开发专家之Sun ONE

精通Spring

Mastering Spring

Spring
thaw the ice of J2EE

罗时飞 编著
飞思科技产品研发中心 监制

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

开发专家之 Sun ONE

精通 Spring

罗时飞　编著

飞思科技产品研发中心　监制

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书深入剖析了当前流行的轻量级开发框架 Spring 技术。全书共分成 3 部分。第一部分，重点阐述 Spring 的架构。这部分内容采用循序渐进的方式，带领开发者进入 Spring 中，主要阐述了 Spring IoC 和 Spring AOP。第二部分，重点阐述 Spring 的使用。这部分内容从简化 Java/J2EE 的角度出发，从 J2EE 平台各个技术层面分析并给出大量的研究实例，对 Spring 提供的 API 进行阐述。主要阐述了 Spring 对 J2EE API 提供的服务抽象。第三部分，重点阐述 Spring 高级专题。这部分内容重点对视图技术进行了研究，因为对于开发 Web 应用而言，前端界面的开发往往工作量很大。因此，使用合理的视图技术开发 Web 应用对于项目的成功与否很关键。另外，Web 应用的安全性往往也是企业应用中最为重要的需求之一，而用于 Spring 的 Acegi 安全框架很好地解决了这个问题，这也是第三部分重点研究的内容之一。

本书所有实例源代码文件请到 http://www.fecit.com.cn “下载专区” 下载。

本书适合所有的 Java、J2EE 开发者阅读。

图书在版编目（CIP）数据

精通 Spring / 罗时飞编著. —北京：电子工业出版社，2005.4
（开发专家之 Sun ONE）
ISBN 7-121-01051-8

Ⅰ.精... Ⅱ.罗... Ⅲ.计算机网络－程序设计 Ⅳ.TP393

中国版本图书馆 CIP 数据核字（2005）第 023551 号

责任编辑：王 蒙
印 刷：北京东光印刷厂
出版发行：电子工业出版社
北京海淀区万寿路 173 信箱 邮编：100036
经 销：各地新华书店
开 本：787×1092 1/16 印张：27.25 字数：697.6 千字
印 次：2005 年 4 月第 1 次印刷
印 数：6 000 册 定价：39.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：010-68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

序

真的是不经意的工作机会，才使得我有从事 Spring 方面的项目架构和开发经验，不只是 Spring，还有 Hibernate 和 Tapestry 也在其中。在 Java Web 框架中，成熟的 Tapestry 现在是前端开发领域中的一枝独秀。Hibernate 成为了 O/R Mapping 领域事实上的标准，尤其是它对 EJB 3.0 的巨大贡献。Spring 更不用说了，它使得开发者和具体 J2EE 平台技术处于“松耦合”的状态。它们的强强联手，使得开发软件产品过程确实是一种快乐，发自开发者内心的快乐。快乐是无限的，您也许不想错过。

Tapestry 推崇 OO 方式开发 Web 前端。它将 HTTP 底层 API 技术屏蔽起来，尤其是开发者再也不用直接面对 JSP 和 Servlet 了。Hibernate 使得 Java 开发者能够高效地对 RDBMS 进行 CRUD 操作，而不管各种 RDBMS 所运行的平台、版本如何。尤其是开发者再也不用直接面对 JDBC、SQL 语句了，至少大部分场合是如此。Spring 是各种 Web 框架的黏合剂，无论是在 Open Source 领域，还是在非 Open Source 领域。当然，Tapestry 和 Hibernate 也在 Spring 的操控范围内。架构级的 Spring 在主导着整个 J2EE 社区开发 Web 应用的方向。因此，Spring 是本书讨论的重点，但是 Tapestry、Hibernate 也将在本书的视野中。

Spring，中文含义暂且理解为“春天”。春天，意味着万物复苏，而 Spring 将担当此任。

Spring IoC，借助于依赖注入设计模式，使得开发者不用理会对象自身的生命周期及其关系，而且能够改善开发者对模式的使用。请记住，管理单个的对象本身不是件难事，难就难在处于 Team 中的各个对象，因为它们之间的关系往往是千变万化的。这正如开发者在不同团队中所处的不同角色一样。借助于 Spring IoC，不仅能够使得应用中对象的关系更加清晰、一致，而且还使得一切对象可控。最为重要的一点是，对象本身的生命周期及对象之间的关系不用再让开发者费神了。

