

<<Spring攻略>>

图书基本信息

书名：<<Spring攻略>>

13位ISBN编号：9787115208903

10位ISBN编号：7115208905

出版时间：2009-8

出版时间：麦(Mak.G.)、丁雪丰、荣浩、杨春花 人民邮电出版社 (2009-08出版)

作者：麦

页数：603

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>

## &lt;&lt;Spring攻略&gt;&gt;

## 前言

几年前，机缘巧合之下，我被一个朋友拉去参加技术活动，在那次活动上我第一次接触到了Spring。说实话，当时怎么都没有想到几年间Spring会有如此迅速的发展，更想不到自己会与Spring如此有缘。从Expert One-on-One J2EE Design and Development到Expert One-on-One J2EE Development without EJB，Spring之父Rod Johnson一次又一次让大家眼前一亮。

而Spring更是被广大Java开发者所接受，可以毫不夸张地说，提到Java企业级应用的开发，第一时间蹦出的词汇中肯定有Spring。

正是因为Spring如此大红大紫，越来越多的人希望了解它、学习它。

每当有人让我推荐学习资料时，我都会让他去网上下载官方文档，这足以满足大多数人的需要。

但如果你比较喜欢飘着墨香的纸质书，而又恰好捧起本书，相信它一样也不会让你失望的。

书中以“问题描述 / 解决方案 / 实现方法”的形式介绍了Spring中的大多数内容和SpringPortfolio中的一些项目，读者既可以跟着作者的思路慢慢学习，也可以“对症下药”，立刻解决眼前的问题。

要是你对Spring已经有了一定的了解，那么本书也能作为一本不错的参考手册，偶尔翻翻也会有意想不到的收获。

正如Spring背后有着一个优秀的团队一样，本书同样也有一个优秀的翻译团队，大家相互帮助、相互鼓励，唯一的目标就是为读者带来最好的作品。

在全书翻译期间，荣浩负责第1~5章，俞黎敏负责第6、7、16、17、18章，杨春花负责第8~12章，而我则负责第13、14、15、19章以及全书的审稿、统稿。

最后，请允许我在此感谢那些在本书的翻译过程中给予我们帮助的朋友和图灵的编辑们，正是有了你们，本书才能最终呈现在读者面前。

亲爱的读者朋友，如果你发现书中存在错误或者有什么意见，请毫不犹豫地告诉我们，我们不胜感激。

。

## <<Spring攻略>>

### 内容概要

Spring着眼于Java/JavaEE应用开发的诸多方面，并针对它们提供了简单的解决方案。

《Spring攻略》涵盖了Spring 2.5从基础概念到高级应用的所有主题，介绍了几种常见的Spring项目，重点关注Spring 2.5为构建企业级Java应用程序所增加的最新特性。

《Spring攻略》的所有主题都是围绕完整真实的项目代码示例展开的，读者可以按照这些示例一步一步地实践，从而迅速掌握使用Spring Framework进行Java/Java EE开发。

《Spring攻略》适合具有丰富开发经验以及从零起步的Java开发人员阅读。

## <<Spring攻略>>

### 作者简介

作者：(美国)麦(Mak.G.) 译者：丁雪丰 荣浩 杨春花 等Gary Mak，澳门人，资深系统架构和应用开发专家，精通在Spring、Hibernate、JPA、Struts、JSF和Tapestry框架上构建企业级应用程序，主要研究方向包括面向对象技术、面向切面技术、设计模式和软件复用技术。  
他还撰写了一系列的Spring和Hibernate教程，这些教程在Java社区里广受关注和欢迎。

