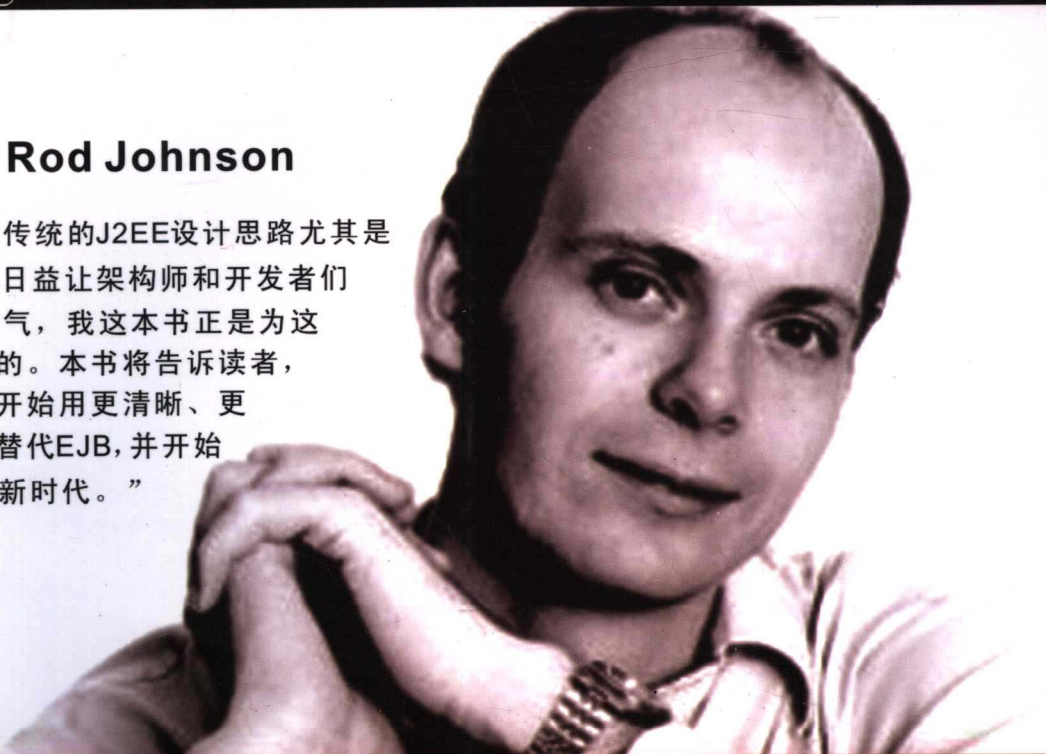


## Rod Johnson

“传统的J2EE设计思路尤其是EJB日益让架构师和开发者们灰心丧气，我这本书正是为这些人而写的。本书将告诉读者，如何从现在开始用更清晰、更高效的方案去替代EJB，并开始迈向web 应用的新时代。”



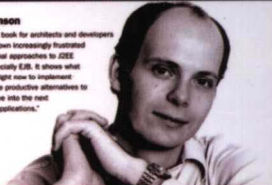
expert one-on-one™

# J2EE™ Development without EJB™

中文版

### Rod Johnson

"I wrote this book for architects and developers who have grown increasingly frustrated with traditional approaches to J2EE design, especially EJB. It shows what you can do right now to implement cleaner, more productive alternatives to EJB and move into the next era of web applications."



expert one-on-one

**J2EE Development  
without EJB**

rod.johnson@jaguar.co.uk



ORDER, INQUIRY, AND SALES TECHNICAL SUPPORT AT WWW.WILEY.COM

Rod Johnson 著  
Juergen Hoeller 译  
JavaEye



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn

expert one-on-one

**J2EE Development without EJB 中文版**

expert one-on-one

**J2EE Development without EJB**

[美] Rod Johnson 著  
Juergen Hoeller

JavaEye 译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

这本书拥有一大堆“看点”。譬如说，它的作者 Rod Johnson 拥有 10 年编写 Java 程序的经验，目前是 Servlet 和 JDO 2.0 两个 JSR 专家组的成员；再譬如说，书中着力介绍的 Spring、Hibernate、WebWork 等都是时下流行的开源框架，IoC、AOP 之类都是时下流行的概念词汇。而最大的看点就赫然摆在这本书的封面上：“without EJB”。我们曾经在无数的书籍和文章中看到，EJB 是 J2EE 的核心技术之一；而 Rod Johnson 的这本书竟然宣称，绝大多数的 J2EE 应用根本不需要 EJB。这种近乎挑衅的姿态令任何一个负责的 J2EE 架构师很难不萌生一探究竟的念头——不论你是打算赞同他还是打算驳斥他。

但所有这些尽皆不是本书最大的价值所在。选择一种架构、一种技术的依据是什么？Rod Johnson 认为，应该是基于实践的证据、来自历史项目或亲自试验的经验，而不是任何形式的偶像崇拜或者门户之见。书中谈到了企业应用方方面面的问题和解决办法，而这些方案无一不是这种“循证方法”的产物。除了把这些方案交给读者，Rod Johnson 通过这本书希望传达的、更为重要的信息正是“循证”的工作方式——那原本就应该是程序员的工作方式。

0-7645-5831-5 expert one-on-one J2EE Development without EJB by Rod Johnson and Juergen Hoeller

Original English language edition Copyright©2004 by Rod Johnson and Juergen Hoeller. All rights reserved in cluding the right of reproduction in whole or in part in any form. This Translation published by arrangement with John Wiley and Sons, Inc.

