

<<精通Spring>>

图书基本信息

书名：<<精通Spring>>

13位ISBN编号：9787121072987

10位ISBN编号：712107298X

出版时间：2008-10

出版时间：电子工业出版社

作者：罗时飞 著

页数：521

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

毫不夸张地说，Spring 2.5是一套有关Java EE API的百科全书，它针对各种Java EE API的使用都提供了一流的、一致的抽象和集成工作，从而统一Java EE API暴露给开发者的客户视图。

开发者都知道，Java EE API的使用非常烦琐，许多与业务无关的技术细节需要开发者悉心打理。

稍有不慎，各种Java EE问题随之而来，而Spring 2.5正是为解决Java EE编程模型中的这些问题而出现的。

为完成各种Java EE API的集成工作，Spring开发团队提供了Spring元框架，即控制反转容器(IOC)和AOP技术实现。

所有的Java EE API集成工作都是在这一元框架基础之上构建的。

从目前来看，Spring 2.5主要提供了三方面的Java EE API集成：DAO层集成技术；Java EE服务及技术；Web层支持。

本书正是围绕Spring 2.5中的上述各项内容而准备的。

时隔两年后，《精通Spring 2.0》（第二版）成功写作完成，并出版发行。

同《精通Spring》（第一版）相比，本次改进、新增的内容非常多，下面总结了本书的特点。

<<精通Spring>>

内容概要

是关于Spring2.5的权威教程，是Java/Java EE开发者必备的参考书。

《精通Spring：深入Java EE开发核心技术》详尽、系统地介绍了JavaEE的基础知识、Spring 2.5的各种功能，以及Spring 2.5的高级使用技巧和最佳实践。

全书共分为5篇；第1篇为综述，主要围绕Java EE5、Spring展开；第2篇介绍Spring2.5核心技术，主要围绕Spring元框架进行阐述；第3篇介绍DAO层集成技术，主要围绕JDBC、Hibernate和JPA等持久化技术展开论述，针对Spring使能应用的事务管理和集成测试，也进行了相关介绍；第4篇介绍Java EE服务及技术的集成，主要围绕企业应用中使用的各种Java EE服务及技术展开论述；第5篇介绍Spring 2.5高级特性，主要从忘却的Spring高级话题和Spring最佳实践角度给出论述；附录A完整地介绍了Spring 2.5支持的各种命名空间及其中的所有元素。

全书理论与实践并重，通过大量的实例帮助读者尽快掌握Spring 2.5的使用技巧，从而提高《精通Spring：深入Java EE开发核心技术》的参考、阅读价值。

《精通Spring：深入Java EE开发核心技术》适合作为Java/Java EE开发者、系统分析师和架构师的参考书，同时，《精通Spring：深入Java EE开发核心技术》非常适合于高校相关专业的学生，以及对Java/Java EE开发有兴趣的各类开发者。

<<精通Spring>>

作者简介

罗时飞，独立Java EE顾问。
西安交通大学硕士。
将近100年的Java EE开发、架构经验，理论与实践并重。
对Java EE平台、敏捷方法及开源技术有深入研究。
出版图书多本，曾被评为电子工业出版社的“优秀作译者”。
他一直在依托网站专职从事企业级Java的推广工作。
其近期代表作品有：《敏捷持续集成（Cruise Control版）——高效研发之道》、《敏捷Acegi、CAS——构建安全的Java系统》、《精通EJB3.0》（译）、《精通Spring 2.0》。

