TURING

MANNING

OSGi in Action
Creating Modular Applications in Java

OSGi 实战

[美] Richard S. Hall
Karl Pauls
Stuart McCulloch
David Savage
著

郭庆 李楠 李守超
谢莹莹 张磊
译

张晋锋 梁晓湛
审

- OSGi规范制定者亲自撰写
- 汇集Apache项目技术权威的实战经验
- 揭秘规范背后的考量，为你的项目成功带来启示

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

OSGi in Action
Creating Modular Applications in Java

OSGi 实战

Richard S. Hall
Karl Pauls
Stuart McCulloch
David Savage
著

郭庆 李楠 李守超
谢莹莹 张磊
译

张晋锋 梁晓湛
审

邮电出版社
北京

图书在版编目（CIP）数据

OSGi实战 / (美) 霍尔 (Richard, S. H.) 等著 ; 郭庆等译. -- 北京 : 人民邮电出版社, 2013.1
（图灵程序设计丛书）
书名原文: OSGi in Action:Creating Modular Applications in Java
ISBN 978-7-115-30067-6

Ⅰ. ①O… Ⅱ. ①霍… ②郭… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第281620号

内 容 提 要

为了弥补OSGi规范在应用指导方面的不足，四位活跃在OSGi开发第一线的技术专家联手打造了本书。本书面向OSGi规范的使用者，系统、全面、深入地阐述OSGi的重要特性及其使用方法。本书还介绍了某些技术的低层实现细节，引领读者畅游OSGi的世界。

本书面向OSGi规范的使用者，通过精彩的讲解和贴近实战的丰富示例，帮助读者完成“入门－进阶－提高”三级跳。

图灵程序设计丛书

OSGi实战

◆ 著 [美] Richard S. Hall Karl Pauls Stuart McCulloch David Savage
译 郭 庆 李 楠 李守超 谢莹莹 张 磊
审 张晋锋 梁晓湛
责任编辑 王军花
执行编辑 李 静

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：30.25
字数：721千字 2013年1月第1版
印数：1－3 500册 2013年1月北京第1次印刷

著作权合同登记号 图字：01-2011-2959号

ISBN 978-7-115-30067-6

定价：99.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版权声明

序

初闻Richard S. Hall是在2003年炎热的夏天。在咖啡桌上，德国电信（Deutsche Telekom）的一位同事告诉我，当地大学的一位老师对OSGi非常了解。首批开源的OSGi框架中，Oscar的作者便是他。在2003年，全方位地采用OSGi实属罕见，所以当时我很好奇。与此同时，Eclipse项目正着手研究采用一套新的模块化系统，我受邀以OSGi专家的身份参与了该项目。对于该项目来说，我觉得Richard能提供很大帮助，所以邀请他加入Equinox委员会。谁料这个颇具意外的邀请拉开了此后一个漫长的邮件交往历程的序幕，这种邮件往来一直延续至今，我希望永远都不要结束。每每遇到规范不够清晰，甚至更糟——当我们试图违背模块化原则时，Richard总会变得怒不可遏。如果我们不得不接受不完善的功能，有时我会觉得他身受煎熬。作为一名受邀的OSGi研究人员，他已然成为规范背后的核心人物，为我们的设计保驾护航，避免框架膨胀，并且一直遵照我们的原则。

Manning出版社向OSGi的核心人员发出了一封邀请函，提议编写本书，当时Richard也收到了该邮件。这封邮件引发了后续关于集体撰写书籍的一系列热烈讨论。之前写书的想法已经反复讨论过。对此我们也与Manning出版社进行了洽谈，但最后我选择退出，并鼓励其他同事继续参与下去。我为何会选择退出？作为OSGi规范的编写者，我很清楚和其他固执己见的人共同撰写一本书需要多大的工作量。我不希望每天晚上以及周末都加班工作，而且没有相应的报酬，虽然我非常喜欢和欣赏这些有才华的同事。很遗憾，我的决定影响了大家，这一工作被搁置了。

直到有一天Richard告诉我他已经重新拾写书计划，打算继续完成之前的工作，目前已经组建了一支更好的团队，有Karl Pauls、Stuart McCulloch和 David Savage。每一位作者都为开源和OSGi规范做出过巨大贡献。Karl在使用Felix方面经验丰富，通过将Felix的安全划分为一个单独的bundle，以此来证明模块化，Karl证明了Felix的框架结构也是模块化的。Stuart在Maven bundle插件、Ops4J以及Peaberry对Guice的扩展等方面做了很多工作。David在Apache的Sigil项目上干得很出色，在Paremus公司也取得了不错的业绩。最了解如何在实践中使用OSGi的人全在这支团队里了。作者将所有经验都融入了本书，这本书令人叹为观止。

