关系型数据库通常无法高效地解决这些问题。

将关系型数据库推到拐点的一个问题就是扩展性（scale）。试问，我们如何将几百甚至几千 TB（terabyte）的数据存储到关系型数据库中？这个问题让我想到了一个笑话，病人说：“大夫，我一这样动就疼”而医生则说：“那就别这样动呗！”暂且把笑话放在一边，存储如此巨量数据的推动力是什么呢？在 2001 年，IDC 报告说“人们创造和复制的数据将会超过 1.8ZB（zettabytes），而且每隔两年就会翻番¹”。新的数据涵盖各种类型，如媒体文件、日志文件、传感器数据（RFID、GPD、遥测设备……）、Twitter 上的消息以及 Facebook 上的帖子。尽管对于企业来讲存储于关系型数据库中的数据依然非常重要，但是这些新型的数据并没有存储在关系型数据库之中。

一般用户所关注的需求是存储大量的媒体文件，而企业却发现存储和分析这些新型数据的重要性。在美国，各行各业的公司所存储的数据超过了 100TB，有的公司的数据甚至超过了 1PB（petabyte）²。大家的共识是持续地分析这些数据会为商业上带来明显的收益。例如，如果产品能够自动汇报其状况，那么公司就可以更容易地掌握产品的行为。为了更好地理解客户，公司在决策制定的过程中可以吸收社交媒体的数据。这甚至引起了主流媒体的报道，例如，Orbitz 发现 Mac 用户偏好较为昂贵的酒店（<http://on.wsj.com/UhSINi>），而 Target 会预测其客户家生孩子的时间（<http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html>），从而能够在公开的出生记录发布之前给客户邮寄优惠券。

大数据（Big Data）通常指的是这样一种流程：存储大量的数据、保持其原始格式、持续地进行分析并且会与其他的数据源结合起来提供某个领域更深入的理解，这种

¹ IDC; *Extracting Value from Chaos* (<http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>) .2011

² IDC; US Bureau of Labor Statistics

领域可能是商业上的也可能是自然科学上的。

很多的公司和科研实验室在大数据这个词流行起来之前就开始这样做了。当前的过程与以往的不同在于，智能的数据分析所带来的价值要高于硬件的成本。现在执行这种类型的分析不再需要购买4万美元一颗的CPU了；商用的硬件集群中每颗CPU的价格是1000美元。对于大型的数据集，存储区域网络（Storage Area Network, SAN）以及网络附属存储（Network Attached Storage, NAS）的价格较为昂贵：每GB（gigabyte）是1~10美元，如果复本构建到数据库中而不是硬件之中，那本地磁盘的成本每GB只有0.05美元。对于商用的硬件集群，使用本地磁盘的数据传输率也要比基于SAN或NAS的系统更高——对于相同价格的系统，前者能快500倍。在软件方面，新的数据访问技术大多数都是开源的。尽管开源并不意味着零成本，但是这显然会降低使用门槛，相对于传统的商业软件，它们能够降低采购的整体成本。

另一个能够区分新型数据存储与关系型数据库的问题域就是关系型的数据模型。如果你想分析上百万人的社交图谱，图形数据库更接近这个领域的模型，使用这种数据库难道不是很自然的事情吗？如果需求持续地要求你修改关系型数据库管理系统（RDBMS）的模式（schema）以及对对象关系映射（ORM）层，那该怎么办呢？可能“无模式（schema-less）”的文档数据库能够减少对象映射的复杂度，相对于僵化的关系型模型，它所提供的系统更易于演化。尽管每种不同的数据库各有其独特之处，但是可以基于其数据模型进行大致分类。基本的分类情况如下：

键/值

我们所熟悉的数据模型，类似于哈希表（hashtable）。

列族

扩展的键/值数据模型，值的数据类型也可以是键/值对的序列。

文档

半结构化数据的集合，如XML或JSON。

图

基于图论，数据模型中包括节点（node）和边（edge），它们都可以包含属性（property）。

这些新的数据库都可以归类在“**NoSQL数据库**”之下。回顾历史，这个名字尽管朗朗上口，但是并不精确，因为它容易让人觉得这些数据库不能进行查询，但事实并不是这样。它的基本含义是摆脱关系型数据模型以及关系型数据库的**ACID**特性（原子性、一致性、隔离性以及持久性）。