## 书籍目录

第一部分 核心概念第1章 控制反转和容器 21.1 使用容器管理组件 21.1.1 问题描述 21.1.2 解决方案 21.1.3 实现方法 31.2 使用服务定位器降低查找组件的复杂性 71.2.1 问题描述 71.2.2 解决方案 71.2.3 实现方法 71.3 应用控制反转和依赖注入 81.3.1 问题描述 81.3.2 解决方案 81.3.3 实现方法 91.4 理解不同类型的依赖注入 101.4.1 问题描述 101.4.2 解决方案 101.4.3 实现方法 111.5 使用配置文件配置容器 131.5.1 问题描述 131.5.2 解决方案 141.5.3 实现方法 141.6 小结 16第2章 Spring简介 172.1 Spring Framework 172.1.1 Spring的模块介绍 172.1.2 Spring的发布版本 192.1.3 Spring的项目 202.2 安装Spring Framework 212.2.1 问题描述 212.2.2 解决方案 212.2.3 实现方法 212.3 建立Spring项目 222.3.1 问题描述 222.3.2 解决方案 222.3.3 实现方法 232.4 安装Spring IDE 242.4.1 问题描述 242.4.2 解决方案 242.4.3 实现方法 242.5 使用Spring IDE的Bean-supporting特性 262.5.1 问题描述 262.5.2 解决方案 262.5.3 实现方法 262.6 小结 31第3章 Spring中的Bean配置 323.1 在Spring IoC容器里配置Bean 323.1.1 问题描述 323.1.2 解决方案 323.1.3 实现方法 323.2 实例化Spring IoC容器 363.2.1 问题描述 363.2.2 解决方案 363.2.3 实现方法 363.3 解决构造器歧义 383.3.1 问题描述 383.3.2 解决方案 383.3.3 实现方法 383.4 指定Bean引用 413.4.1 问题描述 413.4.2 解决方案 413.4.3 实现方法 413.5 通过依赖检查来检查属性 443.5.1 问题描述 443.5.2 解决方案 453.5.3 实现方法 453.6 通过@Required注解检查属性 463.6.1 问题描述 463.6.2 解决方案 463.6.3 实现方法 473.7 XML配置里的Bean自动装配 483.7.1 问题描述 483.7.2 解决方案 493.7.3 实现方法 493.8 使用@Autowired和@Resource自动装配Bean 523.8.1 问题描述 523.8.2 解决方案 523.8.3 实现方法 523.9 继承Bean配置 583.9.1 问题描述 583.9.2 解决方案 583.9.3 实现方法 583.10 为Bean属性定义集合 613.10.1 问题描述 613.10.2 解决方案 613.10.3 实现方法 613.11 为集合元素指定数据类型 673.11.1 问题描述 673.11.2 解决方案 673.11.3 实现方法 673.12 使用工厂Bean和Utility Schema定义集合 693.12.1 问题描述 693.12.2 解决方案 693.12.3 实现方法 693.13 在classpath中扫描组件 713.13.1 问题描述 713.13.2 解决方案 713.13.3 实现方法 723.14 小结 76第4章 高级Spring IoC容器 774.1 通过调用构造器创建Bean 774.1.1 问题描述 774.1.2 解决方案 774.1.3 实现方法 774.2 通过调用静态工厂方法创建Bean 804.2.1 问题描述 804.2.2 解决方案 804.2.3 实现方法 804.3 通过调用实例工厂方法创建Bean 814.3.1 问题描述 814.3.2 