书籍目录

第1篇 综述第1章 Java EE 51.1 Java EE 5引入的新特性1.2 进入EJB 3.0时代1.3 Java EE开发模型的限制性1.4 小结第2章 步入Spring 2.52.1 挑战Java EE 5开发模型2.1.1 轻量级开发模型2.1.2 倡导敏捷开发2.1.3 Spring 2.5的架构价值2.2 有所为和有所不为2.2.1 Spring 2.5提供的功能2.2.2 排除在外2.3 Spring 2.5时代的到来2.4 小结第3章 获得Spring 2.5发布版和源码3.1 获得Spring 2.5持续发布版3.2 获得持续更新的Spring 2.5项目源码3.3 小结第4章 启动Spring 2.5使能项目4.1 开发平台的搭建4.1.1 JDK的安装及设置4.1.2 选用Eclipse IDE和WTP4.1.3 借助插件调试Web应用4.1.4 获取及安装Spring IDE4.2 Spring IDE的使用4.3 小结第2篇 Spring 2.5核心技术第5章 控制反转容器5.1 有关DI容器的背景知识5.2 BeanFactory和ApplicationContext5.3 宿主DI容器配置元数据的不同方式5.3.1 基于XML的DI容器配置元数据5.3.2 基于注解的DI容器配置元数据5.4 基于泛型访问DI容器5.5 支持的不同依赖注入类型5.5.1 设值注入5.5.2 构建器注入5.5.3 属性注入5.5.4 方法注入5.6 Autowiring策略5.6.1 autowire属性5.6.2 元素的dependency-check属性5.6.3 @Required注解5.6.4 @Autowired注解5.6.5 细粒度控制Autowiring策略5.6.6 借用元素5.7 善待depends-on属性5.8 抽象和子Bean定义5.9 别名 (Alias) 5.10 外在化应用参数的配置5.10.1 元素5.10.2 元素5.11 受管Bean的作用范围5.11.1 单例和原型5.11.2 仅仅适合于Web环境的三种作用范围5.12 在Web应用中使用DI容器5.12.1 往Web应用中加载DI容器5.12.2 复合多个配置文件5.12.3 于Web应用中操控DI容器5.12.4 国际化和本地化消息资源5.13 探索命名空间5.13.1 元素5.13.2 元素5.13.3 元素5.13.4 元素5.13.5 元素5.13.6 元素5.14 使用命名空间5.15 操控资源5.15.1 内置的Resource继承链5.15.2 借助DI容器访问各种资源5.15.3 妙用classpath*前缀5.16 回调接口集合及其触发顺序5.16.1 BeanNameAware回调接口5.16.2 BeanClassLoaderAware回调接口5.16.3 BeanFactoryAware回调接口5.16.4 ResourceLoaderAware回调接口5.16.5 ApplicationEventPublisherAware回调接口5.16.6 MessageSourceAware回调接口5.16.7 ApplicationContextAware回调接口5.16.8 @PostConstruct注解5.16.9 InitializingBean回调接口5.16.10 元素的init-method属性5.16.11 @PreDestroy注解5.16.12 DisposableBean回调接口5.16.13 元素的destroy-method属性5.17 小结第6章 面向切面编程6.1 AOP背景知识6.2 AspectJ 6介绍6.2.1 AspectJ的安装及使用6.2.2 Before装备6.2.3 AfterReturning装备6.2.4 AfterThrowing装备6.2.5 After装备6.2.6 Around装备6.2.7 引入 (Introduction) 6.3 Spring AOP的基本概念6.4 Spring AOP的老式用法6.4.1 Before装备6.4.2 基于ProxyFactoryBean的手工代理6.4.3 AfterReturning装备6.4.4 AfterThrowing装备6.4.5 Around装备6.4.6 Introduction引入6.4.7 使用自动代理特性6.4.8 切换代理机制6.4.9 基于ProxyFactory的编程代理6.5 基于@AspectJ的Spring AOP6.5.1 声明切面、pointcut和装备6.5.2 各种装备的使用6.5.3 切换代理机制6.5.4 控制各装备的触发顺序6.5.5 pointcut表达语言6.6 基于元素的AOP6.6.1 声明切面、pointcut和装备6.6.2 各种装备的使用6.6.3 元素6.6.4 切换代理机制6.7 借用AspectJ 6进行领域对象的DI操作6.7.1 直接使用AnnotationBean-ConfigurerAspect切面6.7.2 @Configurable注解6.7.3 借助aop.xml控制启用的特定切面6.7.4 元素6.7.5 元素6.7.6 借用元素6.8 小结第3篇 DAO层集成技术第7章 DAO抽象支持7.1 背景7.2 DAO集成支持7.2.1 DataAccessException异常体系7.2.2 DaoSupport继承链7.2.3 DataAccessUtils实用类7.3 小结第8章 JDBC集成8.1 背景知识及示例8.2 Spring对JDBC提供的支持8.3 运行JDBC版PetClinic实例8.4 JdbcTemplate及相应的支持类8.4.1 JdbcTemplate核心类8.4.2 JdbcDaoSupport支持类8.5 NamedParameterJdbcTemplate及相应的支持类8.5.1 NamedParameterJdbcTemplate模板类8.5.2 NamedParameterJdbcDao-Support支持类8.6 SimpleJdbcTemplate及相应的支持类8.6.1 SimpleJdbcTemplate模板类8.6.2 SimpleJdbcDaoSupport支持类8.6.3 SimpleJdbcInsert辅助类8.6.4 基于JDBC的PetClinic综合示例分析8.7 内置的DataSource继承链8.7.1 用于测试目的的DriverManagerDataSource8.7.2 用于测试目的的SimpleDriverDataSource8.7.3 用于测试目的的单连接DataSource8.7.4 Apache DBCP数据源8.7.5 Java EE容器内置的数据源8.7.6 LazyConnectionDataSource-Proxy数据源8.7.7 TransactionAwareDataSource-Proxy数据源8.7.8 UserCredentialsDataSource-Adapter数据源8.7.9 IsolationLevelDataSource-Adapter数据源8.7.10 WebSphereDataSource-Adapter数据源8.7.11 IsolationLevelDataSource-Router数据源8.8 将JDBC操作

