[第一部分 spring 核心 1](#_Toc2820)

[1、 spring起航 1](#_Toc2815)

[1.3 spring 概览 1](#_Toc3955)

[1.3.1 spring 模块 1](#_Toc18652)

1.3 spring概览

正如你所见，spring框架致力于通过使用DI.AOP和减少引用来简化企业级java应用的开发。虽然这些都是spring做的，使用它还是非常值得。但是，比之前看到的，spring还做了更多。

在spring框架内部，你可以找到几种spring简化java开发的方式。但是在spring的外面，是一个建立在核心框架，继承spring的某些模块的一个极好的项目生态系统，比如 web服务，rest，mobile,和nosql。。。等等。

让我们先来拆分spring核心框架，来看看它采用什么方式实现的。接着我们会回顾spring包里面的其他模块，来扩宽我们的视野。

1.3.1 spring 模块

当你下载spring的配置，分析它的文件结构时，可以发现几个jar文件。

自从spring4.0开始，spring有20个清晰的模块，每个模块有三种jar文件。

全部的jar文件库的目录如图1.6所示。

这些模块可以按功能分成6个类别，如图1.7所示。

把这些模块当成整体使用时，它可以为你提供开发企业级应用需要的一切。

但是如果你不想你的应用全部是用spring来实现，你可以自由的选择spring中合适你的模块，然后，在spring不适合你的应用的时候，去寻找其他的选择。Spring甚至为你提供了对几个流行框架和库的集成，让你不必自己去写这些。

下面来看看spring的每个模块，为了看看每个部分如何配合整体的spring，一次只看一个。

**Spring核心容器：**

Spring的核心是一个容器，用来管理spring应用中的bean通过什么方式来创建，配置，和管理的。在这个模块中，spring bean 工厂作为spring的一部分，提供了依赖反转功能。

你会发现有几个基于这个spring bean 工厂的spring上下文实现，它们提供了一些不同的方式来配置spring。

除了bean工厂和spring上下文，这个模块也提供了一些企业级服务，比如email，jndi访问，ejb集成，和调度。

所有的spring模块都是建立在这个核心容器之上。这个核心容器会贯穿本书始末，我们会在第二章开始深入研究spring的依赖反转。

**springAop模块：**

Spring在它的aop模块中为面向切面编程提供了丰富的支持。这个模块为你在自己的spring应用程序中开发你自己的切面提供了基础。虽然，依赖反转和aop 的支持都能减小应用中对象间的耦合度。但是在aop中，应用程序范围的关注点（比如事务和安全）从对象本身解耦到它们所依赖的切面服务。

我们将会在第四章深入研究spring中的aop服务。

**数据访问和集成：**

在用jdbc时，经常会导致大量的模板代码，像得到一个连接，创建声明，处理结果集，最后在关闭连接。Spring 的jdbc和Dao模块把模板代码抽象出来，因此你可以保持你的数据库代码简洁，并且可以防止来自关闭数据库资源导致的问题。这个模块已经建立了一个语义分明的异常体系，它基于几个数据库服务给予的错误码信息。再也不用尝试破解神秘，专有的sql错误信息了！

对于更喜欢使用对象映射工具而不是直接使用jdbc的人，spring提供了ORM模块。

Spring的ORM支持建立在DAO支持的基础上，提供了一种便利的方式建立与几种orm解决方案对应的dao层。Spring没有尝试实现它自己的orm解决方案，而是为流行的ORM框架提供接口，比如hibernate，jpa，jdo，和ibatis sql mapping。Spring的事务管理可以像支持jdbc一样支持这些ORM框架。

当我们学习到第十章spring数据访问时，你就会看到spring 基于jdbc的抽象模板是怎样极大的简化JDBC的代码。

这个模块还包括了java消息服务（JMS）的一个spring抽象，在与其他应用通讯时，它提供了异步消息集成。并且，自从spring3.0开始，这个模块包含了对象映射XML的特性，它本来是spring web 服务项目里的。

此外，为了对应用中的对象提供事务管理服务，本模块使用了spring的aop模块。

**WEB和远程调用：**

MVC模式是一个被普遍接受的方式用来建立web应用，例如用户界面和应用逻辑分离。

JAVA从来不缺少MVC框架，像Apache Struts,JSF,Web Work,Tapestry 已经成为了最流行MVC框架的选择。

虽然spring已经集成了几种流行的MVC框架，但它的WEB和远程调用是功能很强大的MVC框架，它提升了spring在应用中web层的减小耦合的技术。我们会在5-7章节看到springMVC框架。

