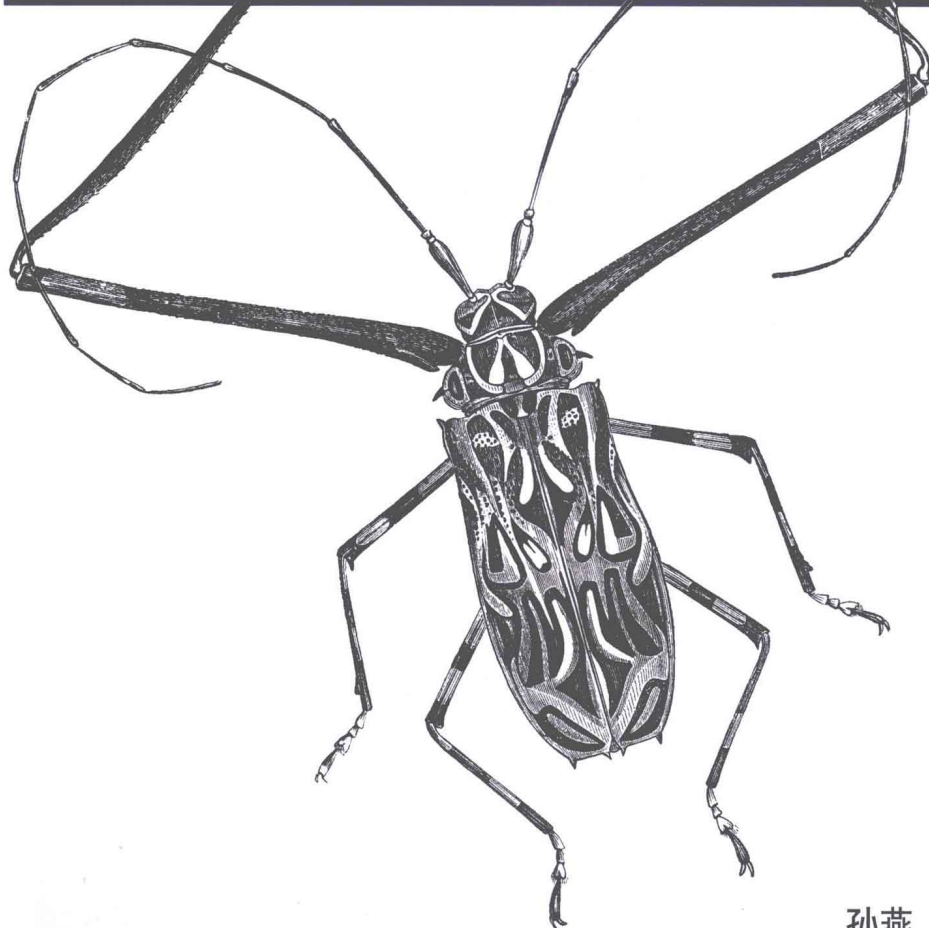


Java SOA Cookbook
实现SOA的技能、技巧与技术

Java SOA Cookbook

中文版



Eben Hewitt 著
孙燕 陈伊文 王俊华 译

O'REILLY®



清华大学出版社

Java SOA Cookbook中文版

Eben Hewitt 著

孙燕 陈伊文 王俊华 译

O'REILLY®

Beijing • Cambridge • Farnham • K ln • Sebastopol • Tokyo

O Reilly Media, Inc.授权清华大学出版社出版

清华大学出版社

北 京

Copyright ©2009 by O'Reilly Media, Inc.

Authorized Simplified Chinese translation edition, by O'Reilly Media, Inc., is published by Tsinghua University Press, 2011. Authorized translation of the original English edition, 2009 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.

本书之英文原版由O'Reilly Media, Inc.于2009出版。

本中文简体翻译版由O'Reilly Media, Inc.授权清华大学出版社于2011年出版。此翻译版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc.的许可。

版权所有，未经书面许可，本书的任何部分和全部不得以任何形式复制。

北京市版权局著作权合同登记 图字：01-2009-5157号

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Java SOA Cookbook中文版 / (美) 休依特 (Hewitt, E.) 著; 孙燕, 陈伊文, 王俊华译.