Spring AOP，借助于 Spring 实现的拦截器，开发者能够实现以声明方式使用企业级服务，比如安全性服务、事务服务。请记住，AOP 能够合理地补充 OOP 技术，Spring AOP 合理地补充了 Spring IoC 容器。没有 Spring IoC 的 Spring AOP 是不完善的，没有 Spring AOP 的 Spring IoC 是不健壮的。借助于 Spring AOP，开发者能够高效地使用 J2EE 企业服务。声明式、基于元数据访问企业级服务，这些都是 Spring AOP 的操控范围。

Spring 服务抽象，借助于各种 J2EE API 抽象，使得开发者能够一致地使用 J2EE 技术，而不管具体是使用什么 J2EE API。借助于 Spring 服务抽象，使得应用代码大大减少，请记住“更少的代码、更少的 Bug”的原则。另外，通过研究 Spring 服务抽象，使得开发者能够快速掌握各种 J2EE API 的核心内容，因为 Spring 抽象服务是架构在 J2EE API 基础之上的。请记住，Template 方法实现是 Spring 服务抽象中至关重要的开发利器。缺少 Template 方法实现，会使 Spring 操作 JNDI、Transaction、JMS、Hibernate 等企业级服务显得生涩。

Spring IoC + Spring AOP + Spring 服务抽象，一起形成了 Spring。这样一个有机的整体，使得构建轻量级的 J2EE 架构成为可能，而且事实证明，非常有效。请记住，Team Work 往往比单打独斗更具威力。如果剥离 IoC、AOP 或服务抽象中的任何一者，则 Spring 必然受伤不轻。正因为 Spring 提供了这样一套完整的利器，才使得它备受开发者推崇，这也是它

同其他类似 Web 框架的区别所在。我们向来不相信有“银弹”，即使它存在，在它未到来之前，赶紧选择 Spring 吧。否则一旦“银弹”到来，您也可能措手不及！

如果您还没有接触过 Spring，则作为一名 J2EE 开发者，不熟悉 Spring，有点勉为其难。因为，架构级的 Spring 框架已成为主流，它已经冲入了 J2EE 的核心，并在引领整个 J2EE 开发、架构的方向。现在正是学习并使用 Spring 开发企业级 Java 应用的时机。如果您还在观望的话，即使您不购买这本图书，作者还是希望您能够去使用 Spring，从而体验 Spring 深层次的开发经验。基于我们的架构、开发经验，Spring 富有内涵。随着您对 Spring 的日益使用，您会慢慢喜欢上它。

本书总共分成 3 部分。第一部分，重点阐述 Spring 的架构。这部分内容循序渐进带领开发者进入 Spring 中。主要在于阐述 Spring IoC 和 Spring AOP。第二部分，重点阐述 Spring 的使用。这部分内容从简化 Java/J2EE 的角度出发，从 J2EE 平台各个技术层面分析，并给出大量的研究实例，对 Spring 提供的 API 进行阐述。主要在于阐述 Spring 对 J2EE API 提供的服务抽象。第三部分，重点阐述 Spring 高级专题。这部分内容重点对视图技术进行了研究，因为对于开发 Web 应用而言，前端界面的开发往往工作量很大。因此，使用合理的视图技术开发 Web 应用对于项目的成功与否很关键。另外，Web 应用的安全性往往也是企业应用中最为重要的需求之一，而用于 Spring 的 Acegi 安全框架很好地解决了这个问题，这也是第三部分重点研究的内容之一。

在架构软件、开发软件过程中，架构师（设计人员、开发者）往往会低估很多不重要的开发环节，最终导致了软件系统发布的延期。这其中，往往会有很多未预期的因素，比如人员变动、对新技术不熟悉、对产品需求的把握不准确，导致产品的延期。对于写书而言，也存在类似的因素。作者在编著《精通 Spring》这本书之前，也低估了各种因素而导致图书交付日期的延迟。但是，这些现在都不是问题，因为我们（与此图书相关的角色）都克服了所有的困难。父母对全家的悉心照顾，爱妻的支持，乖巧听话的、还未满周岁的女儿，出版社为保证图书质量而提出的、富有成效的修改建议，还有那些关心此书的所有朋友、读者都是保证本书能够顺利地并高质量地出版的重要前提。在将近半年的时间中，他们都默默为此书付出了很多、很多。作者很感谢他们，没有他们的付出，就不可能有此书的诞生。当然，也感谢 Spring 开发团队的努力。请记住，热情、激情、友情、亲情是我们生活、工作的动力。

