第一图书网, tushu007.com

<<OSGi实战>>

图书基本信息

书名：<<OSGi实战>>

13位ISBN编号：9787115300676

10位ISBN编号：7115300674

出版时间：2013-1

出版时间：人民邮电出版社

作者：Richard S. Hall,Karl Pauls,Stuart McCulloch,David Savage

译者：郭　庆,李　楠,谢莹莹,张　磊

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

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

第一图书网, tushu007.com

内容概要

OSGi是一种标准技术，用于创建高度模块化的Java应用。
利用OSGi，无需停止运行整个系统就可以安装、启动、停止、更新或卸载组件。
Oracle、BEA和IBM都在积极推动OSGi技术的采用。
《OSGi实战》作者既来自开发一线，又参与了OSGi规范的制定，这就保障了全书内容的权威性。
《OSGi实战》全书历时两年多方才完成，凝聚了作者们多年来的宝贵经验和辛勤汗水。
内容分三部分，涵盖OSGi核心规范解读、OSGi规范的实际应用及相关高级主题，系统、全面、深入地阐述OSGi的重要特性，揭示了大量鲜为人知的技术细节。
此外，本书还通过极具实用价值的实例示范了OSGi的各种应用，能够满足各个层次读者的需求。
通过大师的指引，相信读者一定可以轻松实现“ 入门-进阶-精通 ”三级跳，在OSGi世界里创造出无限可能。

作者简介

《OSGi实战》作者Richard S. Hall是Apache Felix 框架开发组的一位活跃成员，该开发组和其他组一样都是Felix的子项目。
2000年以后Richard参与了OSGi的开源工作并于2004年直接加入OSGi联盟。
Richard是Apache软件组织的一位成员，目前在Oracle的GlassFish项目组工作，主要工作是帮助解决OSGi相关的问题或者从事一些其他他所擅长做的事情。
Karl Pauls实现了Apache Felix Framework Security Provider，他也是Apache Felix框架开发组的一位活跃成员，该项目组和其他组一样都是Felix的子项目。
Karl是Apache软件组织的一名成员，曾经参与了多个Apache以及其他的开源项目。
Karl是Luminis的一位研究员。
Stuart Mcculloch主要负责Apache Felix的maven-bundle插件以及OPS4j中用来快速开发OSGi的Pax-Construct工具。
他还是Peaberry的作者，Peaberry是Guice的扩展用来实现对动态服务的注入。
Stuart是Sonatype的顾问，主要负责依赖注入和模块化。
David Savage在Paremus工作，2005年后他一直致力于OSGi应用的设计和构建，涵盖了众多领域如：构建工具、组件模型、数据持久化、桌面UI、管理方法、消息通信、配置、解析器以及RPC。
借助Sigil子项目David重点在开发工具领域进行了深入研究，为Apache Felix 项目做出了贡献。
他同时还直接参与了OSGi联盟有关OSGi规范的制定工作。