建模成Java对象8.8.1 SqlUpdate辅助类8.8.2 UpdatableSqlQuery辅助类8.8.3 MappingSqlQuery辅助类8.8.4 SqlFunction辅助类8.9 与存储过程交互8.9.1 JdbcTemplate针对存储过程提供的支持8.9.2 StoredProcedure辅助类8.9.3 SimpleJdbcCall辅助类8.10 处理大批量数据8.10.1 JdbcTemplate内置的batchUpdate()方法8.10.2 SimpleJdbcTemplate内置的batchUpdate()方法8.10.3 BatchSqlUpdate辅助类8.11 基于JDBC的LOB集成支持8.11.1 运行及分析imagedb示例应用8.11.2 NativeJdbcExtractor继承链8.11.3 操作LOB字段8.12 如何获得和生成主键8.12.1 KeyHolder及GeneratedKey-Holder实现者8.12.2 DataFieldMaxValue-Incrementer继承链8.13 对行集的支持8.13.1 JdbcTemplate内置的queryForRowSet()方法集合8.13.2 NamedParameterJdbcTemplate内置的queryForRowSet()方法集合8.14 JDBC最佳实践8.15 小结第9章 事务集成9.1 背景知识及示例9.2 Spring对事务提供的支持9.3 Spring眼中的事务管理策略9.3.1 事务定义9.3.2 各种PlatformTransaction-Manager实现9.4 编程式事务9.4.1 TransactionTemplate及相关回调接口9.4.2 使用@Transactional注解和元素9.4.3 拥抱EJB 3.0引入的@TransactionalAttribute注解9.5 声明式事务9.5.1 TransactionProxyFactoryBean辅助类9.5.2 元素9.6 在AspectJ 6应用中使用@Transactional9.6.1 直接使用Annotation-TransactionAspect切面9.6.2 借用元素9.6.3 借用元素9.7 事务集成高级特性9.7.1 Java EE应用服务器的事务集成9.7.2 元素9.7.3 选择合适的事务策略9.8 小结第10章 单元和集成测试10.1 背景知识及示例10.2 Spring对集成测试的支持10.2.1 ReflectionTestUtils实用类10.2.2 运行PetClinic中的集成测试类10.3 遗留JUnit 3.8集成测试支持10.3.1 AbstractSingleSpringContext-Tests支持类10.3.2 AbstractDependencyInjection-SpringContextTests支持类10.3.3 AbstractTransactionalSpring-ContextTests支持类10.3.4 AbstractTransactionalData-SourceSpringContextTests支持类10.3.5 AbstractAnnotationAware-TransactionalTests支持类10.4 新引入的TestContext集成测试框架10.4.1 面向开发者的支持类10.4.2 AbstractJUnit38Spring-ContextTests支持类10.4.3 AbstractTransactionalJUnit38-SpringContextTests支持类10.4.4 AbstractJUnit4Spring-ContextTests支持类10.4.5 AbstractTransactionalJUnit4-SpringContextTests支持类10.5 集成测试最佳实践10.6 小结第11章 Hibernate集成11.1 背景知识及示例11.2 Hibernate Tools介绍11.2.1 Ant支持11.2.2 Eclipse支持11.3 Spring对Hibernate提供的支持11.4 运行Hibernate版PetClinic实例11.5 基于Hibernate集成的CRUD操作11.5.1 HibernateTemplate模板类11.5.2 HibernateCallback回调接口11.5.3 关于SessionFactory.getCurrentSession()方法的使用11.6 LocalSessionFactoryBean11.7 AnnotationSession-FactoryBean11.8 事务管理支持11.9 基于Hibernate的LOB处理11.10 为imagedb示例启用JTA事务11.11 集成测试支持11.11.1 分析Hibernate版PetClinic实例的集成测试工作11.11.2 混合使用JDBC和Hibernate11.12 小结第12章 Java持久化API集成12.1 背景知识及示例12.2 Spring对JPA提供的支持12.3 基于JPA集成的CRUD操作12.3.1 JpaTemplate模板类12.3.2 JpaCallbck回调接口12.3.3 @PersistenceContext注解12.4 AbstractEntityManager-FactoryBean继承链12.4.1 LocalEntityManager-FactoryBean辅助类12.4.2 LocalContainerEntity-ManagerFactoryBean辅助类12.4.3 DataSourceLookup继承链12.5 事务管理支持12.6 装载期织入(LTW) 12.6.1 ReflectiveLoadTimeWeaver实现类12.6.2 InstrumentationLoad-TimeWeaver实现类12.6.3 LoadTimeWeaver继承链在JPA集成中的应用12.7 SharedEntityManagerBean12.8 集成测试支持12.8.1 AbstractJpaTests支持类12.8.2 AbstractAspectjpaTests支持类12.8.3 混合使用JDBC和JPA12.9 小结第4篇 集成Java EE服务及技术第13章 JNDI集成13.1 背景知识及示例13.2 Spring对JNDI提供的支持13.3 JndiObjectFactoryBean13.4 元素13.5 JndiTemplate和JndiCallback的使用13.6 小结第14章 EJB 3.0集成14.1 背景知识及示例14.2 Spring对开发EJB 3.0组件提供的支持14.3 Spring对访问EJB 3.0组件提供的支持14.3.1 借助JndiObjectFactory-Bean辅助类14.3.2 org.springframework.ejb.access包14.3.3 元素和元素14.4 关于遗留EJB 2.x支持14.5 小结第15章 线程池和任务调度集成15.1 Spring提供的线程池支持15.1.1 SyncTaskExecutor执行器15.1.2 SimpleAsyncTaskExecutor执行器15.1.3 ThreadPoolTaskExecutor和ConcurrentTaskExecutor执行器15.1.4 TimerTaskExecutor执行器15.1.5 SimpleThreadPool-TaskExecutor执行器15.1.6 commonj.WorkManager-TaskExecutor执行器15.1.7 jca.work.WorkManager-TaskExecutor继承链15.2 Spring提供的任务调度支持15.2.1 针对java.util.Timer的任务调度支持15.2.2 针对Quartz的任务调度支持15.2.3 针对java.util.concurrent的任务调度支持15.2.4