这支团队承担了撰写此书的艰巨任务，在撰写过程中我一直和他们保持着密切联系——这不仅是因为我们都在OSGi联盟共事，还有一个重要的原因：在创作一本介绍OSGi的著作过程中，很可能会发现OSGi规范的缺点和不足，很明显这又会引发新一轮的激烈讨论，讨论通过Skype或者电子邮件进行。很遗憾，大多数情况下讨论都以该团队的获胜而告终。

他们请我写些文字介绍OSGi的历史，通过最大限度地压缩，最终仅仅用了4536个字，我想

这就是OSGi。这正是我想要的：对高质量的追求，不仅体现在书中细节方面，还体现在组织形式上。不同于目前市面上的其他书，本书并没有使用大量的代码清单，概括出几个步骤从而实现某个预期结果。恰恰相反，其撰写方式我很喜欢：不仅介绍使用OSGi的细节，更不厌其烦介绍这其中的缘由，可谓一部释理之书。

时下正需要这样一本书。我很清楚搞懂OSGi实非易事。尽管OSGi是在面向对象的基础上构建起来的，但是它增加了一系列新的设计原语，用以阐释传统面向对象设计的一些缺点。当应用规模变大，需要用到一系列开源项目或第三方代码时，这些缺点就会表现出来。在软件构建过程中，面向对象仍可算是一项颇具价值的技术。但是面向对象模式不适用于大规模的构建块（组件）一起协同工作同时又不产生更多耦合的情况。我们拼命地使用工厂、类加载黑客等模式降低对象间的耦合，但是当达到一定程度时，降低耦合度的工作会占用我们大部分精力。依赖注入降低了代码耦合，但是大量代码需要使用XML编写，XML语言本身的语法在编程领域不是很合适。注解为处理耦合提供了另一程度的支持，但使用注解本身又会带来一系列问题。在降低耦合方面我们的处理方式大都是在做表面文章，因为在传统Java运行时环境下，这些边界并未清晰界定。

OSGi不同于传统Java。它将应用视为对等模块的相互协作：模块会主动适应环境而不是假设环境会主动适应模块。适应环境需要该环境真实存在，然而这正是OSGi最大的创新之处：μServices。μServices是模块间相互联系的纽带，它允许模块随时间不断演化而不会影响其他模块。在最近的一次OSGi社区交流中，David Savage使用spiky（尖刻、易怒的）一词来描述模块，用以说明一组模块之间的耦合会使修改模块变得非常困难。μServices是OSGi中的一个设计原语，它很强大，甚至可以在应用运行的情况下动态完成对模块的更新或者安装操作。通过将模块间的交互具体化，减缓模块间的冲突。

μServices是一个崭新的概念，理解它需要换一个角度思考，这不同于时下盛行的Java。在很多方面，OSGi很像25年前的面向对象程序设计，它提供了一些新的设计原语，而这可能不同于时下的一些主流思路。面向对象需要注入新的活力，这包括一些新的设计原语，比如多态、继承、类和对象。OSGi使用bundle和μServices缔造了一些新的概念。我相信这些设计原语会成为继面向对象之后的下一代软件设计概念。这本书非常优秀，借助它你能够真正地理解OSGi，并了解它的所有好处。

Peter Kriens
OSGi 技术总监

致　谢

感谢Peter Kriens专业的审读意见，它们提高了本书的质量，同时感谢他为本书作序。此外，还要感谢本书手稿的早期读者，以及在撰写本书过程中通过论坛为我们提供宝贵意见和建议的读者们。

特别感谢在本书撰写的不同阶段，花费时间阅读本书手稿的同行评审人员，他们是：Cheryl Jeroza、David Kemper、Gabor Paller、Jason Lee、Massimo Perga、Joseph Ottinger、Jeroen Benckhuijsen、Ted Neward、Denis Kurilenko、Robert “Kebernet” Cooper、Ken Chien、Jason Kolter、Jeremy Flowers、Paul King、Erik van Oosten、Jeff Davis、Doug Warren、Peter Johnson、Costantino Cerbo、Dmitry Sklyut、David Dossot、Mykel Alvis、Eric Swanson、Patrick Steger、Jeff Addison、Chad Davis、Peter Pavlovich、Ramarao Kanneganti、Steve Gutz、Tijs Rademakers、John Griffin 和 Sivakumar Thyagarajan。他们的建议使本书更加完善。最后还要感谢Norman Richards在出版过程中对本书终稿进行的技术校对。