解决方案 814.3.3 实现方法 814.4 使用Spring的工厂Bean创建Bean 834.4.1 问题描述 834.4.2 解决方案 834.4.3 实现方法 834.5 根据静态字段声明Bean 854.5.1 问题描述 854.5.2 解决方案 854.5.3 实现方法 854.6 根据对象属性声明Bean 864.6.1 问题描述 864.6.2 解决方案 864.6.3 实现方法 864.7 设置Bean的作用域 884.7.1 问题描述 884.7.2 解决方案 884.7.3 实现方法 884.8 定制Bean的初始化和销毁过程 904.8.1 问题描述 904.8.2 解决方案 904.8.3 实现方法 914.9 使Bean感知容器 954.9.1 问题描述 954.9.2 解决方案 954.9.3 实现方法 964.10 创建Bean后置处理器 964.10.1 问题描述 964.10.2 解决方案 964.10.3 实现方法 974.11 外部化Bean配置 1004.11.1 问题描述 1004.11.2 解决方案 1004.11.3 实现方法 1004.12 解析文本信息 1014.12.1 问题描述 1014.12.2 解决方案 1014.12.3 实现方法 1014.13 使用应用程序事件进行通信 1034.13.1 问题描述 1034.13.2 解决方案 1034.13.3 实现方法 1044.14 在Spring里注册属性编辑器 1064.14.1 问题描述 1064.14.2 解决方案 1064.14.3 实现方法 1064.15 创建定制的属性编辑器 1084.15.1 问题描述 1084.15.2 解决方案 1094.15.3 实现方法 1094.16 加载外部资源 1104.16.1 问题描述 1104.16.2 解决方案 1104.16.3 实现方法 1114.17 小结 113第5章 动态代理和经典的Spring AOP 1145.1 非模块化的横切关注点所带来的问题 1155.1.1 追踪方法 1165.1.2 验证参数 1195.1.3 找出问题 1205.2 使用动态代理模块化横切关注点 1225.2.1 问题描述 1225.2.2 解决方案 1225.2.3 实现方法 1225.3 使用经典的Spring通知来模块化横切关注点 1275.3.1 问题描述 1275.3.2 解决方案 1275.3.3 实现方法 1285.4 使用经典的Spring切入点匹配方法 1355.4.1 问题描述 1355.4.2 解决方案 1355.4.3 实现方法 1365.5 为Bean自动创建代理 1385.5.1 问题描述 1385.5.2 解决方案 1385.5.3 实现方法 1385.6 小结 140第6章 Spring 2.x AOP和AspectJ支持 1416.1 在Spring中启用AspectJ注解支持 1416.1.1 问题描述 1416.1.2 解决方案 1426.1.3 实现方法 1426.2 用AspectJ注解声明切面 1446.2.1 问题描述 1446.2.2 解决方案 1446.2.3 实现方法 1446.3 访问连接点信息 1496.3.1 问题描述 1496.3.2 解决方案 1496.3.3 实现方法 1496.4 指定切面优先级 1506.4.1 问题描述 1506.4.2 解决方案 1506.4.3 实现方法 1506.5 重用切入点定义 1526.5.1 问题描述 1526.5.2 解决方案 1526.5.3 实现方法 1526.6 编写AspectJ切入点表达式 1546.6.1 问题描述 1546.6.2 解决方案 1546.6.3 实现方法 1546.7 为Bean引入行为 1586.7.1 问题描述 1586.7.2 解决方案 1586.7.3 实现方法 1586.8 为Bean引入状态 1616.8.1 问题描述 1616.8.2