针对CommonJ的任务调度支持15.3 小结第16章 Java消息服务集成16.1 背景知识及示例16.2
Spring对JMS消息提供的支持16.3 借助JmsTemplate发送JMS消息16.4 同步和异步消费JMS消息16.4.1
借助JmsTemplate同步接收JMS消息16.4.2 AbstractMessageListener-Container容器16.4.3 元素16.5
JMS事务管理16.6 小结第17章 JavaMail集成17.1 背景知识及示例17.2 Spring对JavaMail提供的支持17.3
发送简单邮件17.4 发送含有附件的邮件17.5 发送含有HTML和内嵌资源的邮件17.6 小结
第18章 远程服务集成18.1 远程服务背景知识及示例18.2 Spring对远程服务提供的支持18.3
RMI/RMI-IIOP集成18.4 Hessian和Burlap集成18.4.1 DispatcherServlet和HttpRequestHandlerServlet辅助类18.4.2
宿主在Sun JDK 6.0内置的HTTP服务器中18.5 HTTP Invoker支持18.6 Web服务集成18.6.1
JAX-RPC集成18.6.2 JAX-WS集成18.7 基于JMS的远程服务18.8 小结第19章 Java管理扩展集成19.1
背景知识及示例19.2 Spring对JMX提供的支持19.3 自动注册MBean组件19.3.1 关于MBeanExporter的autodetectMode和registrationBehavior属性19.3.2
Hibernate暴露的StatisticsService MBean19.4 将POJO导出成MBean组件19.5 控制MBean组件的管理接口19.5.1
AbstractConfigurable-MBeanInfoAssembler继承链19.5.2 基于注解的Metadata-MBeanInfoAssembler19.5.3
和元素19.5.4 面向异步处理的Lifecycle接口19.6 控制MBean组件的ObjectName19.6.1
KeyNamingStrategy实现类19.6.2 IdentityNamingStrategy实现类19.6.3 MetadataNamingStrategy实现类19.7
发送与接收JMX通知19.8 通过应用访问MBean组件19.9 小结第20章 Java EE连接器架构集成20.1
背景知识及示例20.2 Spring对JCA提供的支持20.3 CciTemplate及相关回调接口20.4 将JCA操作建模成Java对象20.5
事务管理20.6 宿主在JCA适配器中的DI容器20.6.1 将SpringContextResource-Adapter部署到RAR中20.6.2
ResourceAdapterFactory-Bean辅助类20.7 小结第5篇 Spring 2.5高级特性第21章 忘却的Spring高级话题21.1
分发和监听事件21.2 AOP拦截器链21.3 DataSourceUtils、Session-FactoryUtils、EntityManager-FactoryUtils21.4
Web层集成支持21.5 元素21.6 如何优雅地销毁DI容器21.6.1 Web应用类型21.6.2 EJB应用类型21.6.3 Java SE应用、集成测试类型21.7
DI容器的分层管理21.8 脚本集成21.9 小结第22章 Spring最佳实践22.1 注重分层架构设计22.2 合理采纳注解技术22.3
日志管理策略22.4 善待Java EE容器内置的类装载机22.5 逐步采纳Spring 2.522.6 小结附录A 基于XML Schema的权威配置指南A.1
XML配置文件A.2 命名空间A.2.1 元素A.2.2 元素A.2.3 元素A.2.4 元素A.3 命名空间A.3.1 元素A.3.2 元素A.3.3 元素A.3.4
元素A.3.5 元素A.3.6 元素A.4 命名空间A.4.1 元素A.4.2 元素A.4.3 元素A.4.4 元素A.4.5 元素A.4.6 元素A.4.7
元素A.4.8 元素A.5 命名空间A.5.1 元素A.5.2 元素A.5.3 元素A.6 命名空间A.6.1 元素A.6.2 元素A.6.3 元素A.7
命名空间A.7.1 元素A.7.2 元素A.7.3 元素A.8 命名空间A.8.1 元素A.8.2 元素A.9 命名空间A.9.1 元素A.9.2 元素A.9.3
元素A.9.4 元素A.10 命名空间A.11 小结