—北京: 清华大学出版社, 2011.8

书名原文: Java SOA Cookbook

ISBN 978-7-302-25297-9

I. ①J… II. ①休… ②孙… ③陈… ④王… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字 (2011) 第066444号

责任编辑: 龙啟铭

封面设计: Karen Montgomery, 张 健

责任校对: 徐俊伟

责任印制: 杨 艳

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62795954, jsjjc@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 178×233 印 张: 43.5 字 数: 994 千字

版 次: 2011 年 8 月第 1 版 印 次: 2011 年 8 月第 1 次印刷

印 数: 1~3000

定 价: 79.00 元

Java SOA Cookbook中文版

O'Reilly Media, Inc. 介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”，创建第一个商业网站（GNN），组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

前言	1
----------	---

第一部分 SOA基础

第1章 SOA入门	15
-----------------	----

1.1 概述	15
1.2 定义服务	16
1.3 定义SOA	20
1.4 识别服务候选对象	23
1.5 识别不同种类的服务	27
1.6 为服务建模	28
1.7 使服务可组合	31
1.8 支持SOA工作	33
1.9 选择试验项目	39
1.10 建立治理机制	42
本章小结	45

第2章 XML Schema和SOA数据模型	47
------------------------------	----

2.1 概述	47
2.2 为SOA设计Schema	49
2.3 创建规范的数据模型	56
2.4 使用Chameleon命名空间设计	60

2.5 对Schema进行版本控制	62
2.6 参考Schema	64
2.7 常见Schema类型.....	65
2.8 根据单个Schema验证XML文档	68
2.9 根据多个Schema验证XML文档	71
2.10 使用正则表达式限制Schema类型.....	72
2.11 使用Schema枚举	75
2.12 从Schema生成Java类	75
2.13 从Java生成Schema.....	82
2.14 在Ant中从XML Schema生成Java源文件	87
2.15 从Schema生成XML文档实例	88
2.16 定制从Schema生成Java类的方式	90
2.17 在编组和解组过程中根据Schema进行验证.....	95
2.18 在编组和解组过程中收集Schema验证事件.....	98
本章小结	100

第3章 使用XML和Java..... 101

3.1 概述	101
3.2 读取XML数据流.....	102
3.3 编写XML数据流.....	108
3.4 过滤XML流中的数据	110
3.5 从XML文档选择值	113
3.6 更新XML文档中的值	118
3.7 将Java对象转换成XML文档实例	120
3.8 将XML文档实例转换成Java对象	124
3.9 从XML文档生成Schema.....	126
3.10 不使用JAXB将XML转换成Java	128
3.11 在JAXB中自定义代码生成	130
3.12 在Linux上查找包含给定类的JAR.....	131
3.13 透明替换XML文件	133
本章小结	136

第二部分 Web服务

第4章 准备工作	139
4.1 概述	139
4.2 使用公开的Web服务进行测试	140
4.3 安装Metro	146
4.4 安装Oracle WebLogic	148
4.5 创建和部署最简单的Web服务	150
4.6 创建服务并将其部署到WebLogic	152
4.7 设置Maven 2服务和客户端项目	155
4.8 理解WSDL	160
4.9 使用NetBeans中的引用来生成Web服务客户端	163
4.10 通过Metro监控SOAP流量	166
4.11 通过TCPMon监控SOAP流量	169
第5章 基于SAAJ的Web服务	173
5.1 概述	173
5.2 创建带有限定名称的SOAP元素	177
5.3 创建完整的SOAP消息	179
5.4 将SOAP响应写出到输出流	182
5.5 基于现有的SOAP信封创建Web服务客户端	183
5.6 从SOAP消息提取内容	187
5.7 使用原始XML源和DOM创建Web服务客户端	189
5.8 添加MIME头	192
5.9 添加命名空间声明	193
5.10 指定SOAPAction	193
5.11 向元素添加属性	199
5.12 从SOAP消息去掉头	200
5.13 向SOAP请求添加头	201
5.14 访问所有SOAP头元素	209
5.15 向传出的SOAP消息添加附件	211
5.16 访问传入的附件数据	212

5.17 在没有WSDL的情况下连接到SAAJ端点	213
5.18 使用SOAP Actor	215
5.19 通过Dispatch进行异步调用	216
5.20 在客户端基于Schema验证载荷	217
5.21 提供基于SAAJ的Web服务	224
5.22 发送和接收SOAP故障	229
本章小结	232