此外，对于面向用户的WEB 应用程序，这个模块也提供了几个远程调用的选择，用在互相通讯的应用程序中。Spring的远程调用能力包括 RMI,Hessian,Burlap,JAX-WS，和Spring的自己http调用。Spring也对发布和使用REST风格的API提供一流的支持。

在第十五章，我们将会检验spring的远程调用。并且你会在第十六章学回创建和使用REST风格的API。

**基础设施模块**

Spring的基础设施模块包含对JVM增加代理的支持。特别地，它对tomcat提供了一个织入的代理，随着他们被类加载器加载，转化成class文件。

如果它听起来很难理解，不要担心。这个基础设施模块提供的功能，很少在日常中被用到，并且我们不会在这本书里涉及到它。

**测试**

认识到开发者自己写测试的重要性，spring提供了一个模块专门用来测试spring应用。

在这个模块中，你会发现一个由mock对象实现组成的集合，mock对象是用来写单元测试代码，这些代码与JNDI, Servlet和组件一起工作的。针对集成水平的测试，这个模块对在spring应用上下文中加载bean的集合和在上下文中和bean工作提供了支持。

总览本书，许多例子会通过spring测试工厂提供的测试来驱动。

**1.3.2 spirng 生态**

走近spring时，会发现它远比我们看到的多。事实上，spring远多于我们下载的spring框架。如果你停止在spring核心框架，你会错过更大的spring生态提供的丰富的潜藏的能力。

Spring生态包含了几个框架和库，他们建立在spring核心框架和彼此的基础上。完整的spring生态为java开发的几乎每一个方面带来了spring程序设计模型。

完整的讲述spring生态的知识将会花费几本书，这已经超出了本书的范围。但是我们会看一些spring生态的元素，那些超出了spring核心框架的特性。

**springWEB流**

springWEB流建立在spring核心MVC框架的基础上，它提供了对建立会话，流基础的应用的支持，会指导用户达到目标。我们会在第八章讨论更多关于springWEB流的内容。

**Spring WEB 服务**

虽然spring核心框架提供了声明式发布的spring bean 作为 web 服务，但是这些服务是建立在一个可以说比最新协议模型低级一些的架构之上。服务的协议已经被bean的接口决定了。Spring web 服务提供了最新的web 服务 模型，它的服务实现专门为了满足服务协议而写的。

我们不会讨论关于 spring web 服务 在这本书里。

**Spring 安全**

安全是一个应用关键的方面。Spring安全被spring aop实现，它提供了一种声明式的安全机制，应用在以spring为基础的应用中。你会在第九章看到怎样给应用的web层增加spring的安全机制。我们会在第十四章继续讨论spring安全来检测如何保护方法调用。若要更近一步的了解，请访问spring安全的主页。http://project.spring.io/spring-security/.

**Spring 集成**

许多企业级应用必须与其它的企业级应用交互。Spring集成提供了一些常见集成模式的实现，它们使用spring的声明式风格。

**Spring 批处理**

当需要进行大量数据的处理操作，没有什么比批处理更有效的了。如果你正在开发一个批处理的应用，那么你可以使用spring的批处理，通过使用spring健康面向对象的开发模型来做它。

Spring的批处理超出了本书的范畴，，，，，，

**Spring DATA**

在spring应用中，Spring data 会使与不同种类的数据库工作变得容易。虽然很多年来关系型数据库在企业应用中已经非常普遍，但是现代应用正在认识到，不是所有的数据都最适合用表中的行和属性来提供服务。一种新的数据库，一般是指NoSQL数据库，提供了与数据工作新方式，它比传统的关系型数据库更适合。

不管你是否使用正在使用一个文件数据库，像 MongoDB,图表数据库像Neo4j,或者即使是一个传统的关系型数据库。Spring为持久化数据提供了一个简明的编程模型。它包括针对许多的数据库类型的一种为你自动创建repository 实现的机制。

我们会在第十一章看到使用spring data 简化 jpa的 开发，然后在第十二章基础讨论一点关于nosql数据库的知识。