**Trademarks:** Wiley, the Wiley Publishing logo, Wrox, the Wrox logo, Programmer to Programmer, Expert One-on-One, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates. J2EE and EJB are trademarks of Sun Microsystems, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

**The Wrox Expert Series** trade dress is a trademark of John Wiley and Sons, Inc. in the United States and/or other countries. Used by permission

本书简体中文专有翻译出版权由 John Wiley and Sons Inc. 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。

版权贸易合同登记号：图字：01-2004-5209

## 图书在版编目（CIP）数据

expert one-on-one J2EE Development without EJB 中文版 / (美) 詹森 (Johnson,R.), (美) 赫鲁 (Hoeller,J.) 著; JavaEye 译. —北京: 电子工业出版社, 2005.9  
书名原文: expert one-on-one J2EE Development without EJB  
ISBN 7-121-01684-2

I.e... II.①詹...②赫...③J... III.JAVA 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2005) 第 094357 号

责任编辑: 周 筠 刘铁锋

印 刷: 北京智力达印刷有限公司印刷

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

经 销: 各地新华书店

开 本: 787×980 1/16 印张: 36.5 字数: 800 千字

印 次: 2005 年 9 月第 1 次印刷

定 价: 59.80 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

# 译者序

几年以后，当人们重新审视这本《*Expert One-on-One J2EE Development without EJB*》时，他们会记起这本书曾经如何引领 J2EE 架构与开发的潮流。

作为译者，我可以毫不费力地帮这本书找出一大堆“看点”。譬如说，它的作者 Rod Johnson 拥有 10 年编写 Java 程序的经验——那正是 Java 诞生至今的时间，他目前是 Servlet 和 JDO 2.0 两个 JSR 专家组的成员，他的前一本书《*Expert One-on-One J2EE Design and Development*》曾经风靡一时；再譬如说，书中着力介绍的 Spring、Hibernate、WebWork 等都是时下流行的开源框架，IoC、AOP 之类都是时下流行的概念词汇。而最大的看点就赫然摆在这本书的封面上：“without EJB”。我们曾经在无数的书籍和文章中看到，EJB 是 J2EE 的核心技术之一；而 Rod Johnson 的这本书竟然宣称，绝大多数的 J2EE 应用根本不需要 EJB。这种近乎挑衅的姿态令任何一个负责的 J2EE 架构师很难不萌生一探究竟的念头——不论你是打算赞同他还是打算驳斥他。

然而在我看来，所有这些都有其价值，但尽皆不是本书最大的价值所在。任何一个从事 J2EE 应用开发的程序员或多或少都曾有过这样的感觉：这个世界充斥着形形色色的概念和“大词”，如同一个幽深广袤的魔法森林般令人晕头转向，不知该追随这位导师还是该信奉那个门派。这时，Rod Johnson 发出振聋发聩的一呼：尔等不必向泥胎偶像顶礼膜拜，圣灵正在尔等自身——这就是他在书中一直倡导的“循证架构”（evidence-based architecture）。选择一种架构、一种技术的依据是什么？Rod Johnson 认为，应该是基于实践的证据、来自历史项目或亲自试验的经验，而不是任何形式的偶像崇拜或者门户之见。书中谈到了企业应用方方面面的问题和解决办法，而这些方案无一不是这种“循证方法”的产物。除了把这些方案交给读者，Rod Johnson 通过这本书希望传达的、更为重要的信息正是“循证”的工作方式——那原本就应该是程序员的工作方式。

所以，在这篇你翻开封面首先读到的文字里，我可以负责任地告诉你：那看上去有些骇人听闻的“without EJB”字样仅仅是 Rod Johnson 用来吸引你的一个噱头而已——EJB 诞生至今已经有 7 年之久，如果如今的 J2EE 应用仍旧不能摆脱这样一种“古老”的技术，岂不为天下笑？这里的关键是：EJB 正是 Java 世界里最大的“泥胎偶像”，围绕着它的门派之争历来激烈无比。透过“without EJB”这个标题，Rod Johnson 真正想说的是“without buzzword”——扔掉一切可笑的门户偏见，不再仅仅因为看到“EJB”这么一个貌似神秘的缩写词而奉若圭臬或是弃若敝屣，完全根据技术本身和实打实的经验来判断是否有用、如何使用一种技术，这才是 Rod Johnson 希望传递给读者的态度。

于是你可以相信，我在开头处说的那句话并非夸大其词：你手上的这本《*Expert*

*One-on-One J2EE Development without EJB*》将会引领一种影响深远的潮流，但这潮流不是 Spring 和 Hibernate，也不是 IoC 和 AOP，甚至不是“轻量级架构”，而是一切实事求是的“循证架构”的工作方式。惟有掌握这种工作方式，你才能够真正自信满满地挺起胸膛说：“我选择的架构是适合应用需求的架构。”毕竟，衡量架构师价值的标准不应该是他知道多少概念，而是他能否做出合乎需求的架构。