## &lt;&lt;Spring攻略&gt;&gt;

解决方案 1616.8.3 实现方法 1616.9 用基于XML的配置声明切面 1636.9.1 问题描述 1636.9.2 解决方案  
1636.9.3 实现方法 1636.10 加载时在Spring中织入AspectJ切面 1656.10.1 问题描述 1656.10.2 解决方案  
1666.10.3 实现方法 1666.11 在Spring中配置AspectJ切面 1706.11.1 问题描述 1706.11.2 解决方案 1706.11.3 实  
现方法 1706.12 将Spring Bean注入到领域对象中 1716.12.1 问题描述 1716.12.2 解决方案 1716.12.3 实现方  
法 1716.13 小结 174第二部分 基础主题第7章 Spring对JDBC的支持 1787.1 直接使用JDBC的问题 1787.1.1  
建立应用程序数据库 1797.1.2 理解DAO设计模式 1807.1.3 用JDBC实现DAO 1807.1.4 在Spring中配置数据  
源 1827.1.5 运行DAO 1837.2 用JDBC模板更新数据库 1847.2.1 问题描述 1847.2.2 解决方案 1847.2.3 实现方  
法 1847.3 用JDBC模板查询数据库 1887.3.1 问题描述 1887.3.2 解决方案 1897.3.3 实现方法 1897.4 简  
化JDBC模板的创建 1937.4.1 问题描述 1937.4.2 解决方案 1937.4.3 实现方法 1947.5 在Java 1.5中使用简单  
的JDBC模板 1967.5.1 问题描述 1967.5.2 解决方案 1967.5.3 实现方法 1967.6 在JDBC模板中使用具名参数  
1997.6.1 问题描述 1997.6.2 解决方案 1997.6.3 实现方法 1997.7 将JDBC操作建模为细粒度的对象 2017.7.1  
问题描述 2017.7.2 解决方案 2017.7.3 实现方法 2017.8 在Spring JDBC框架中处理异常 2057.8.1 问题描述  
2057.8.2 解决方案 2057.8.3 实现方法 2057.9 小结 209第8章 Spring中的事务管理 2118.1 事务管理的问题  
2118.2 选择事务管理器实现 2168.2.1 问题描述 2168.2.2 解决方案 2178.2.3 实现方法 2178.3 用事务管理  
器API编程式地管理事务 2188.3.1 问题描述 2188.3.2 解决方案 2188.3.3 实现方法 2188.4 用事务模板编程  
式地管理事务 2198.4.1 问题描述 2198.4.2 解决方案 2208.4.3 实现方法 2208.5 用经典的Spring AOP声明式  
地管理事务 2228.5.1 问题描述 2228.5.2 解决方案 2228.5.3 实现方法 2228.6 用事务通知声明式地管理事务  
2248.6.1 问题描述 2248.6.2 解决方案 2248.6.3 实现方法 2258.7 用@Transactional注解声明式地管理事务  
2268.7.1 问题描述 2268.7.2 解决方案 2268.7.3 实现方法 2268.8 设置传播事务属性 2278.8.1 问题描述  
2278.8.2 解决方案 2278.8.3 实现方法 2288.9 设置隔离事务属性 2318.9.1 问题描述 2318.9.2 解决方案  
2318.9.3 实现方法 2328.10 设置回滚事务属性 2388.10.1 问题描述 2388.10.2 解决方案 2388.10.3 实现方法  
2388.11 设置超时和只读事务属性 2398.11.1 问题描述 2398.11.2 解决方案 2398.11.3 实现方法 2398.12 用加  
载时织入来管理事务 2408.12.1 问题描述 2408.12.2 解决方案 2418.12.3 实现方法 2418.13 小结 244第9章  
Spring对ORM的支持 2459.1 直接使用ORM框架的问题 2459.1.1 利用Hibernate API和Hibernate XML映射  