将此书献给所有的 Java、J2EE 开发者。

J2EE 架构师　罗时飞

2005 年 3 月于广州

前　言

关于本套丛书

从来没有任何事物像互联网那样，对人类的活动产生如此深刻的影响，无论是政府、企业，以及个人，莫不如此。与此同时，IT 产业也正面临着一场变革——传统应用向基于 Internet/Web 的服务模式转化。

翻开历史，我们可以看到互联网的形成和发展就是以分布性、开放性和平台无关性为基础的，这是 Internet 与生俱有的属性。随着互联网应用的发展，又引入了诸如 RPC/COM/CORBA 等技术，但这些技术在实际应用中，又存在着很多不足和局限。它们的特定协议也难以通过防火墙，因而不适于在 Web 上应用开发。为了进一步开发基于 Web 的应用，相继出现了 Sun 公司的 Sun ONE（Open Net Environment 开放网络环境）和 Microsoft 公司的.NET 两大 Web 服务技术体系。其中，Sun ONE 以 Java 技术为核心，更接近或者满足于互联网在智能化 Web 服务上对分布性、开放性和平台无关性的要求，同时其在健壮性、安全性、组件化等方面也更为成熟稳定，获得了众多 IT 厂商和产品的支持，是目前惟一在市场上得到了广泛应用的技术体系。

Sun ONE 体系结构以 Java 语言为核心，包括 J2SE/J2EE/J2ME，并基于一系列开放和流行标准、技术及协议。要特别指出的是，Sun ONE 体系结构本身作为开放式体系结构，在得到 IBM/BEA/Oracle/Sybase 等这些 IT 巨擘支持的同时，更得到了互联网上 Open Source 社区的青睐。我们很容易地从网上免费获得和使用包括 Java 集成开发环境、Java 数据库，甚至是中间件（Application Server）服务器等产品，以及它们的源代码。这对于加速国内中小企业的信息化建设和自有知识产权产品开发、提高企业应用和软件行业的整体水平，无疑是一次难得的机会。

综观国内的技术发展，广大的 Java 程序开发人员及正在转向 Java 体系进行开发的技术人员虽然已面临这一令人激动和鼓舞的转型期，却苦于没有足够的相关资料和文献，尤其对国内的最新 Java 技术动态和技术现状知之甚少，而图书市场上 Java 的书籍尽管汗牛充栋，但精品罕见，能反映出 J2EE 及 Sun ONE 的框架全貌的书籍更是奇缺。

电子工业出版社计算机图书研发部为进一步推动国内 Java 技术的应用与发展，不失时机地推出了《开发专家之 Sun ONE》系列丛书。

本套丛书以 Sun ONE 整体架构为基础，全面体现了 Sun ONE 的技术核心——Java 的应用开发。丛书从各个角度深入 Java 应用开发的各个层面，涵盖了 Java 技术的所有重要思想和实践，体现了最新的 Java 技术进展和动态，大幅度提升读者的理论和应用水平。同时，丛书重点突出实用性。书中引入了大量的行业应用范例，使读者不仅能快速掌握开发技能，而且对于开发者进行综合系统分析也有所裨益。

关于本书

自从 J2EE 平台诞生，它就专注于企业级 Java 市场。这其中，出现了各种不同的新技

术方向、新基础框架，以及新技术社区。Open Source 社区在推动 J2EE 发展方面起了不可替代的作用。开发者都知道，该社区开发的产品紧跟最新的技术规范、针对 J2EE 规范本身，以及基于 J2EE 的应用中的缺陷能够快速给出最佳解决方案。最重要的一点是 Open Source 是开放的，相应产品的升级不需要应用开发者维护和升级。新的观点、技术框架在经过业界的使用和磨炼后，Open Source 产品便可以为企业级 Java 应用提供基础支撑作用。

另外，开发者通过使用 Open Source 产品，可以获得最佳实践和业界的丰富经验，也不需要锁定在商用产品上，从而为开发者提供了最有保障的机制。

Spring 作为解决开发者和 J2EE 平台技术间的使用隔阂起到了很重要的作用。直接借助于 J2EE API 开发应用，使得开发者将大部分精力花费在同资源交互的处理上，而对实际应用系统的开发却不多，这使得开发 J2EE 效率低下的问题暴露无疑。开发者都知道，J2EE 平台技术解决了传统两层架构中的问题，即不存在平台服务。但是，这种服务的使用模式对于开发者而言，现有的 J2EE API 的复杂性使得开发者很为难。因此，Spring 试图消除上述隔阂。作为架构级的框架，Spring 是很出色的。无论开发者是否打算使用 Spring 开发 Java、J2EE 应用，研究 Spring 都是有必要的。