“可是，”有人说了，“我一直都是按照别人/网上/书上的建议来做架构的。要亲自考察各种各样的技术，还要根据项目情况比较它们的优劣，我可没这份时间。”

那么，他也应该没时间去架构。

## 关于译者

本书的翻译以一种奇特的方式进行：共有 8 人参与其中——值得一提的并不是这个人数，毕竟数人、十数人乃至数十人合译一本书在国内都早已不算新鲜。然而参与本书翻译的 8 个人，都算得上是国内 J2EE 技术圈里小有名气的人物，他们是：

- 刘天北（网名 Jacques Lebrun），北京奥捷特信息技术有限公司技术总监，资深 J2EE 架构师，《*Core J2EE Patterns*》（第二版）一书的译者。他负责翻译本书第 3 章“各种架构”、第 4 章“简单性的红利”和第 16 章“示例应用系统”。
- 李鲲（网名 dlee），独立软件咨询顾问，EJB 的老用户。他负责翻译本书第 5 章“EJB，五年间”和第 12 章“替换其它的 EJB 服务”。
- 杨戈（网名 Yanger），Communication Central Software 公司产品经理，Spring 中文论坛（<http://spring.jactiongroup.net/>）站长，国内最早接触和介绍 IoC 容器概念的先行者。他负责翻译本书第 6 章“轻量级容器与控制反转”和第 7 章“Spring 框架简介”。
- 石一楹（网名 potian），杭州天视智能系统有限公司技术总监，AOP 标准化组织“AOP 联盟”（AOP-Alliance，<http://aop-alliance.sourceforge.net>）成员。他负责翻译本书第 8 章“基于 AOP 概念的声明性中间件”和第 11 章“远程调用”。
- 范凯（网名 Robbin），独立软件咨询顾问，Java 视线网站（<http://www.javaeye.com>）站长，在国内较早使用和宣传 Hibernate 技术。他负责翻译本书第 9 章“事务管理”和第 10 章“持久化”。
- 钱安川（网名 moxie），独立 J2EE 技术咨询师，对 J2EE 表现层技术、尤其是各种 MVC 框架颇有研究。他负责翻译本书第 13 章“Web 层设计”。
- 曹晓钢，南京动能软件公司研发部经理，满江红开源网站（<http://www.redsaga.com/mambo/>）站长，资深 J2EE 架构师，《深入 Java 虚拟机》一书的译者，《深入浅出 Hibernate》一书的作者。他负责翻译本书第 14 章“单元测试与可测试性”和第 15 章“性能与可伸缩性”。
- 最后，是这篇译序的捉刀人，CSDN 网站技术主编熊节（网名 gigix）。他负责翻译了

本书最没有技术含量的三章：第 1 章“为什么要‘J2EE without EJB’”、第 2 章“目标”和第 17 章“结语”，以及前言、作者简介之类的边角文字。

这本《*Expert One-on-One J2EE Development without EJB*》涵盖了 J2EE 应用架构的方方面面，以其内容之丰、跨度之大，要翻译它已经超出了笔者的能力范畴。幸而得到其他 7 位专家鼎力相助，方能顺利完成翻译工作。读者若是能从书中有所收获，全赖 7 位专家对技术准确到位的理解和阐述。除翻译之外，笔者还负责全书的文字修润，如果读者感觉本书读来诘屈聱牙，责任全在我一人身上。

由于本书是众人集体劳动的结晶，参与翻译工作的诸人又都是通过网络彼此结识，是以深感网上技术社群的重要性。本书给译者带来的所有收入将全部捐献给身患重病的程序员王俊 (<http://news.csdn.net/news/newstopic/22/22735.shtml>)，希望他能够早日康复。

最后，祝您阅读愉快。

# 关于作者

**Rod Johnson** 是一名企业 Java 架构师，在保险、电子商务和金融行业的信息化领域有丰富的经验。他是欧洲最大的门户网站之一的 J2EE 架构师，并且以顾问的身份参与了大量的项目。

Rod 在悉尼大学（University of Sydney）获得了音乐和计算机科学的学位。在回到软件开发领域之前，他还获得了音乐学的博士学位。他有相当深厚的 C/C++ 技术背景，并且从 Java 和 J2EE 发布之初就已经开始使用它们。他积极地参与到 Java 社群过程（JCP）之中，是 JSR-154（Servlet 2.4）和 JDO 2.0 规范专家组的成员。自 2000 年以来，他参与写作了好几本 J2EE 方面的书籍，畅销书 *Expert One-on-One J2EE Design and Development*（Wrox, 2002 年）便是出自他的笔下。

Rod 在开源社群颇有声名：他是 Spring 框架开源项目（[www.springframework.org](http://www.springframework.org)）的作者之一，这个框架是从 *Expert One-on-One J2EE Design and Development* 书中的代码发展而成的。他经常在业界最高层次的大会上发表演讲。目前他住在伦敦。