<<OSGi实战>>

书籍目录

目　录第一部分　OSGi：模块化、生命周期和服务第1章　揭开OSGi的面纱　21.1　OSGi的定义和目标　31.1.1　Java模块化的不足　31.1.2　OSGi能帮助你吗　61.2　OSGi架构概览　61.2.1　OSGi框架　71.2.2　将它们结合起来　101.3　Hello, world!　101.3.1　模块层示例　101.3.2　生命周期层示例　121.3.3　服务层示例　131.3.4　场景设置　151.4　OSGi的相关技术　161.4.1　Java EE　161.4.2　Jini　171.4.3　NetBeans　171.4.4　JMX　171.4.5　轻量级容器　181.4.6　Java业务集成　181.4.7　JSR 277　191.4.8　JSR 294　191.4.9　SCA　191.4.10　.NET　201.5　小结　20第2章　精通模块化　212.1　什么是模块化　212.2　为什么使用模块化　242.3　模块化绘图程序　242.4　bundle　272.4.1　bundle在物理模块化中扮演的角色　282.4.2　bundle在逻辑模块化中扮演的角色　292.5　使用元数据定义bundle　302.5.1　可读信息　312.5.2　bundle标识　322.5.3　代码可见性　342.5.4　类搜索顺序　422.6　完成绘图程序设计　432.6.1　提高绘图程序的模块化　442.6.2　启动新的绘图程序　462.7　OSGi依赖解析　462.7.1　自动解析依赖　472.7.2　使用约束保证一致性　512.8　回顾模块化绘图程序的好处　562.9　小结　59第3章　生命周期　603.1　生命周期管理　603.1.1　什么是生命周期管理　613.1.2　为什么需要生命周期管理　623.2　OSGi bundle的生命周期　633.2.1　将生命周期引入绘图程序　633.2.2　OSGi框架在生命周期中的作用　653.2.3　bundles激活器的清单文件条目　663.2.4　生命周期API　673.2.5　生命周期状态图　723.2.6　bundle缓存和框架重启　733.3　在bundle中使用生命周期API　743.3.1　配置bundle　753.3.2　部署bundle　763.3.3　检查框架状态　813.3.4　持久化bundle状态　823.3.5　事件监听　853.3.6　bundle自我销毁　873.4　动态扩展绘图程序　893.5　生命周期与模块化　963.5.1　解析bundle　963.5.2　刷新bundle　983.5.3　当更新操作没有完成更新　1013.6　小结　103第4章　学习服务　1044.1　什么是服务、为什么使用服务、什么时候用服务　1044.1.1　什么是服务　1044.1.2　为什么使用服务　1064.1.3　什么时候应该使用服务　1094.1.4　什么时候不应该使用服务　1104.1.5　仍然不确定　1104.2　OSGi服务实战　1114.2.1　发布服务　1124.2.2　查找并绑定服务　1144.3　处理动态性　1174.3.1　避免常见的陷阱　1184.3.2　监听服务　1214.3.3　服务跟踪　1264.4　在绘图示例中使用服务　1294.4.1　定义图形服务　1294.4.2　发布图形服务　1294.4.3　跟踪图形服务　1304.5　将服务关联到模块层和生命周期层　1314.5.1　为什么不能看到我的服务　1314.5.2　能否提供一个bundle特有的服务　1324.5.3　应该在何时释放服务　1334.5.4　何时注销服务　1334.5.5　应该将接口分开打包吗　1334.6　标准服务　1344.6.1　核心服务　1344.6.2　compendium服务　1354.7　小结　136第5章　深入分析模块化特性　1385.1　管理导出　1385.1.1　导入导出包　1395.1.2　隐式导出属性　1425.1.3　强制导出属性　1435.1.4　导出过滤　1455.1.5　复制导出　1465.2　导入解耦　1475.2.1　可选导入　1475.2.2　动态导入　1485.2.3　可选导入与动态导入的比较　1495.2.4　日志示例　1505.3　需要的bundle　1535.3.1　声明bundle依赖关系　1545.3.2　聚合分割包　1555.3.3　bundle依赖的相关问题　1585.4　将bundle划分为片段　1585.4.1　片段　1595.4.2　本地化中使用片段　1615.5　处理与环境相关的问题　1645.5.1　依赖执行环境　1655.5.2　构建本地库　1665.6　小结　168第二部分　OSGi实践第6章　走近bundle　1706.1　将JAR转换成bundle　1706.1.1　选取bundle标识　1716.1.2　导出包　1736.1.3　发现需要导入的包　1776.1.4　嵌入与导入　1816.1.5　增加对生命周期的支持　1816.1.6　JAR文件转换为bundle的简要说明　1836.2　分割一个应用到多个bundle　1846.2.1　创建一个大型bundle　1846.2.2　将代码拆分到多个bundle中　1936.2.3　降低模块耦合　1976.2.4　是否要转换成bundle　2016.3　小结　204第7章　测试应用程序　2067.1　迁移测试到OSGi　2067.1.1　容器内测试　2077.1.2　创建测试bundle　2087.1.3　覆盖所有基础　2107.2　模拟OSGi　2127.2.1　测试期望的行为　2137.2.2　模拟实战　2147.2.3　模拟意外情景　2157.2.4　处理多线程测试　2167.2.5　暴露竞态条件　2187.3　OSGi高级测试　2197.3.1　OSGi测试工具　2207.3.2　在多个框架中运行测试　2217.3.3　单元测试　2257.3.4　集成测试　2267.3.5　管理测试　2287.4　小结　230第8章　调试应用　2328.1　调试bundle　2328.1.1　调试实战　2348.1.2　使用HotSwap解决问题　2398.2　解决类加载相关问题　2438.2.1　ClassNotFoundException与NoClassDefFoundError　2448.2.2　类型转换问题　2468.2.3　使用uses约束　2478.2.4　同Class.forName()划清界限　2498.2.5　线程上下文类加载器　2528.3　追踪内存泄漏　2548.4　悬挂服务　2588.4.1　查找悬挂服务　2598.4.2　防止悬