其一，Spring 将现有 J2EE 开发中存在的问题都摆出来了，这给开发者指引了正确的方向，从而避免犯错误。

其二，Spring 将这些问题或者部分解决，或者一直在解决中。借助于 Spring，开发者能够快速掌握使用 J2EE 技术的最佳实践。至少，开发者不会再误用 J2EE 技术了。

其三，通过研究 Spring，能够看到整个 Java/J2EE 的发展趋势，处于开发中的 J2EE 5.0 印证了这一点。因此，在某种程度上，Spring 是未来使用 J2EE 技术的缩影。

其四，Spring 将各种业界广泛使用的技术集成进来，以便于开发者使用。Spring 将各种 Open Source 和非 Open Source 技术、框架集成在一起。这也体现出 Spring 的通用性、易用性。

借助于 Spring 架构企业级 Java 应用，确实能够为开发者提供有益的帮助。随着对 Spring 使用的日益深入，开发者就越会体会到 Spring 为开发者架构、开发 Web 应用考虑得多么周到、细致。Spring 本身就是为开发者而来的，比如 Spring IoC、Spring AOP、Spring J2EE 服务抽象、Spring Acegi 等这些都是为开发者准备的开发利器。

优美的架构（比如，合理进行分层、各层又是有机的整体）、一致的开发模式（比如，借助于为 J2EE 服务提供的模板方法操作资源）、富有生机的 Spring 开发者社区（比如，新版本的发布周期非常短），这就是 Spring。

我们的联系方式如下：
咨询电话：（010）68134545　　68131648
电子邮件：support@fecit.com.cn
服务网址：http://www.fecit.com.cn　　http://www.fecit.net
通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

目　录

第一部分　Spring 架构分析

第二部分　Spring 应用开发

第一部分 Spring 架构分析

本部分内容将带领开发者逐渐进入到架构级的 Spring 框架中。如果您还未接触过 Spring，则务必阅读第 1 章内容。因为，它会带您进入 Spring 殿堂。如果需要对 Spring 框架的安装和构建过程进行系统性研究，则不要错过第 2 章内容。第 3、4 章分别对 Spring IoC 容器、Spring AOP 实现进行了展示，它们是 Spring 框架的核心。在有了这些基础知识后，我们将进一步深入到 Spring 的架构中，即第 5 章内容。上述所有内容都是围绕 Spring 的架构展开的，而且本书的后续内容会持续深入到 Spring 的架构中。

从 Spring 的源代码到 Spring 的安装、构建过程，再到 Spring 提供的 IoC 和 AOP，这些内容都在此进行了深入的阐释。

当然，开发者在阅读第二、三部分时，可以翻阅这部分内容，从而能够加深对 Spring 架构的掌握程度。

第 1 章　Spring 启程

本章内容将给出若干个基于 Spring 开发的相关实例，使得开发者能够了解 Spring 基础知识，从而为深入了解 Spring 其他内容奠定基础。

全书实例代码的安装请开发者参考“附录 A　实例代码安装”。

1.1　背 景 知 识

作为 Open Source 框架，Spring 是很令人振奋的。Spring 从一开始，直到现在，而且包括以后，将持续获得更多开发者的青睐，因为它是优秀的、是为开发者设计的实用框架。现存的 Open Source 框架很多，但是属于 Java/J2EE 架构级的优秀框架不多，包括现在流行的 Struts、Tapestry 等[1]都不是。研究 Spring 的过程，是很有意义的一件事情，因为它将业界的各种开发经验（包括 Open Source 的和非 Open Source 领域的）融入到其中。

Spring 提供的控制反转（Inversion of Control，IoC[2]）和面向方面编程（Aspect-Oriented Programming，AOP[3]）插件式架构降低了应用组件之间的依赖性。借助于 XML 定义文件，开发者能够在运行时连接不同的应用组件。这对于单元测试特别有用，特别是那些需要针对不同客户实施不同配置的应用而言。

目前，存在的依赖注入类型有 3 种：基于设值（setter-based）方法、基于构建器（constructor-based）以及基于接口（interface-based）注入。Spring IoC 支持前两种，即借助于 Spring 开发者可以通过构建器，或者设值方法创建对象，并对对象的状态进行管理。

其中，依赖注入是 Spring 框架的基础。Spring 在基于依赖注入的基础之上，同时还提供了其他大量的功能，比如 Spring MVC 框架、事务管理框架、DAO 支持、支持主流的 O/R Mapping 工具、支持各种标准 J2EE 组件技术的集成（JMS/JavaMail/EJB/Web 服务等）、集成各种视图（Web 视图和非 Web 视图）技术。这也是使用 Spring IoC 容器优于其他 IoC 容器的理由。