可以通过这个邮件地址联系到 Rod：[expert@interface21.com](mailto:expert@interface21.com)。下面是 Rod 的话：

我首先要感谢我的妻子 Kerry，为她永不间断的爱和支持。然后我要感谢那些给了我们实际支持的人，感谢 Gary Watson、Andrew Smith 和 Jason Carreira 详尽地审阅了全部手稿，感谢 Alef Arendsen 的审阅和极有价值的性能评测，感谢 Peter den Haan 审阅了几个章节，感谢 Renaud Pawlak 严格审阅了与 AOP 相关的内容，感谢 Steve Jefferson、Thomas Risberg 和 Dmitriy Kopylenko 的审阅。

我还要感谢所有与我分享——不管是当面还是通过 e-mail——J2EE 开发经验的开发者和架构师。

和往常一样，与 Juergen 一同工作总是一件乐事。

Juergen Hoeller 是 werk3AT 公司的资深系统架构师和顾问，这家公司在奥地利提供复杂的 web 解决方案和针对 J2EE 的咨询顾问。

Juergen 在林茨大学（University of Linz）获得了计算机科学的硕士学位，专长是 Java、OO 建模和软件工程。他曾经在各种 J2EE 应用服务器上开发了数不胜数的项目，从企业应用集成到基于 web 的数据呈现无所不包。Juergen 在 J2EE web 应用、O/R 映射和事务管理方面的经验尤其深厚。

Juergen 是 Spring 框架项目的领导者之一，并且经常活跃在很多社群论坛上，自然也包括 TheServerSide 网站。Juergen 说：

首先，我要感谢我的妻子 Eva，她给了我无穷的爱和支持，并且宽容了我的忙碌。

我还要感谢 werk3AT 公司的同事，尤其是 Werner Loibl，他尊重我做的所有事，而且对 Spring 框架和本书做出了巨大的贡献。

衷心感谢 Thomas Risberg 和 Alef Arendsen 细致审阅本书并提供很有价值的反馈。感谢所有参与讨论、帮助我们厘清思路的开发者，不论他们是否站在 Spring 这一边。

不管开发 Spring 还是写作这本书，与 Rod 一道工作都是特别令人愉快的事。

## Credits

**Vice President and Executive Group Publisher**

Richard Swadley

**Vice President and Executive Publisher**

Bob Ipsen

**Vice President and Publisher**

Joseph B. Wikert

**Executive Editorial Director**

Mary Bednarek

**Executive Editor**

Robert Elliott

**Editorial Manager**

Kathryn A. Malm

**Development Editor**

Adaobi Obi Tulton

**Technical Editors**

Gary Watson

Andrew Smith

Jason Carreira

**Production Editors**

Felicia Robinson

Eric Newman

**Copy Editors**

C. M. Jones

Michael Koch

**Media Development Specialist**

Kit Malone

**Text Design & Composition**

Wiley Composition Services



# 前言

就其涵盖的范围而言，读者捧在手中的这本书已经算得上短小轻薄了，这是因为本书所讲述的主题并不像你所想象的那么复杂。

J2EE 的“正统”使得人们在很多简单的问题上投入了大量的精力。说实话，J2EE 行业的情况常常会让人感到：这些人坚信世界上根本就没有简单的问题。

很多——也许是绝大多数——J2EE 应用程序都遭遇了过度工程，并因此拥有一个毫无必要的、复杂的体系结构。过度工程会带来巨大的开销，而 J2EE 开发者们往往一厢情愿地认为：复杂的体系结构能够极大地降低未来的成本，足以抵消前期增加的成本。可惜，这样的因果关系在软件工程领域通常是痴人说梦。预先设计的复杂度越高，就意味着有越多的代码需要编写和维护，意味着系统中潜伏着越多的 bug，意味着需要越多的时间才能向用户展示系统的功能。说到底，预先设计的复杂度越高，失败的机会就越大，成本开销也越高。

J2EE 的过度工程多半与 EJB 有关。正如我在 *Expert One-on-One J2EE Design and Development* 中曾经说过的，EJB 常常被误用。这是一个严重的问题，因为 EJB 有可能带来巨大的复杂度——甚至比它所掩盖起来的复杂度还要大。而且，EJB 提供的一些服务也被过高地估计了。譬如说，大凡有经验的开发者和架构师，只要用实体 EJB 访问过关系型数据，几乎没有谁愿意重复这个痛苦的经历——至少，如果有 JDO、Hibernate 和其他的透明持久化技术可供选择的话。

从 2002 年末开始，对 EJB 的批评逐渐成了家常便饭。对于一种并不完美的技术，挑毛病总是很容易的，而找出合适的替代品就不那么容易了。大多数 J2EE 应用程序并不能从 EJB 那里得到更多的利益，本书便为这些项目的开发者指出了一条更好的新路。本书并不是纯粹的理论探讨，而是对于“如何在有限的时间和预算范围内设计、实现高性能 J2EE 应用程序”的实践指南。我们推荐的体系结构来自大量的经验，而且有众多开源解决方案能够满足其中常见的基础设施需求，我们还根据这个体系结构创建了一个极具现实意义的示例应用，这些都足以证明它的价值。