第6章 用JAX-WS创建Web服务应用程序..... 233

6.1 概述	233
6.2 从命令行调用Web服务	236
6.3 使用JAX-WS注解名称属性	239
6.4 调用最简单的Web服务	243
6.5 创建客户端代理	244
6.6 从Servlet或EJB使用Web服务	251
6.7 从JSP使用Web服务	255
6.8 在SOAP消息中使用JAXB注解实例	257
6.9 在Maven项目中使用wsimport	258
6.10 在wsngen和wsimport中处理版本错误	260
6.11 向SOAP请求添加头	263
6.12 截取请求以执行特定于协议的工作	274
6.13 拦截请求以对载荷执行操作	281
6.14 多个处理程序调用之间共享数据	282
6.15 在请求中传递二进制数据	287
6.16 在SOAP消息中使用二进制数据	288
6.17 在客户端启用二进制优化	291
6.18 使用Metro根据Schema验证SOAP载荷	292
6.19 对JAX-WS客户端实现异步调用	294
6.20 覆盖SEI中的端点地址	301
本章小结	302

第7章 提供基于SOAP的Web服务	303
7.1 概述	303
7.2 组装用于部署的服务	304
7.3 确定服务开发模型	306
7.4 选择编码、使用和参数样式	312
7.5 基于Java服务端点实现生成WSDL和可移植结果	318
7.6 创建基本的Web服务	322
7.7 指定命名空间	326
7.8 创建Web服务操作	327
7.9 指定Web服务消息部分	328
7.10 指定操作返回值	329
7.11 定义无参数操作	330
7.12 定义带有Void返回类型的操作	331
7.13 创建使用基于自定义WSDL和自定义Schema的复杂类型的Web服务	332
7.14 指定SOAP绑定样式、使用和参数样式	347
7.15 配置标准自定义绑定	348
7.16 从服务排除Public方法	351
7.17 创建带有XML视图的服务提供类	351
7.18 实现服务器端处理程序链	359
7.19 提供有状态的服务	362
7.20 添加带有方法参数的头	365
7.21 访问服务中的传入头参数	367
7.22 为SOAP操作或WS-Addressing操作提供值	367
7.23 优化服务器上二进制内容的传输	369
7.24 获得和共享有关用户和请求的数据	370
7.25 通过Holder<T>使用头引用	371
本章小结	374
第8章 REST式Web服务	375
8.1 概述	375
8.2 使用Servlet在HTTP服务上创建POX	385
8.3 使用JAX-WS创建REST式服务	390

8.4 使用套接字创建REST式服务的客户端	391
8.5 应用：使用SSL、Atom发布和Google Finance REST API	394
8.6 创建Jersey JAX-RS实现	404
8.7 使用Eclipse和Tomcat创建Jersey项目	406
8.8 使用Jersey创建Hello World程序	408
8.9 为同一类型的多个资源创建单一路径	410
8.10 在路径模板中限制值的结构	411
8.11 访问查询参数	412
8.12 在响应中将自定义类型汇集到XML中	414
8.13 为同一资源提供不同的具象	416
8.14 创建资源	423
8.15 使用表单和URI工作	429
8.16 使用SAAJ访问REST式服务	436
8.17 在具象上设置元数据	439
8.18 删除资源	441
8.19 重定向到其他服务	442
8.20 访问HTTP头	445
8.21 使用Cookie工作	447
8.22 使用异常和响应状态码	451
8.23 使用WADL	455
8.24 使用自定义Reader和Writer与资源交互	456
本章小结	465

第三部分 业务流程

第9章 使用BPEL编排服务	469
9.1 概述	469
9.2 确定业务设计方法	473
9.3 选择业务流程建模语言	474
9.4 获取Apache ODE BPEL引擎	476
9.5 将流程部署到Apache ODE	478
9.6 理解BPEL流程基础	480

9.7 使用免费的图形设计器创建BPEL流程	485
9.8 创建调用伙伴服务的BPEL流程	486
9.9 将BPEL流程部署到OpenESB的BPEL服务引擎	509
9.10 测试已经部署的BPEL流程	512
9.11 安装Active Endpoints BPEL设计器	515
9.12 安装Active Endpoints BPEL引擎	516
9.13 在Active Endpoints设计器中创建BPEL流程	518
9.14 将流程部署到Active Endpoints服务器	520
9.15 使用网络服务伙伴	521
9.16 从BPEL流程调用伙伴服务	522
9.17 使用BPEL变量操作数据	524
9.18 使用文字	527
9.19 合并值	528
9.20 根据运行时条件选择要执行的活动	529
9.21 在一个序列中执行多个活动	531
9.22 使用逻辑划分分组活动	532