要摆脱ACID特性的主要驱动力在于，很多的应用程序提高了可扩展写（scaling write）的优先级，并且希望即便系统的某一部分失效，其他部分依然可以继续运作。尽管在关系型数据库中，可以通过在数据库之前使用内存缓存来实现可扩展读，但是进行可扩展写要困难得多。为了标识这类应用程序，通常将其命名为“BASE”系统，在这里缩写代表着基本可用（basically available）、可扩展性（scalable）、最终一致性（eventually consistent）。具有键/值数据模型的分布式数据网格并没有归类到这种新的NoSQL数据库之中。然而，它们提供了与NoSQL数据库类似的特性，如数据的可扩展性以及组合计算能力和数据的分布式计算功能。

从上面简短的介绍中，你能够了解到数据访问的现状，目前正在发生的是一场革命，关注数据的人会非常兴奋。关系型数据库并没有消亡，在很多企业的运作中它依然是核心，并且会持续很长的时间。但是，趋势很明显：新的数据访问技术解决了关系型数据库所无法解决的问题，因此作为开发人员，我们必须扩充自己的技能，要能够处理这两种技术。

Spring 框架长期以来都致力于简化 Java 应用程序的开发，尤其是使用 Java 数据库连接（Java Database Connectivity, JDBC）或对象关系映射器编写基于 RDBMS 的数据访问层方面。在本书中，我们力图帮助开发人员使用这些新技术高效地编写 Java 应用程序。Spring Data 项目直接处理这些新的技术，因此你能够将已有的 Spring 知识延伸到它们之中，或者通过使用 Spring Data，也能够更深入地学习 Spring。不过，我们也没有抛弃关系型数据库。Spring Data 为了 Spring 能支持 RDBMS 扩展了新功能。

如何阅读本书

本书希望你提供实用的Spring Data项目的介绍，该项目的目的在于帮助Java开发人员使用最新的数据处理和管理工具，同时能够以最新的方式使用传统的数据库。我们首先会为你介绍这个项目，阐述SpringSource以及该团队背后的驱动力，并描述了示例工程的领域模型，这个例子将会在后面的章节中使用，此外还介绍了如何获取和搭建示例代码（第1章）。

接着将会讨论Spring Data Repository的基本理念，因为它们是后续各种数据存储中都会涉及的通用主题（第2章）。对于Querydsl也是如此，我们将会在第3章中讨论这个话题。这两章为探讨特定存储与Repository的集成以及高级查询功能奠定了坚实的基础。

为了从Java开发人员熟悉的领域开始，我们会用一些时间来介绍传统的持久化技术，如JPA（第4章）和JDBC（第5章）。这两章阐述了在Spring已有的对JPA和JDBC的支持之上，Spring Data的模块提供了什么新特性。

在完成这些之后，将会引入Spring Data项目所支持的NoSQL存储：以MongoDB为例介绍文档数据库（第6章），以Neo4j为例讲解图数据库（第7章），以Redis为例讲解键/值存储（第8章）。列族数据库HBase将会在稍后的章节中介绍（第12章）。这

些章节中涵盖的内容包括了将领域类映射到特定存储的数据结构、通过所提供的应用程序编程接口（Application Programming Interface, API）与存储进行便捷地交互以及对Repository抽象的使用。

我们将会介绍Spring Data REST导出器（第10章）以及Spring Roo集成（第9章）。这两章的内容都基于Repository抽象并且能够很容易地将Spring Data所管理的实体导出到Web之中，可以作为表述性状态转移（Representational State Transfer, REST），也可以作为基于Spring Roo所构建的Web应用程序。

接下来本书将会进入大数据的世界——具体来说也就是Hadoop和Spring for Apache Hadoop，我们将会介绍如何使用Hadoop技术实现样例以及Spring Data模块如何明显简化Hadoop的使用（第11章）。基于这些内容，将会使用Spring Batch和Spring Integration项目来构建复杂的大数据管道——这些项目都非常适合于大数据的处理（第12章和第13章）。

最后一章将会讨论Spring Data对GemFire的支持，这是一个分布式的数据网格解决方案（第14章）。

本书中的约定

本书中使用了如下的排版约定：

斜体字 (*Italic*)

表示新的术语、URL、电子邮箱、文件名以及文件扩展名。

等宽字体 (Constant width)

用于程序清单，以及在段落中对程序中元素的引用，如变量或函数名、数据库、数据类型、环境变量、语句和关键字。

加粗的等宽字体 (Constant width bold)

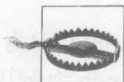
表示需要用户输入的命令或其他文本。

斜体的等宽字体 (*Constant width italic*)