Manning出版社的员工在这一备受煎熬的过程中，一直为我们提供支持和帮助。在此要特别感谢开发编辑Cynthia Kane对我们的容忍。还要感谢Marjan Bace、Michael Stephens以及出版团队成员Tiffany Taylor、Katie Tennant 和 Gordan Salinovic。

最后，感谢Apache Felix社区贡献的代码以及这些年来的交流与讨论。

下面是个人感谢。Richard要感谢他的妻子和女儿，对因为写书而不能陪伴她们表示深深的歉意。Karl 感谢他的妻子Doreen、女儿Elisabeth 和儿子 Holger，谢谢你们的爱、支持和理解。Stuart感谢他亲爱的妻子Hayfa的鼓励，使他有动力完成本书。David感谢他了不起的家人，尤其是妻子Imogen对他完成本书的支持与鼓励。

前　言

早在2000年，我刚开始从事OSGi相关技术工作时，无法预料自己会在10年后继续从事该领域的工作。那时，OSGi瞄准了嵌入式市场，但那不是我感兴趣的领域。我想创建高动态性、模块化的应用，恰巧OSGi给了我实现上述目标的可能。当时，还没有免费的OSGi框架的实现，所以2000年12月我在柏林自由大学工作时，开始着手开发自己的OSGi开源框架实现Oscar。后来Oscar随我一起搬到格勒诺布尔，因为我去了约瑟夫·傅里叶大学工作。也就是在这里Oscar开始蓬勃发展。

随着OSGi技术的发展，Oscar于2004年加入了ObjectWeb开源联盟，后来又于2005年发展成为Apache 软件基金会的Felix项目。我很幸运地受到了OSGi联盟的邀请，直接参与制定OSGi第4版规范，OSGi R4规范于2004年发布。自那时起，我参与了OSGi规范制定的相关工作。起初我是一名学术研究者，2008年我加入Sun公司（已被Oracle收购）的Glass Fish团队，最近一直从事行业相关问题的研究。在最近10年间很多东西都发生了改变。

OSGi技术已不再局限于嵌入式市场，而是发展成为一个面向Java的成熟模块化系统。这种转变在2004年Eclipse IDE采用OSGi重构其插件系统时，帮了很大的忙。由于Spring和其他主要的应用服务器在企业应用领域对OSGi技术的采纳，进一步促成了OSGi技术的持续发展。尽管Java模块化仍在不断发展，但在未来很长一段时间里OSGi技术会扮演重要角色。现在说说本书。

几年以来我一直想写一本介绍OSGi的书，但是考虑到任务的艰巨以及没有时间，我并未将这一念头付诸实践。在2008年的夏天，我觉得是时候行动起来了，于是开始写作但很快又停滞不前了。直到Karl和Stuart以及后来David的加入，我才最终得偿所愿。我们各自有着不同的OSGi经验，将我们各自擅长的方面组合在一起，就包括了OSGi的全部内容。尽管如此，本书的写作仍旧历时两年多，期间经历过工作的变动以及几个孩子的降生。希望我们的努力能够对你有所帮助。

Richard S. Hall

关于本书

OSGi规范编写规整、内容翔实，若需了解OSGi技术细节，OSGi规范将是一个很好的选择。但在阅读该规范的过程中，你可能会发现它面向的是该规范的实现者而非使用者。本书的创作初衷就是打造一本面向用户的OSGi规范配套指南，以弥补OSGi规范在应用指导方面的不足。本书的目标并非简单整理、汇总OSGi的功能与特性，而是要系统、全面、深入地阐述OSGi的重要特性，并告诉读者如何使用OSGi。我们的主要思路是，忽略某些实现细节，介绍一些使用说明，尽可能简洁地阐述OSGi规范。

为此，在撰写本书的过程中，我们试图将讨论重点限定在应用OSGi技术所必需掌握的最常见概念、特性及原理上，但这并不意味着我们会回避所有复杂难懂的细节。开始使用OSGi，你会发现OSGi对模块化提出了更加严格的要求，这些要求很可能会打破过去很多习惯的做法。最后，你会了解一些技术原理，从而能够有效地调试、诊断在使用OSGi过程中遇到的各种问题。