1.2　运行 Spring 实例应用

为阐述清楚 Spring IoC 框架帮助开发者完成了哪些工作，本章给出的运行实例起到了很重要的作用，即开发者通过代码能够很清楚地看到 Spring 的具体内容。

---

[1] 开发者可以参考网站：http://www.waferproject.org，即 Web 应用框架项目。

[2] 控制反转（IoC），又称之为依赖注入（Dependency Injection，DI）。本书约定：它们的含义一致。至于有关 IoC 和 DI 的更详细解释，开发者可以参考网址：http://www.martinfowler.com/articles/injection.html。同时，本书第 3 章将深入研究 Spring IoC。

[3] 本书第 4 章将深入研究 Spring AOP。

1.2.1 实例 1：example1

example1（所有的源代码位于 example1 项目中）并没有使用 Spring 框架。本章将通过一系列重构步骤，并最终借助于 Spring 提供的 IoC 框架来实现实例应用。接下来，给出大体步骤（详细内容，请参考 example1 项目）。

首先，开发者需要实现 FileHelloStr.java 类。

```
package com.openv.spring;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

import java.util.Properties;

/**
 * 基于文件形式，读取 HelloWorld 所需的字符串
 *
 * @author luoshifei
 */
public class FileHelloStr {
   protected static final Log log = LogFactory.getLog(FileHelloStr.class);
   private String propfilename;

   public FileHelloStr(String propfilename) {
      this.propfilename = propfilename;
   }

   public String getContent() {
      String helloworld = "";

      try {
         Properties properties = new Properties();
         InputStream is = getClass().getClassLoader().getResourceAsStream
          (propfilename);
         properties.load(is);
         is.close();
         helloworld = properties.getProperty("helloworld");
      } catch (FileNotFoundException ex) {
         log.error(ex);
      } catch (IOException ex) {
         log.error(ex);
```

```
        }

        return helloworld;
    }
}
```

上述类实现了对传入的、名为 propfilename 的属性文件的读取工作，并能够打印出相关异常信息。

其次，开发者还需要实现 HelloWorld.java 类。

```
package com.openv.spring;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;


/**
 * 获得HelloWorld字符串
 *
 * @author luoshifei
 */
public class HelloWorld {
    protected static final Log log = LogFactory.getLog(HelloWorld.class);

    public String getContent() {
        FileHelloStr fhStr = new FileHelloStr("helloworld.properties");
        String hellworld = fhStr.getContent();

        return hellworld;
    }
}
```

上述类调用了 FileHelloStr。

第三步，开发者需要实现 HelloWorldClient.java 类，从而实现对 HelloWorld 类的调用。

```
package com.openv.spring;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;


/**
 * HelloWorld客户应用
 *
 * @author luoshifei
 */
public class HelloWorldClient {
    protected static final Log log = LogFactory.getLog(HelloWorldClient.class);
```

```
    public static void main(String[] args) {
        HelloWorld hw = new HelloWorld();
        log.info(hw.getContent());
    }
}
```

第四步，开发者还需要编写 HelloWorld.properties 文件，其内容如下。

```
helloworld = "Hello World!"
```

最后，运行 example1 应用。通过两种方式可以运行 HelloWorldClient 客户应用。其一，通过 Eclipse IDE；其二，通过 Ant build.xml 文件的 run 任务。实例输出信息如下：

```
Buildfile: D:\workspace\example1\build.xml
compile:
run:
    [java] 2004-12-12 14:44:43 com.openv.spring.HelloWorldClient main
    [java] 信息: "Hello World!"
BUILD SUCCESSFUL
Total time: 2 seconds
```

其中，example1 使用到的 Ant build.xml 文件内容如下。

```
<?xml version="1.0"?>
<project name="example1" default="run" basedir=".">
    <path id="lib">

        <fileset dir="../examplelib/jakarta-commons">
            <include name="commons-logging.jar"/>
        </fileset>

    </path>

    <target name="run" depends="compile" description="Run HelloWorldClient">

        <java classname="com.openv.spring.HelloWorldClient" fork="yes">
            <classpath refid="lib"/>
            <classpath path="classes"/>
        </java>

    </target>

    <target name="compile">

        <mkdir dir="classes"/>

        <javac destdir="classes" source="1.5" target="1.5"
                deprecation="false" optimize="false" failonerror="true">
            <src path="src"/>
            <classpath refid="lib"/>
        </javac>
```