表示这些文本需要根据用户提供的值或上下文确定的值进行替换。



这个图标代表提示、建议或一般说明。



这个图标代表警告或提醒。

代码示例

这本书的目的在于帮助你做好工作。一般来说，可以在程序或文档中使用本书的代码，除非复制了本书中大部分的代码，否则不需要获得我们的许可。例如，如果你写了一个程序，使用了本书中的几段代码是不需要许可的。销售和发行刻有O'Reilly图书中示例的光盘（CD-ROM）则需要得到许可。如果引用本书和利用书中的示例代码来回答问题是不需要许可的。但是，如果在你的产品文档中大量使用本书的示例代码，那么就需要许可了。

我们赞赏大家在使用代码时注明信息来源，但这并不是强求的。信息来源通常要包括标题、作者、出版社以及ISBN。比如“Spring Data by Mark Pollack, Oliver Gierke, Thomas Risberg, Jon Brisbin, and Michael Hunger (O'Reilly). Copyright 2013 Mark Pollack, Oliver Gierke, Thomas Risberg, Jonathan L. Brisbin, and Michael Hunger, 978-1-449-32395-0”。

如果你觉得对示例代码的使用超出了合理引用或上面给出的许可范围，可以通过 permissions@oreilly.com 联系我们。

本书代码示例已发布在GitHub上 (<https://github.com/SpringSource/spring-data-book>)。

Safari® 在线书店

Safari Books Online Safari在线书店 (www.safaribooksonline.com) 是应需而变的数字图书馆，它能够以书籍或视频的形式提供专家级的内容 (<http://www.safaribooksonline.com/content>)，这些内容来自于世界范围内领先的技术和商业领域的作者。

技术专家、软件开发人员、Web设计师以及商业和创意人士都选择Safari在线书店作为研究、解决问题、学习以及认证培训的主要信息来源。

Safari在线书店针对组织 (<http://www.safaribooksonline.com/organizations-teams>)、政府部门 (<http://www.safaribooksonline.com/government>) 以及个人 (<http://www.safaribooksonline.com/individuals>) 提供了多种产品 (<http://www.safaribooksonline.com/subscriptions>) 以及定价策略。订阅者可以通过一个完全可检索的数据库访问上千种图书、培训视频以及尚未出版的书稿，这些内容来自像O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology这样的几十家出版社 (<http://www.safaribooksonline.com/publishers>)。想了解Safari在线书店的更多信息，请访问在线站点 (<http://www.safaribooksonline.com/>)。

如何联系我们

请把对本书的评论和问题发给出版社。

美国：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）

奥莱利技术咨询（北京）有限公司

O'Reilly的每一本书都有专属网页，可以在那找到关于本书的相关信息，包括勘误、样例以及其他信息。可以通过以下地址访问该页面：

<http://oreil.ly/spring-data-1e>。

对于本书的评论和技术性的问题，请发送电子邮件到：

bookquestions@oreilly.com

关于本书的更多信息、会议、资料中心和网站，请访问以下网站：

<http://www.oreilly.com>

想了解我们的图书、培训课程、会议以及新闻等信息，可以访问我们的主页：

<http://www.oreilly.com>

<http://www.oreilly.com.cn>

致谢

我们要感谢Rod Johnson和Emil Eifrem，正是他们开启了Spring Data项目。

非常感谢David Turanski的贡献，他帮助我们编写了GemFire这一章。感谢Richard McDougall对大数据的分析，我们在本书的前言中用到了它，感谢Costin Leau帮忙编写Hadoop示例应用程序。

我们还要感谢O'Reilly Media，尤其是带领这个项目的Meghan Blanchette，制作编辑Kristen Borg以及文字编辑Rachel Monaghan。感谢Greg Turnquist、Joris Kuipers、Johannes Hiemer、Joachim Arrasz、Stephan Hochdörfer、Mark Spritzler、Jim Webber、Lasse Westh-Nielsen以及所有为本书提供技术审阅的人。感谢这个项目的社区，你们提交的反馈和缺陷促使我们不断进步。最后，还要特别感谢朋友和家人的宽容、理解和支持。

关于作者

Mark Pollack博士曾在布鲁克黑文国家实验室研究高能物理学方面的大数据解决方案，随后转移到金融服务领域担任前端交易系统的技术领导和架构师。他长期关注软件开发流程的最佳实践和改善，2003年就参与了核心Spring（Java）的开发，并在2004年成立了Microsoft对应的项目也就是Spring.NET。Mark现在领导着Spring Data项目，在使用大数据和NoSQL数据库这些新技术时，这个项目能够简化应用的开发。