按照编写思路，本书可分为三大部分：

(1) 解读OSGi核心规范；

(2) 阐述如何在实践中使用规范；

(3) 介绍一些与OSGi相关的高级主题。

本书第一部分将从用户视角，重点介绍OSGi核心规范中的基本内容。这部分将围绕OSGi的三层架构，即模块层、生命周期层和服务层，逐层展开介绍。但这不是介绍OSGi的唯一方式，通常是从实现一个简单服务的bundle开始介绍。在我们看来，这种介绍方式有一个弊端：由于一下子直接涉及了OSGi的全部三层，这就要求一开始就必须立即解释这三层的具体含义。

分层介绍的好处在于，我们把要介绍的概念依据层次进行了明确划分。例如，介绍模块化那一章（第2章）重点解释与模块层相关的概念，基本没有提到生命周期层和服务层。这种介绍方式也非常自然，因为模块层是OSGi的基础，生命周期层在它的基础上构建，服务层又构建在生命周期层的基础上。这部分还会重点分析在不使用高层架构的情况下，如何使用OSGi的低层架构，这在某些时候是非常必要的。

基于第一部分对OSGi核心规范的介绍，本书第二部分将从更加实用的角度介绍如何运用这项技术。这部分会详细介绍如何将现有JAR文件转换成bundle，如何对bundle进行测试、调试和管理。若要深入学习OSGi的使用方法，本书前两部分是不错的选择。

第三部分介绍一些与OSGi相关的高级主题，包括面向服务的组件模型、框架的启动、安全性以及分布式计算等。该部分是一块跳板，带你走进一个充满无限可能的OSGi世界。

路线图

第1章从较高层次概述OSGi，以及该技术旨在解决的问题。为避免介绍时过于抽象，本章通过“Hello, world!”示例来阐述OSGi框架的不同层，OSGi的主要内容将安排在后续章节介绍。此外，本章还介绍了Java以及其他相关技术对模块化的支持情况。

第2章分析OSGi框架的模块层。本章首先对计算领域的模块化进行了简单的介绍，进而引出OSGi中的模块化概念，即bundle。之后介绍OSGi基于元数据声明创建模块的方法，并展示通过该方法将一个简单的绘图程序模块化的过程。最后分析OSGi中的一项关键任务：如何解析bundle依赖。

第3章介绍OSGi框架中的生命周期层。本章首先介绍常规的生命周期管理，进而细述OSGi对bundle的动态生命周期管理。并创建一个简单的OSGi shell示例，介绍OSGi中与生命周期相关的API，同时对之前的绘图程序进行调整，使其具有OSGi生命周期的特征。

第4章分析OSGi框架中的服务层。本章首先介绍服务的定义，以及使用服务的目的和时机。随后通过一些简单的示例，介绍如何提供服务、使用服务，并通过迭代的方式介绍如何处理服务动态性的独特方面。最后，我们将应用动态服务来修改之前的绘图程序，并以此结束对服务层的讨论。

第5章再次讲述模块层，重点分析模块层中一些更高级、更细微的特性。本章会介绍处理bundle依赖和内容的其他方法，如利用bundle级别的依赖及bundle片段。此外，你还会了解到bundle如何处理执行环境与本地库。

第6章从实践的角度介绍如何将JAR文件转换为bundle，包括定义bundle元数据，打包bundle内容，增加生命周期支持。同时还将以一个开源项目为例，展示如何将一个应用划分为多个bundle。

第7章说明如何对bundle和基于OSGi的应用进行测试。本章分析了如在OSGi 中运行现有测试和模拟OSGi API。除了单元测试、集成测试，本章还介绍了测试管理以及辅助测试工具。

在探讨完测试后，第8章将介绍如何调试bundle。本章将介绍在命令行和Eclipse IDE下进行调试的方法。同时还将介绍如何通过配置开发环境来提高效率。最后，说明利用OSGi进行开发时遇到的常见问题及解决方法。

第9章讨论如何管理项目中的bundle，包括为包和bundle定义有意义的版本号，管理bundle的配置数据，并在此过程中描述OSGi相关服务。此外还介绍了触发bundle自动启动和初始化的方法。

第10章继续探讨与管理相关的内容，但侧重点从单个bundle转向了多个bundle，重点介绍一系列部署bundle及其依赖的方法，同时介绍如何控制bundle的启动顺序。

第11章介绍了和OSGi相关的面向组件编程，通过具体实例，分析了一种被称为*声明式服务*（Declarative Services）的标准OSGi组件框架。随后介绍如何使声明式服务与POJO协同工作，如何通过声明式服务简化动态服务的处理。