尽管 EJB 已经暴露出很多问题，但它仍然被大量采用，原因无外两种：“流行”和“恐惧”。EJB 一度非常流行，就连不懂技术的管理者们也多少听说过它的大名，关于 J2EE 体系结构的书籍也很少提到 EJB 的替代品——尽管确实有一些出类拔萃的替代品存在。而且，人们还害怕这些替代品会使情况变得更糟。譬如一种常见的论调说：如果没有 EJB 容器的帮助，开发者们就得亲手解决诸如事务管理之类复杂的问题。本书将让读者看到：这些恐惧大多是毫无根据的。在每一处潜藏着复杂问题的地方，读者都将看到：EJB 的替代品们以更为精彩的方式解决了这些问题。

对于典型的 J2EE 应用程序开发，本书展示了一种更为简单的解决之道——比 Java Pet Store 所示范的“经典”J2EE 蓝图方案要简单得多。我们的方案能够降低开发成本、缩短开发周期、提高可维护性和运行效率。

本书所介绍的架构方案是 J2EE 体系结构朝着更简单、更理性的方向迈出的第一步，它适于与敏捷方法结合使用。我们的方案采用了面向方面的程序设计（Aspect Oriented Programming, AOP）等新近发展的技术，并且从 .NET 等与 J2EE 竞争的平台借鉴了很多有益的思想。

作为本书的作者，我希望能够帮助读者搭建能够满足需求的、尽可能简单的应用程序——自然，也应该是最便宜、最具可维护性和可测试性的应用程序。

令人惊讶地，“EJB 之优劣”在 J2EE 社群中已经成了一个感情问题。在这个问题上，J2EE 社群呈现出一种彻底的两极分化：要么是除非被逼无奈、否则决不使用 EJB；要么是认为大凡怀疑 EJB 的人都是懒惰无知的异教徒，几乎没有任何中间派。

读者大概都知道，我不是 EJB 的支持者，但我也曾经用 EJB 开发过很多应用程序。我对于 EJB 的任何观点，都源于丰富的个人经验和对 EJB 规范的深入了解。在本书中，我也努力澄清自己在这个问题上的立场：我的目标是帮助读者开发出出色的应用程序，而不是对抗 EJB。读完本书之后，你应该有能力评估 EJB 在每个应用程序中的价值。如果在充分了解 EJB 及其替代品之后，你仍然坚信 EJB 能够最好地满足你的需求，就使用 EJB 吧。本书不会黑白分明地对读者说“不要使用 EJB”，但你必须清楚了解自己的选择。

## 本书为谁而作

本书的目标读者是对技术有充分把握、希望更加有效地使用技术的 J2EE 架构师和开发者。本书的内容是企业应用开发中的“why”和“how”，而不仅仅是“what”。所以，在这里你看不到 API 的列举，也看不到对 J2EE 基础服务（例如 JNDI 和 JTA）的介绍——有很多书和文章已经介绍了这些主题。

对于 web 应用的开发者来说，本书中的内容尤其有用。不过，即便不是开发 web 应用，绝大多数 J2EE 开发者仍然能够从中找到一些有价值的东西。也许读完本书以后，你会断定 EJB 正是适合你的选择，即便如此，按照本书所提出的标准，你也能够清楚地知道自己为什么使用 EJB、EJB 究竟为你提供了哪些价值。

## 本书的目标

本书的目标是要帮助读者解决实际问题。我们希望借本书向读者展示一种——与 Sun 公司的蓝图应用所展示的、基于 EJB 来管理业务逻辑的传统 J2EE 方案相比——更为简单有效的 J2EE 开发之道。

当然，我们也希望能给读者带来更多的快乐。

## 本书有哪些内容

本书涵盖了下列内容：

- EJB 存在的问题和 J2EE 架构的常识；
- 成功 J2EE 项目的核心价值；

- J2EE 应用程序（尤其是 web 应用）的有效架构；
- J2EE 架构及实现的常见错误；
- 如何针对你的应用找到最简单、最具可维护性的架构；
- 控制反转（Inversion of Control, IoC）和面向方面的程序设计（Aspect-Oriented Programming, AOP）：两种重要的技术，并且最近在 J2EE 开发中愈加显得重要。

第 6 章至第 12 章介绍了如何用轻量级的、更具可测试性的替代品来取代 EJB 提供的服务，尤其是：

- 事务管理。这是企业级应用的基础，也是使用 EJB 最常见的动机。我们将看到事务管理方面的替代方案，并分别讨论声明性和编程性的事务管理。
- J2EE 应用程序的数据访问。除了事务管理之外，这是企业级应用中的另一个核心问题——而且 EJB 对这个问题的解决非常糟糕。
- 如何使用 AOP 来解决很多企业级软件开发中的常见问题。

我们还将讨论：

- web 层设计，以及 web 层在一个设计良好的 J2EE 应用中的地位。
- 测试和测试驱动的开发方法，以及如何有效地测试 J2EE 应用。
- 性能和可伸缩性。