持久化对象 2479.1.2 利用Hibernate API和JPA注解持久化对象 2509.1.3 用Hibernate作为JPA引擎持久化对  
象 2529.2 在Spring中配置ORM资源工厂 2559.2.1 问题描述 2559.2.2 解决方案 2569.2.3 实现方法 2569.3  
用Spring的ORM模板持久化对象 2619.3.1 问题描述 2619.3.2 解决方案 2619.3.3 实现方法 2629.4  
用Hibernate的上下文会话持久化对象 2679.4.1 问题描述 2679.4.2 解决方案 2679.4.3 实现方法 2679.5  
用JPA的上下文注入持久化对象 2709.5.1 问题描述 2709.5.2 解决方案 2709.5.3 实现方法 2709.6 小结 273  
第10章 Spring MVC框架 27410.1 用Spring MVC开发简单的Web应用程序 27410.1.1 问题描述 27410.1.2 解  
决方案 27410.1.3 实现方法 27510.2 将请求映射到处理程序 28410.2.1 问题描述 28410.2.2 解决方案  
28410.2.3 实现方法 28510.3 用Handler Interceptor拦截请求 28710.3.1 问题描述 28710.3.2 解决方案 28710.3.3  
实现方法 28810.4 解析用户区域 29010.4.1 问题描述 29010.4.2 解决方案 29010.4.3 实现方法 29010.5 外部化  
区分区域的文本信息 29210.5.1 问题描述 29210.5.2 解决方案 29210.5.3 实现方法 29210.6 按名称解析视图  
29310.6.1 问题描述 29310.6.2 解决方案 29310.6.3 实现方法 29310.7 将异常映射到视图 29610.7.1 问题描述  
29610.7.2 解决方案 29610.7.3 实现方法 29610.8 构造ModelAndView对象 29710.8.1 问题描述 29710.8.2 解决  
方案 29710.8.3 实现方法 29810.9 创建带有参数化视图的控制器 29910.9.1 问题描述 29910.9.2 解决方案  
29910.9.3 实现方法 30010.10 用表单控制器处理表单 30110.10.1 问题描述 30110.10.2 解决方案 30110.10.3  
实现方法 30210.11 用向导表单控制器处理多页表单 31310.11.1 问题描述 31310.11.2 解决方案 31310.11.3  
实现方法 31310.12 将多个动作分组到一个控制器 32110.12.1 问题描述 32110.12.2 解决方案 32110.12.3 实  
现方法 32110.13 创建Excel和PDF视图 32610.13.1 问题描述 32610.13.2 解决方案 32610.13.3 实现方法  
32610.14 用注解开发控制器 33010.14.1 问题描述 33010.14.2 解决方案 33010.14.3 实现方法 33010.15 小结  
336第11章 整合Spring与其他Web框架 33711.1 在通用的Web应用程序中访问Spring 33711.1.1 问题描述  
33711.1.2 解决方案 33711.1.3 实现方法 33811.2 整合Spring与Struts 1.x 34211.2.1 问题描述 34211.2.2 解决  
方案 34211.2.3 实现方法 34211.3 整合Spring与JSF 34711.3.1 问题描述 34711.3.2 解决方案 34811.3.3 实现方法  
34811.4 整合Spring与DWR 35211.4.1 问题描述 35211.4.2 解决方案 35211.4.3 实现方法 35211.5 小结 355第12  
章 Spring对测试的支持 35612.1 用JUnit和TestNG创建测试 35712.1.1 问题描述 35712.1.2 解决方案