第12章继续研究OSGi组件框架的高级组件。本章将介绍Blueprint，它面向的是那些已经熟悉Spring技术的企业应用开发者。另外，本章还研究了Apache Felix iPOJO组件框架。最终我们将发现，OSGi组件框架的优势之一就是，多个组件可以通过服务实现协同工作。

第13章从bundle开发转向分析OSGi框架的启动，描述了配置、创建OSGi框架的标准方法。

最后还将展示如何利用标准API，将OSGi框架嵌入到已有的应用之中。

第14章深入研究在安全环境中操作OSGi，描述相关问题和解决办法。本章解释了OSGi如何扩展标准的Java安全架构，使其具备灵活性且易于管理。最后本章还将展示如何创建支持安全功能的OSGi框架，并创建一个安全的示例应用。

第15章是本书的最后一章，简要介绍了在OSGi中如何使用与Web相关的技术。我们讨论了一些常用Web应用技术的使用方法，例如servlet、JSP和WAR 文件，同时我们也将介绍如何发布和使用这些Web服务。

代码及排版规范

本书所有示例的配套代码均可从Manning的网站免费下载：www.manning.com/OSGiinAction。

正文中，代码体用来表示代码和JAR文件中清单文件的头标识及程序输出。除非必须通过签名来区分，否则方法的引用一般不包括签名。楷体用于表示新的术语。

代码清单中会包含很多注释，强调重要概念，有时，代码清单中还会给出数字符号，而相应解释在下文中。

作者在线

购买本书的读者可以免费访问Manning出版社的内部论坛，在那里你可以对本书进行评论，提问技术问题，并获得本书作者以及其他用户的帮助。在浏览器中输入网址www.manning.com/OSGiinAction，可以进入论坛或者进行订阅。此页面会告诉你完成注册后如何进入论坛，你可以获得何种帮助，以及论坛的行为规则等信息。

Manning出版社承诺在读者之间以及读者和作者间建立有意义的交流平台。但并不承诺本书的部分作者一定会参与其中，作者在论坛上所作的一切贡献都是自愿的（且是无偿的）。我们建议你尝试提出一些富有挑战性的问题，这样可以激发他们的兴趣！

只要本书没有绝版，就可以通过出版社的网站访问作者在线论坛或查看以前的讨论文档。

关于实战系列

实战系列通过介绍、概述以及实例展示等方式，旨在帮助读者更好地学习和记忆。认知科学研究发现，人类最容易记住的事情是那些自主探索过的事情。

尽管Manning没有认知科学家，但我们深信若要使学习效果变得更加持久，必须经过一系列过程，比如尝试、饶有兴趣地复述所学知识。只有经过积极的探索和实践，人们才能理解并记住新知识，或者掌握新知识。人类总是在实践中进行学习。实战书最大的特色是采用了实例驱动的编写模式。它鼓励读者大胆尝试，动手调试代码，探索新思路。

策划这套书还有另外一个比较现实的原因：我们的读者都很忙。他们可以通过学习本书而从事某一职业或者解决某些问题。他们需要一本深入浅出、同时又能学到所需内容的书。他们需要的是一本能够在实践中给予其帮助的书。实战系列图书就是为这类读者量身打造的。

关于封面插图

本书封面上的插图的标题是“士兵”。插图来自一本土耳其奥斯曼帝国的服饰画册，由伦敦老邦德街的William Miller于1802年1月1日出版。画册的扉页已经丢失，因此我们很难推断准确的创作时间。此书的目录同时使用英语和法语标识插图，每张图片都有创作它的两位艺术家的名字，他们一定会为自己的作品出现在两百年后的一本计算机编程类图书的封面上而倍感惊讶。

Manning出版社的一个编辑在位于曼哈顿西26街“Garage”的古董跳蚤市场买到了这本画册。卖主是住在土耳其安卡拉的一个美国人，交易时间是在那天他准备收摊的时候。这位编辑没带够买这本画册所需的现金，并且卖主礼貌地拒绝了他使用信用卡和支票的请求。卖主当天晚上要飞回安卡拉，看起来好像没什么希望了。怎么办呢？两个人最后通过握手约定的君子协议方式达成一致。卖主提议通过银行转账付款，编辑在纸上抄下了收款银行的信息，随后画册就到他手里了。不用说，第二天我们就把款付给了卖主。我们感谢这位陌生人能如此信任我们的同事。这让我们回忆起了那个很久以前的美好时代。

我们Manning人崇尚创造性和主动性，而以两个世纪以前的丰富多彩的地区生活作为图书封面的素材，使得计算机商业充满趣味性，这本画册中的这张图片，把我们带到了那时的生活中。

目 录