本书中涉及的特定技术包括：

- 使用 JDBC、Hibernate 和 JDO 访问数据。
- Struts、WebWork、Spring MVC 和 JSP 等 web 层技术。
- 使用开源产品开发健壮的应用基础设施，尽量减少需要自主开发、测试和维护的代码量。大多数 J2EE 基础设施问题都已经有着出色的开源产品可以解决，我将帮助读者合理地采用这些开源产品，从而将精力集中在自己的应用独有的问题上。

## 需要的知识

本书不是 EJB 或者 J2EE 的入门读物。

我们希望读者了解 J2EE 核心 API 和服务，例如 JNDI、JTA、数据库连接池等。

我们希望读者对于 J2SE 有充分的了解，包括反射、内部类、动态代理、JDBC、JAXP、JNDI 等。

我们希望读者对于 OO（Object-Oriented，面向对象）设计有充分的理解和经验。

本书讲述的主题是 EJB 的替代品，因此读者并不需要具备 EJB 的详细知识，但如果读者熟悉 EJB，将对阅读本书大有帮助。如果你希望快速了解 EJB，不妨看看 Ed Roman 的 *Mastering Enterprise JavaBeans, Second Edition* (Wiley, 2001)。

读者需要理解 EJB 背后的中间件基本概念，例如资源池、远程调用、事务管理等。读者还应该清楚采用 n 层架构（而非客户端-服务器架构）的基本动机。

读者需要理解 web 界面的基本概念，以及 J2EE web 应用中所使用的 MVC 架构模式。

如果你还不熟悉 AOP，不必担心，我会在第 8 章中介绍这个主题，使你能够开始实现具有 AOP 能力的应用程序。我还将在第 8 章中提供一份阅读清单，你可以从中获得更多关于 AOP 的知识。

## 推荐读物

本书是 *Expert One-on-One J2EE Design and Development* (Wrox, 2002) 的续篇。当然只读本书也未尝不可，不过我希望读者能够首先阅读它的前作，因为其中详细介绍了很多编程实践方法，这些内容在本书中往往只能一笔带过。该书的第 4 章、第 9 章、第 11 章至第 13 章与本书的关系尤为紧密，它们是撰写本书的背景。

我还强烈推荐 Martin Fowler 的 *Patterns of Enterprise Application Architecture* (Addison-Wesley, 2002)，其中探讨了企业级软件开发中的很多常见问题，并且与具体的实现技术（例如 J2EE）保持了足够健康的距离。Martin Fowler 是我最欣赏的技术作家之一，他的作品总是值得一读的。单是 Fowler 的分布式对象第一法则（“不要分布你的对象”），就已经值回书价了。此外，该书还引入了 **POJO** (Plain Old Java Object, 普通 Java 对象) 这个术语，在这个“大词”横行的年代，Fowler 用这个词让普通 Java 对象赶上了潮流。在本书中，我也将使用这个术语。

我一直坚信：J2EE 应用首先应该是一个 OO 应用。因此我强烈推荐 OO 领域的经典著作：*Design Patterns: Elements of Reusable Object-Oriented Software* (Gamma, Helm, Johnson 和 Vlissides, Addison-Wesley, 1995)。即便对于 J2EE 应用程序，这本书中列举的 23 个模式仍然是最有价值的——比那些针对实现技术的“J2EE 模式”要有价值得多。

*Core J2EE Patterns* (Alur, Crupi 和 Malks, 2003) 是一部重要的著作，这很大程度上是因为它定义了 J2EE 模式的标准名称。在本书中，我将引用其中的几个模式，例如 **Service Locator**、**Business Delegate**、**Front Controller** 等，如果已经了解这些模式，对于阅读本书将不无裨益。*Core J2EE Patterns* 更多地展现了传统的 J2EE 架构思路（不过第二版已经有了相当可观的改进），但无论如何，这是一个非常有用的资源。

我建议你每天关注最新的 J2EE 话题。我最中意的一些 J2EE 网站有：

- ❑ **TheServerSide**。最重要的 J2EE 门户，如果希望随时了解 J2EE 的发展，这里是最好的地方。在这里可以找到关于很多 J2EE 问题的讨论，以及大量有用的资源，例如技术文章、书籍章节预览、产品预览等。
- ❑ **Artima.com** ([www.artima.com](http://www.artima.com))。面向 Java/J2EE 的软件站点，由 **Bill Venners** 管理。
- ❑ **Core J2EE Patterns** 网站 ([www.corej2eepatterns.com/index.htm](http://www.corej2eepatterns.com/index.htm))。

- 各个 blog。Java blogger 们构成了一个小圈子，彼此探讨着一些非常重要的思想。如果你希望开始探索这个重要的信息渠道，[www.javablogs.com](http://www.javablogs.com) 会是一个不错的起点，这里著名的 blogger 包括 Rickard Oberg、Ted Neward、Gavin King、Bob Lee (“Crazy Bob”) 和 Jon Tirsen 等人。

我建议你每天关注中间件——而不仅仅是 J2EE——的最新信息。.NET 的整体架构和 J2EE 非常相似，和 J2EE 一样，.NET 的模式文献也在日益增加。这方面有用的资源包括：

- MSDN (<http://msdn.microsoft.com/>)。
- “微软.NET 企业级解决方案模式”网站 (<http://msdn.microsoft.com/architecture/patterns/Esp/default.aspx>)。