35712.1.3 实现方法 35712.2 创建单元测试和集成测试 36112.2.1 问题描述 36112.2.2 解决方案 36212.2.3 实现方法 36212.3 单元测试Spring MVC控制器 37012.3.1 问题描述 37012.3.2 解决方案 37012.3.3 实现方法 37012.4 管理集成测试中的应用程序上下文 37412.4.1 问题描述 37412.4.2 解决方案 37412.4.3 实现方法 37512.5 将测试夹具注入到集成测试中 38012.5.1 问题描述 38012.5.2 解决方案 38112.5.3 实现方法 38112.6 管理集成测试中的事务 38412.6.1 问题描述 38412.6.2 解决方案 38412.6.3 实现方法 38512.7 在集成测试中访问数据库 38912.7.1 问题描述 38912.7.2 解决方案 39012.7.3 实现方法 39012.8 利用Spring的通用测试注解 39312.8.1 问题描述 39312.8.2 解决方案 39312.8.3 实现方法 39412.9 小结 395第三部分 高级主题第13章 Spring Security框架 39813.1 保护URL访问 39813.1.1 问题描述 39813.1.2 解决方案 39813.1.3 实现方法 39913.2 登录Web应用程序 40713.2.1 问题描述 40713.2.2 解决方案 40713.2.3 实现方法 40813.3 验证用户 41113.3.1 问题描述 41113.3.2 解决方案 41113.3.3 实现方法 41213.4 指定访问控制决策 41913.4.1 问题描述 41913.4.2 解决方案 41913.4.3 实现方法 42013.5 保护方法调用 42213.5.1 问题描述 42213.5.2 解决方案 42213.5.3 实现方法 42213.6 处理视图中的安全 42513.6.1 问题描述 42513.6.2 解决方案 42513.6.3 实现方法 42513.7 处理领域对象的安全 42713.7.1 问题描述 42713.7.2 解决方案 42713.7.3 实现方法 42713.8 小结 436 第14章 Spring Portlet MVC框架 43714.1 用Spring Portlet MVC开发简单的Portlet 43714.1.1 问题描述 43714.1.2 解决方案 43714.1.3 实现方法 43814.2 将Portlet请求映射到处理程序 44414.2.1 问题描述 44414.2.2 解决方案 44414.2.3 实现方法 44414.3 用简单表单控制器处理Portlet表单 45314.3.1 问题描述 45314.3.2 解决方案 45314.3.3 实现方法 45314.4 使用注解来开发Portlet控制器 46014.4.1 问题描述 46014.4.2 解决方案 46014.4.3 实现方法 46014.5 小结 465第15章 Spring Web Flow 46615.1 用Spring Web Flow管理简单的UI流程 46615.1.1 问题描述 46615.1.2 解决方案 46615.1.3 实现方法 46715.2 用不同的流程类型进行Web流程建模 47215.2.1 问题描述 47215.2.2 解决方案 47215.2.3 实现方法 47315.3 保护Web流程 48215.3.1 问题描述 48215.3.2 解决方案 48315.3.3 实现方法 48315.4 在Web流程中持久化对象 48415.4.1 问题描述 48415.4.2 解决方案 48515.4.3 实现方法 48515.5 集成Spring Web Flow与JSF 49015.5.1 问题描述 49015.5.2 解决方案 49015.5.3 实现方法 49115.6 小结 496第16章 Spring远程服务和Web服务 49716.1 通过RMI暴露和调用服务 49816.1.1 问题描述 49816.1.2 解决方案 49816.1.3 实现方法 49916.2 通过HTTP公开和调用服务 50216.2.1 问题描述 50216.2.2 解决方案 50216.2.3 实现方法 50216.3 选择Web服务的开发方法 50516.3.1 问题描述 50516.3.2 解决方案 50516.3.3 实现方法 50516.4 利用XFire公开和调用Web服务 50716.4.1 问题描述 50716.4.2 解决方案 50716.4.3 实现方法 50716.5 定义Web服务契约 51216.5.1 问题描述 51216.5.2 解决方案 51216.5.3 实现方法 51316.6 利用Spring-Ws实现Web服务 51616.6.1 问题描述 51616.6.2 解决方案 51616.6.3 实现方法 51716.7 利用Spring-Ws调用Web服务 52216.7.1 问题描述 52216.7.2 解决方案 52216.7.3 实现方法 52216.8 利用XML封送技术开发Web服务 52516.8.1 问题描述 52516.8.2 解决方案 52516.8.3 实现方法 52516.9 利用注解创建服务端点 53016.9.1 问题描述 53016.9.2 解决方案 53016.9.3 实现方法 53016.10 小结 531第17章 Spring对EJB和JMS的支持 53217.1 用Spring创建EJB 2.x组件 53217.1.1 问题描述 53217.1.2 解决方案 53217.1.3 实现方法 53317.2 在Spring中访问EJB 2.x组件 53717.2.1 问题描述 53717.2.2 解决方案 53817.2.3 实现方法 53817.3 在Spring中访问EJB 3.0组件 54217.3.1 问题描述 54217.3.2 解决方案 54217.3.3 实现方法 54217.4 用Spring发送和接收JMS消息 54517.4.1 问题描述 54517.4.2 解决方案 54517.4.3 实现方法 54617.5 在Spring中创建消息驱动的POJO 55817.5.1 问题描述 55817.5.2 解决方案 55917.5.3 实现方法 55917.6 小结 564第18章 Spring对JMX、电子邮件和调度的支持 56518.1 将Spring Bean导出为JMX MBean 56518.1.1 问题描述 56518.1.2 解决方案 56518.1.3 实现方法 56618.2 发布和监听JMX通知 57518.2.1 问题描述 57518.2.2 解决方案 57518.2.3 实现方法 57518.3 在Spring中访问远程JMX MBean 57718.3.1 问题描述 57718.3.2 解决方案 57718.3.3 实现方法 57718.4 用Spring的电子邮件支持来发送电子邮件 58018.4.1 问题描述 58018.4.2 解决方案 58018.4.3 实现方法 58018.5 用Spring的JDK Timer支持进行调度 58718.5.1 问题描述 58718.5.2 解决方案 58718.5.3 实现方法 58718.6 利用Spring的Quartz支持进行调度 58918.6.1 问题描述 58918.6.2 解决方案 58918.6.3 实现方法 58918.7 小结 593第19章 Spring中的脚本编程 59419.1 用脚本语言实现Bean 59419.1.1 问题描述 59419.1.2 解决方案 59419.1.3 实现方法 59419.2 将Spring Bean注入脚本 59819.2.1 问题描述 59819.2.2 解决方案 59819.2.3 实现方法 59819.3 从脚本中刷新Bean 60019.3.1 问题描述 60019.3.2 解决方案 60019.3.3 实现方法 60019.4 定义内联脚本 60119.4.1 问题描述 60119.4.2 解决方案 60119.4.3 实现方法 60119.5 小结 603