第10章 高级BPEL编排..... 535

10.1 概述	535
10.2 并行执行活动	535
10.3 同步并行执行的活动	536
10.4 什么也不做	538
10.5 在特定时间点执行活动	539
10.6 在指定延时后执行活动	540
10.7 选择性的事件处理	541
10.8 异常处理	542
10.9 显式抛出异常	545
10.10 停止流程	547
10.11 对BPEL消息数据执行XSL转换	547
10.12 校验带内消息数据	549
10.13 相关集	550
10.14 循环	554

10.15 给业务流程添加人工任务	558
10.16 从BPEL调用REST式网络服务	559
本章小结	560

第11章 SOA管理 561

11.1 概述	561
11.2 分配角色	562
11.3 创建SOA路线图	564
11.4 跟踪服务	567
11.5 确定服务的数据所有权方案	569
11.6 在SOA中处理遗留程序和异质性	572
11.7 文档化服务	573
11.8 建立服务注册表	574
11.9 打包相关服务	580
11.10 撤销服务	581
11.11 浏览UDDI注册表	582
11.12 编程查询UDDI注册表	582
11.13 理解SOA投资收益率	585

第四部分 互操作性和服务质量

第12章 Web服务的互操作性 595

12.1 概述	595
12.2 处理数组	596
12.3 抽象寻址	598
12.4 在Java服务中使用寻址	600
12.5 在客户端上显式启用寻址	602
12.6 在客户端上显式地禁用寻址	605
12.7 在WSDL的传输层上进行抽象寻址	606
12.8 对异常寻址	606
12.9 在Visual Studio中创建.NET网络服务	610
12.10 在C#中创建.NET网络服务	612

12.11 创建.NET网络服务	619
12.12 为网络服务创建Ruby客户端	625
12.13 为.NET服务创建Ruby客户端	627
12.14 遵守基本概要	629
12.15 基本概要符合度的自动化测试	632
12.16 互操作性最佳做法	633
12.17 使用模块化的WSDL.....	634
第13章 服务质量	637
13.1 概述	637
13.2 理解可靠的消息传输	639
13.3 使用可靠的消息传输配置Java Web服务.....	646
13.4 使用可靠的消息传输配置Java客户端.....	648
13.5 配置Java Web服务在WebLogic上具有可靠的消息传输	649
13.6 WebLogic可靠消息传输错误处理器	650
第14章 企业服务总线.....	652
14.1 概述	652
14.2 什么是ESB.....	653
14.3 作为模式集合的ESB.....	659
14.4 JBI	660
14.5 商业ESB	664
14.6 开源ESB	671
本章小结	678

前言

概述

有人说SOA就像雪花——没有两片是相同的。情况确实如此，因为面向服务架构的主要目的是为企业集成提供一种松散耦合的架构，而企业的内部情况却是千差万别。另外，SOA是从业务角度来设计接口的，而业务角度向来是由开发人员决定的。

因此，对于一名试图阐述实现选择和最佳实践的作者而言，这带来了一定的挑战。在过去几年中，出现了许多有关SOA的书籍，这些书籍从架构师或管理人员的视角来介绍SOA的总体概念，它们给出的是SOA的概念图景，但不是从实践前沿的角度来进行说明。

那些从架构师角度介绍SOA的书籍往往只是提供一些长的项目单，列出重要的和即将出现的WS-*规范，尽管这些书籍成功地梳理出了构建SOA的某些抽象方式，但并没有告诉程序员/架构师如何进行实际工作。也就是说，许多SOA书籍可能会告诉读者要做什么，但没有告诉读者如何去做。

例如，它们可能指出应该构建组合服务。这听起来很有说服力，也很重要，但是，接下来当你回到办公桌打开集成开发环境时，才意识到自己不知道要输入什么。有些书籍可能会更进一步，提供了基于XML的语言（如BPEL）的语法概述，但在告诉你如何实际使用它们之前就结束了。很难对这些书籍求全责备，因为SOA的本质就意味着你可以用各种工具来构建它。

编写本书的目的是要告诉读者如何实际使用SOA的一些基础构件：Web服务、编排、策略等，本书意在为开发人员填充现实世界中的空白。但要做到这一点，我必须从实际

出发。要使本书的篇幅可以控制，我必须将重点放在最重要的内容上，同时省去某些内容。

本书的重点如下。

基于SOAP的Web服务

.NET或Java EE 5中的SOAP Web服务是一个组件，其注释生成一个它所提供服务的XML描述（称为WSDL）。此描述并不特定于编写组件时所处的平台，因此，用其他语言编写的客户端可以调用SOAP Web服务。