## 要使用本书，需要做些什么

如果要运行本书的示例应用和范例代码，你需要：

- 一个 J2EE web 容器或者应用服务器。对于使用单一数据库、不需要 JTA 的范例，我们使用 Tomcat 5.0.16；对于用到 JTA 的范例，我们使用 WebLogic 8.1 Express。书中所有的代码可以不加修改地在大多数别的应用服务器上运行（这是不使用 EJB 一个附带的好处！），所以你可以放心地把这些代码部署到你喜欢的应用服务器上。在示例应用的发布说明中指出了该应用在哪些应用服务器上进行过测试。
- 一个关系型数据库，以及相应的 JDBC 驱动。作为示范，我们使用的是 HSQLDB (<http://hsqldb.sourceforge.net/>)。不过，最多需要一些细小的更改，可以在任何具备事务能力的数据库上执行这些范例。
- Spring 框架，读者可以在 <http://www.springframework.org> 下载这个框架，以及使用 Spring 进行开发时需要的很多资源。本书附带的示例应用是在 Spring 1.0 最终版本的基础上编写的，不过它应该能够不加修改地在 Spring 的后续版本上执行。
- Hibernate 2.1 O/R 映射产品，读者可以在 <http://www.hibernate.org/> 下载它。
- Jakarta Ant 1.5.3，行业标准的 Java 构建工具。也许你更喜欢用 Maven（一个更复杂也更强大的构建工具）来构建你的项目，即便如此，你也应该清楚 Ant 的用法。脚本驱动的、可重复的项目构建步骤是非常重要的，具体使用 Ant 还是 Maven 倒在其次。
- 其他的第三方库，包括 Jakarta Commons Logging 和 Jakarta Commons Attributes 等。在完整的 Spring 发布版本中已经包含了必要的 jar 文件，更多细节请参阅 Spring 和示例应用的文档。

所有必需的软件都是开源或者允许开发者免费使用的。

## 示例应用

本书独创性地提供了一个完整的开源示例应用。技术作家们的时间都很宝贵，对于我们之中的大多数人来说，在工作之余再抽出时间来写作确实是件辛苦的事。正因为如此，很多书的示例应用仅仅是对正文起一点可怜的补充作用而已。我的前一部作品 *Expert One-on-One J2EE Design and Development* 也是如此，虽然其中提供了一些相当复杂的基础设施代码，但那仍然不能算是一个真正意义上的示例应用。

对于绝大多数技术作家来说，要提供一个更具现实意义的开源示例应用几乎是力所不能及的。然而，如果能够避免像大多数示例应用那样过度简化问题，那将是善莫大焉，因为在 J2EE 的世界里，恶魔总是躲在细节里。要衡量一个整体方案的价值，就得看它是否能够解决真实世界的复杂问题，而不能只看一个过度简化的例子。

本书的示例应用实现了大家耳熟能详的“Java 宠物店”（pet store）——这个例子的需求比较简单，而且广为人知，这样我们就不必再浪费宝贵的时间和篇幅来详细介绍问题领域的情况了。

我们的示例代码是以 Clinton Begin 的 JPetStore 为基础的，那本身就是一个很有价值的示例，读者可以在 <http://www.ibatis.com> 找到它。我们的示例应用对 JPetStore 进行了改进，使用了我们推荐的体系结构，引入了大量的接口，使用了控制反转的容器，并且采用基于 AOP 的声明性事务管理。我们相信，经过这些改进，我们得到的是一个比原来的 JPetStore 更简单、更灵活、更具可扩展性的应用程序。你也可以将我们的示例应用与 Sun 提供的 Java Pet Store 做个比较，看看我们的方案究竟比 J2EE 蓝图方案简单了多少，看看轻量级方案是如何取代 EJB 的地位的。

我们的示例应用展示了书中讨论的下列要点：

- ❑ 一个定义良好的业务接口层。
- ❑ 以一个控制反转容器为例介绍轻量级容器体系结构（参见本书第 3 章）。
- ❑ 使用 AOP 为 POJO 提供声明性事务管理。底层可供选择的事务管理技术包括 JTA、JDBC 等，切换事务管理机制不需要对 Java 业务代码做任何修改。
- ❑ 使用 MVC web 层（Spring MVC 或者 Struts 1.1）。在示例应用中，我们采用这两种框架分别实现了 web 层，让它们访问同一个业务接口层。
- ❑ 使用 Data Access Object<sup>1</sup> 模式解开业务逻辑与数据库结构及访问持久化数据的具体技术之间的耦合。
- ❑ 使用源码级元数据属性简化编程模型，尤其是事务管理。
- ❑ 在业务接口层的基础上提供多种协议的远程服务支持。

读者可以在 <http://www.wrox.com> 下载我们的示例应用。这是 Wrox 出版社的网站，读者需要首先通过搜索框或者标题列表进入本书的网页，然后点击“Download Code”链接即可下载与本书有关的全部源码。