## 章节摘录

插图：在本章中，你学习了如何通过Spring远程服务支持来公开和调用远程服务。

Spring支持许多远程服务技术，例如RMI、Hessian、Burlap和HTTP Invoker。

虽然Spring没有对Web服务提供类似的支持，但是有一个名为XFire的外部类库可以实现类似功能。

Spring远程服务支持对于不同的远程服务技术都是一致的。

在服务器端，Spring让你通过ServiceExporter公开任意Bean作为远程服务。

在客户端，Spring提供了各种各样的代理工厂Bean，让你可以为远程服务创建本地代理，以便能够当它是本地Bean一样地使用远程服务。

你还了解了采用Contract First方法开发Web服务的优点，以及如何利用Spring - WS来开发这种服务。

第一步是定义服务契约，它由两个部分组成：数据契约和服务契约。

数据契约通过XSD定义，而服务契约则是通过WSDL定义的。

你必须自己定义XSD文件，然后从XSD文件中生成WSDL文件。

一旦为Web服务定义好契约，就可以根据这个契约实现服务本身了。

服务请求由端点进行处理，它们类似于Web应用程序中的控制器。

你可以利用XML解析或者XML封送技术来实现端点。

你可以创建服务客户端，根据Web服务的契约来调用这个服务。

还可以利用：XML解析或者XML封送技术来实现服务客户端。

在下一章中，你将会了解到Spring对两种主要的Java EE技术即EJB和JMS的支持。

## <<Spring攻略>>

### 媒体关注与评论

“ .....此书的内容如此朴实无华，但正是我苦苦寻觅的，这让我非常震惊！

” ——DamOdar Chetty, Software Engineering Solutions, Inc. “ 我很少发表评论，但这次是例外。

这是迄今为止最好的图书，你一定会手不释卷。

此书可读性极好，内容结构严谨有序。

我真的很惊奇，它怎么能够如此详细。

” ——Amazon.com评论

## <<Spring攻略>>

### 编辑推荐

《Spring攻略》是目前国外人气最高、口碑最好的一本介绍Spring的图书。

《Spring攻略》内容详实，示例生动丰富，代码实用。

可操作性强。

它不仅涵盖了Spring 2.5从基础概念到高级应用的所有主题。

而且深入浅出地介绍了几种常见的Spring项目。

其参考价值不言而喻。

书中采用了“问题描述/解决方案/实现方法”的方式，读者可以轻松地查找特定问题的解决方案，事半功倍。

通过《Spring攻略》，读者可以迅速掌握Spring来构建强大的企业级Java应用程序，成为Spring高手。

Spring专家力作理论与实践完美结合问题描述，解决方案，实现方法

## <<Spring攻略>>

### 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>