Oliver Gierke是SpringSource的工程师，这是VMware¹的一个子部门，目前他担任Spring Data JPA、MongoDB以及核心模块的领导者。他参与企业级应用和开源项目的开发已经超过了6年，其工作的关注点在软件架构、Spring以及持久化技术方面。他经常在德国以及一些国际会议上进行演讲，写过很多技术文章。

Thomas Risberg目前是Spring Data团队的成员，关注于MongoDB和JDBC扩展项目。他也是Spring框架项目的提交者，主要的贡献在于对JDBC框架的增强。Thomas在VMware的Cloud Foundry团队，为Cloud Foundry所支持的各种框架和语言开发集成方案。他是《*Professional Java Development with the Spring Framework*》²一书的合著者，这本书出版于2005年，作者还包括Rod Johnson、Juergen Hoeller、Alef Arendsen以及Colin Sampaleanu。

Jon Brisbin是SpringSource Spring Data团队的成员，致力于为开发人员提供有用的库，从而简化下一代数据形式的管理。他曾帮忙将Grails GORM的对象映射器转移到基于Java的应用程序之中，并为集成Riak数据存储与RabbitMQ消息代理提供组件。除此之外，他还针对事件应用模型撰写博客并发表演讲，他勤奋工作的领域还包括为前沿的非阻塞模式和传统的基于JVM的应用搭建桥梁。

Michael Hunger长期热衷于软件开发。他尤其关注开发软件的人、软件技艺、编程语言以及代码效率提升。最近两年间，他与Neo Technology协作开发Neo4j图数据库。作为Spring Data Neo4j的领导者，他为对象-图映射开发出了便利且完整的解决方案。他还参与Neo4j云托管。作为一名开发人员，Michael喜欢使用各种编程语言、每天学习新的东西、参与有趣且有前景的开源项目，并且参与编写了多本与编程相关的图书。Michael还是InfoQ活跃的编辑和采访者。

¹ 译者注：目前 SpringSource 和 Cloud Foundry 均已经被转移到了 Pivotal 之中。

² 译者注：本书中文名为《Java 框架高级编程》。

目录

第一部分 背景知识

第 1 章	Spring Data 项目	3
1.1	为 Spring 开发人员提供的 NoSQL 数据访问功能	3
1.2	主题概述	4
1.3	领域	5
1.4	示例代码	6
1.4.1	将源码导入到 IDE	6
第 2 章	Repository: 便利的数据访问层	11
2.1	快速入门	11
2.2	定义查询方法	14
2.2.1	查找查询的策略	14
2.2.2	衍生查询	14
2.2.3	分页和排序	15
2.3	定义 Repository	16
2.3.1	调整 Repository 接口	17
2.3.2	手动实现 Repository 方法	18
2.4	IDE 集成	20
2.4.1	Intelli IDEA	21
第 3 章	使用 Querydsl 实现类型安全的查询	23
3.1	Querydsl 简介	23
3.2	生成查询元模型	26
3.2.1	构建系统集成	26
3.2.2	所支持的注解处理器	27
3.2.3	使用 Querydsl 对存储进行查询	28
3.3	集成 Spring Data Repository	28
3.3.1	执行断言	29
3.3.2	手动实现 Repository	29

第二部分 关系型数据库

第 4 章 JPA Repository	33
4.1 示例工程	33
4.2 传统方式	38
4.3 启动示例代码	39
4.4 使用 Spring Data Repository.....	42
4.4.1 事务性	45
4.4.2 Repository 与 Querydsl 集成.....	46
第 5 章 借助 Querydsl SQL 实现类型安全的 JDBC 编程	48
5.1 示例工程与搭建过程	48
5.1.1 HyperSQL 数据库.....	49
5.1.2 Querydsl 的 SQL 模块.....	50
5.1.3 构建系统集成	53
5.1.4 数据库模式	54
5.1.5 示例工程的领域实现	54
5.2 QueryDslJdbcTemplate.....	57
5.3 执行查询	58
5.3.1 Repository 实现起步.....	59
5.3.2 查询单个对象	60
5.3.3 OneToManyResultSetExtractor 抽象类.....	61
5.3.4 CustomerListExtractor 实现.....	63
5.3.5 RowMapper 的实现类	64
5.3.6 查询对象列表	65
5.4 插入、更新和删除操作	65
5.4.1 使用 SQLInsertClause 进行插入操作.....	65
5.4.2 使用 SQLUpdateClause 进行更新操作	66
5.4.3 使用 SQLDeleteClause 进行删除行操作	67

第三部分 NoSQL

第 6 章 MongoDB: 文档存储	71
6.1 MongoDB 简介	71
6.1.1 设置 MongoDB	72
6.1.2 使用 MongoDB Shell.....	73