由于很多书有类似的标题，通过 ISBN 编号来搜索也许是最方便的。本书的 ISBN 是 0-7645-5831-5。

下载这些代码之后，你可以用自己喜欢的解压缩工具将其解压缩。此外，你也可以到 Wrox 代码下载首页（<http://www.wrox.com/dynamic/books/download.aspx>）查看本书和其他 Wrox 图书相关的代码。

---

<sup>1</sup> 译者注：Data Access Object（数据访问对象，DAO）模式，用于封装数据访问及操作逻辑的设计模式，详见《J2EE 核心模式（第二版）》第 8 章。

## 版式约定

为了帮助读者更好地阅读本书，我们在书中使用了下列约定：

框中的文字带有非常重要、必须牢记的信息。

提示、小技巧，以及超出当前讨论范围的“闲话”用缩进的楷体字表示，就像本行字这样。

在正文中，我们使用了下列风格：

- 首次引入重要词汇时，用**黑体**表示，并在其后的括号中标注原文；
- 我们这样表示键盘输入：**Ctrl+A**；
- 我们用等宽字体表示文件名、URL 和正文中的代码，例如：`persistence.properties`；
- 对于不同的代码，我们展示的格式也有所不同：

在示例代码中，我们用灰色背景突出新引入的代码和重要的代码。对于不那么重要或者曾经在前面出现过的代码，则不用灰色背景突出。

## 勘误表

我们已经竭尽全力避免在正文或代码中留下任何错误。但是，世事无法完美，错误总会出现。如果你在我们的书中找到了错误，例如拼写错误或者代码错误，请不吝指正。将你的勘误信息发给我们，它也许就可以帮助另一位读者节省好几个小时的时间，而且也能帮助我们写出更好的作品。

请浏览 Wrox 网站，从搜索框或者标题列表找到本书的主页，在“图书详细信息”页面上点击“Book Errata”链接即可进入本书的勘误页面。在这里，你可以看到所有业已提交的勘误信息。另外，在 <http://www.wrox.com/misc-pages/booklist.shtml> 页面列举出了每本 Wrox 图书的勘误信息。

如果你不想让自己提交的错误信息出现在这个“图书勘误”页面上，也可以在 <http://www.wrox.com/contact/techsupport.shtml> 页面填写表单，将你找到的错误直接发送给我们。我们会检查你提交的信息，如果情况属实，我们也会将它张贴在本书的勘误页面上，并在本书的后续版本中更正错误。

## P2P.wrox.com

如果希望就本书内容与作者和其他读者展开讨论，请加入 Wrox 的 P2P 论坛 (<http://p2p.wrox.com>)。这是一个基于 web 的论坛系统，你可以在这里针对 Wrox 图书和相关技术与其他读者交流。这个论坛还提供了订阅功能，可以将你感兴趣的主题自动发送到你的信箱中。Wrox 的作者、编辑、技术专家和读者们都会在这里出现。

在 <http://p2p.wrox.com>，你可以找到几个不同的子论坛，不仅有与图书相关的讨论，还有与应用程序开发相关的讨论。如果要加入此论坛，请按照下列步骤做：

1. 访问 <http://p2p.wrox.com>，点击“Register”链接；
2. 阅读使用说明，点击“Agree”；
3. 填写必要的注册信息，以及你愿意提供的可选信息，点击“Submit”按钮，即可加入此论坛；
4. 稍后，你将收到一封电子邮件，遵照其中的指示即可核实你的用户名，并完成注册过程。

即便不加入 P2P 论坛，你也可以阅读其中的消息；但要张贴你自己的消息，就必须加入此论坛。

加入此论坛之后，你就可以发布消息，也可以回复别人的消息。你也可随时在 web 上阅读所有消息。如果你希望通过电子邮件订阅某个论坛的新消息，请在论坛列表上点击该论坛图标上的“Subscribe”链接。

更多关于如何使用 Wrox P2P 的信息，请参阅 P2P FAQ。在这份 FAQ 中可以找到此论坛系统的工作方式及一些关于 P2P 和 Wrox 图书的常见问题的答案。如果想阅读这份 FAQ，请在 P2P 的任何页面上点击“FAQ”链接。



# Contents

## 目 录

关于作者 .....	v
前言 .....	xvii
<b>第 1 章 为什么要“J2EE Without EJB” .....</b>	<b>1</b>
聚光灯下的 EJB .....	1
J2EE 还剩什么 .....	3
站在十字路口的 J2EE .....	4
前行的路 .....	5
主旋律 .....	5
轻量级框架和容器 .....	10
我们还应该使用 EJB 吗 .....	11
小结 .....	12
<b>第 2 章 目标 .....</b>	<b>13</b>
生产率 .....	13
问题 .....	14
传统 J2EE 方案解决生产率问题的办法 .....	15
提升生产率更好的办法 .....	20
OO .....	26
业务需求的重要性 .....	28
经验过程的重要性 .....	28
小结 .....	29
<b>第 3 章 各种架构 .....</b>	<b>31</b>
架构性构件 .....	31
业务服务层 .....	31
向外部暴露业务对象 .....	35
数据访问层, 或 EIS 层 .....	40
J2EE 架构 .....	42
两种 EJB 架构 .....	42
两种非 EJB 架构 .....	47
J2EE 架构实例 .....	54