章节摘录

第1章Java EE 5：简要回顾Java EE 5中各种组件的开发，Annotation注解技术被Java EE广泛应用；随后介绍了EJB 3.0组件的开发。

在Java EE 5全面拥抱Annotation注解技术的同时，也暴露出Java EE开发模型的若干局限性，而这方面的内容也在本章给出了讨论。

第2章步入Spring 2.5:从宏观上将Spring 2.5告知给开发者。

比如，它所倡导的POJO开发模型是如何超越Java EE开发模型的、Spring2.5的架构价值、Spring2.5涉足的具体技术领域等。

第3章获得Spring 2.5发布版和源码：为开发和调试Spring使能应用，开发者必须懂得如何下载Spring 2.5。

第4章启动Spring 2.5使能项目：“工欲善其事，必先利其器”，为如期交付高质量的复杂软件产品，研发团队必须搭建一流的开发平台。

与此同时，我们还对Spring IDE的使用给出了详细介绍，缺乏Spring IDE的Spring 2.5使能项目的开发过程会显得非常生涩。

简化Java EE 5的开发和部署模型是整个企业级Java领域一致的呼声，Java EE 5确实也达到了这一目标。

在大范围采用Java SE 5引入的Annotation注解技术后，部署描述符几乎是多余的了。

如今，资源和组件的依赖性注入也可以交由Annotation注解完成。

无论是EJB 3.0组件还是Servlet，无论是Web服务(JAX-WS)还是JSF受管Bean（后端Bean），Annotation注解技术的应用都随处可见。

<<精通Spring>>

编辑推荐

专业人士，权威经典。

Spring 2.5是迄今为止完美的Java EE架构级框架，全面深入、多维度演绎Spring 2.5的各个方面，本书蕴含作者多年Java EE研发实践及经验。

凝聚Java魅力，成就开发专家 看清Java万花的本质，从复杂的表象中寻找普遍的规律，深刻理解Java的核心思想。

关于Spring2.5的权威教程，是Java / Java EE开发者、架构师必备的参考书 对JavaEE5及Spring2.5进行了综述。

包括Java EE5，步入Spring2.5，获得Spring2.5发布版和源码。

启动Spring2.5使能项目 详细介绍Spring2.5核心技术。

包括控制反转容器，面向切面编程 对DAO层集成技术进行了详细阐述。

包括DAO抽象支持。

JDBC集成，事务集成，单元和集成测试。

Hibernate集成，Java持久化API集成 深入讲解Java EE服务及技术集成。

包括JNDI集成，EJB3.0集成.线程池和任务调度集成。

Java消息服务集成，Java Mail集成，远程服务集成。

Java管理扩展集成，Java EE连接器架构集成 专注于Spring2.5高级特性的研究。

包括忘却的Spring高级话题，Spring最佳实践 对Spring2.5内置的各种XML Schema支持进行了全方位阐述。

包括基于XML Schema的权威配置指南 全书理论与实践并重，通过大量的实例帮助读者尽快掌握Spring2.5的各种基本和高级使用技巧，从而提高本书的参考和阅读价值

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>