挂服务 2598.5 小结 261第9章 管理bundle 2629.1 包和bundle的版本控制 2629.1.1 有效的版本控制 2639.1.2 包的版本控制 2649.1.3 bundle的版本控制 2669.2 配置bundle 2679.2.1 配置管理服务 2689.2.2 元类型服务 2769.2.3 首选项服务 2799.3 延迟启动bundle 2819.3.1 激活策略 2819.3.2 使用激活策略 2839.4 小结 284第10章 管理应用 28510.1 部署bundle 28510.1.1 管理代理 28510.1.2 OSGi bundle仓库 28710.1.3 部署管理服务 29410.2 指定bundle激活顺序 30110.2.1 介绍启动级别服务 30210.2.2 使用启动级别服务 30310.3 小结 306第三部分 高级主题第11章 组件模型和框架 30811.1 面向组件 30811.1.1 什么是组件 30911.1.2 为什么需要组件 31011.2 OSGi与组件 31111.2.1 OSGi面向服务的组件模型 31111.2.2 改进OSGi组件模型 31211.2.3 使用组件的绘图示例 31611.3 声明式服务 31611.3.1 构建声明式服务组件 31711.3.2 使用声明式服务提供服务 31811.3.3 利用声明式服务使用服务 31911.3.4 声明式服务组件生命周期 32411.4 小结 330第12章 高级组件框架 33112.1 Blueprint容器 33112.1.1 Blueprint架构 33212.1.2 用Blueprint提供服务 33212.1.3 通过Blueprint使用服务 33512.1.4 Blueprint组件生命周期 33912.1.5 Blueprint高级特性 34312.2 Apache Felix iPOJO 34712.2.1 构建iPOJO组件 34812.2.2 通过iPOJO提供服务 34912.2.3 通过iPOJO使用服务 35012.2.4 iPOJO组件生命周期 35512.2.5 使用iPOJO实例化组件 35812.3 混合和匹配 36212.4 小结 364第13章 启动和嵌入OSGi框架 36513.1 标准启动和嵌入 36513.1.1 框架API概览 36613.1.2 创建框架实例 36713.1.3 配置框架 36913.1.4 启动框架实例 37013.1.5 停止框架实例 37113.2 启动框架 37213.2.1 确定安装哪些bundle 37313.2.2 干净地关闭 37313.2.3 配置、创建和启动框架 37413.2.4 安装bundle 37413.2.5 启动bundle 37513.2.6 启动主bundle 37513.2.7 等待关闭 37613.3 嵌入框架 37713.3.1 内部还是外部 37813.3.2 谁在控制 38013.3.3 嵌入式框架示例 38113.4 小结 386第14章 确保应用程序的安全 38814.1 使用安全或者不使用安全 38814.2 安全：努力尝试使用 39014.3 OSGi特定的权限 39314.3.1 PackagePermission 39314.3.2 BundlePermission 39414.3.3 AdminPermission 39514.3.4 ServicePermission 39614.3.5 相对文件权限 39714.4 使用条件权限管理服务管理权限 39714.4.1 条件权限 39714.4.2 条件权限管理服务 39814.4.3 bundle 位置条件 40014.4.4 使用Conditional-Permission Admin 40014.4.5 实现一个策略文件读取器 40414.5 数字签名的bundle 40514.5.1 学习术语 40614.5.2 创建证书和签名bundle 40614.5.3 BundleSignerCondition 40814.6 本地权限 41014.7 高级权限管理 41114.7.1 自定义条件概览 41114.7.2 基于日期的条件 41214.7.3 用户输入条件 41414.8 汇总 41714.9 小结 421第15章 Web应用和Web服务 42215.1 创建Web应用 42215.1.1 使用HTTP服务规范 42415.1.2 使用Web应用规范 43215.1.3 标准WAR：Web URL处理程序 43615.2 提供和使用Web服务 43715.2.1 提供一个Web服务 43815.2.2 使用Web服务 44115.2.3 发布服务 44515.3 小结 452附录A 构建 bundle 453附录B OSGi标准服务 466

编辑推荐

《OSGi实战》由Richard S. Hall、Karl Pauls、Stuart McCulloch、David Savage联手打造，系统、全面、深入地阐述OSGi的重要特性及其使用方法。
理论与实践相结合，能指导实际开发。
《OSGi实战》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。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com