这使得基于SOAP的服务成为SOA的一个重要组成部分，本书的大量内容用于讲述如何使用XML、SOAP Web服务以及支持它们的Java API。

与其他技术（如POX over HTTP）相比，刚开始使用SOAP可能很复杂，这导致了人们对SOAP的批评。供应商们为各种优化功能（包括可靠性、安全性、位置透明性等）实现了SOAP标准，让用户以标准、可互操作的方式获得这些功能是非常有价值的。

虽然用户可能回想起了几年前出版的许多Web服务书籍，但事情已经有了很大改变。在Java SE 6和EE 5中创建SOAP Web服务变得完全不同，本书将包括最新的资料，并且不会止步于创建Web服务，还将告诉用户如何将它们实际组合起来。

REST式（RESTful）Web服务

REST（Representational State Transfer，具象状态传输）是一种在互联网架构上构建应用的方式，它与SOAP相对（至少是在大众的心中）。在“REST式Web服务”一章中，我们将分析这一观点，然后相当全面地介绍如何以各种方式创建REST式Web服务，包括使用新的JAX-RS规范JSR 311（REST式Web服务的Java API）。本书还将介绍如何使用已经成为REST实际标准的主流API，如Atom发布协议（Atom Publishing Protocol）。

Java EE 5

虽然书中提供了通过Java之外的语言来使用Web服务的例子，但本书的重点绝对是用Java实现SOA。这样做的原因很简单：我的背景和专业领域是Java。

本书不会介绍如何用Java EE 5和Java SE 6之前的版本来编写Web服务，因为有许多书籍对此都有介绍。在最新的Java版本中，Web服务方面有了很大变化。仅在最近的一年中，Web服务在注释、新的API以及发展中的各种WS-*规范实现方面发生了真正变化。本书将只介绍最新的内容。

SOA

和O'Reilly的众多其他实战指南相比，本书有一个方面与众不同：解决方案并不总

是给出代码示例，因为SOA的有关问题并不总是代码问题。本书的重点是如何实现Java EE 5 Web服务以及使用各种相关技术，如BPEL编排和WS-*规范。涉及这些主题的章节提供了具体、真实的代码示例来说明如何完成实际工作。在这方面，本书与O'Reilly的其他实战指南没什么区别。但是，只要有可能，本书也为SOA的“人员问题”提供了解决方案，如组织和投资回报率。这些章节不涉及代码解决方案，因为它们不是代码问题。本书努力将这限定于那些有明确解决方案、建议或最佳实践的问题，以遵从一般的实战指南格式。不过，并不总是能够成功这样做，许多这类主题都受到热烈争议。此外，我需要将这类主题留给常见SOA的许多优秀书籍。如果需要一本关于这类SOA问题的真正好书，我推荐Nicolai M. Josuttis的“SOA in Practice”一书（O'Reilly, <http://www.oreilly.com/catalog/9780596529550>）。

Glassfish和WebLogic

虽然有很多提供Web服务功能的优秀应用服务器，但Glassfish是第一个支持Java SE 6的应用服务器，并且以可互操作方式完全实现了新的Java Web服务标准。这里并没有像部分供应商那样认为RPC已经过时。尽管书中的许多示例在其他容器（如Oracle的WebLogic 10gR3或Axis2）中进行了测试和展示，但Glassfish v2是这些示例的默认容器。

本人在可移植性方面做了很大努力，因此，不用太费劲，用户应该能够让这些示例在自己的引擎中工作。

BPEL编排

随着公司在寻求采用适合于开发人员、系统分析人员和架构师使用的方式来组织、改善和表示其业务流程，WS-BPEL标准是近些年来的一个重要发展。通过使用BPEL，用户可以将多个Web服务结合在一个工作流中，向外界展示一个单一编排视图，从而创建组合服务。由于用户可以用一种非常松散的耦合方式来组合多个Web服务，编排也有助于促进服务的重用。

一些开源实现（如Apache ODE）以及许多商业产品都支持BPEL，本书将逐一介绍它们。

企业服务总线（ESB, Enterprise Service Bus）

严格来说，ESB并不是SOA所必需的，但它往往是成熟SOA体系的重要组成部分。ESB是服务集成点之间的中介、路由器和间接层。在早期的EAI（Enterprise Application Integration，企业应用集成）工作中，人们很快发现需要进行很多点对点集成，这要求为每个连接节点到任何其他节点的路由创建一个特定接口。ESB降低了这种复杂性。用户可以只将服务连接到ESB，而不是将每个服务都连接到所有其他服务。通过这种方式，ESB实现了与任何计算机总线一样的功能，它负责仲裁工作并将消息路由到相应的服务。