

Spring框架

Spring框架是由于软件开发的复杂性而创建的。Spring使用的是基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅仅限于服务器端的开发。从简单性、可测试性和松耦合性角度而言，绝大部分Java应用都可以从Spring中受益。

◆**目的：解决企业应用开发的复杂性**

◆**功能：使用基本的JavaBean代替EJB，并提供了更多的企业应用功能**

◆**范围：任何Java应用**

Spring是一个轻量级控制反转(IOC)和面向切面(AOP)的容器框架。

起源

要谈Spring的历史，就要先谈J2EE。J2EE应用程序的广泛实现是在1999年和2000年开始的，它的出现带来了诸如事务管理之类的核心中间层概念的标准化，但是在实践中并没有获得绝对的成功，因为开发效率，开发难度和实际的性能都令人失望。

曾经使用过EJB开发JAVA EE应用的人，一定知道，在EJB开始的学习和应用非常的艰苦，很多东西都不能一下子就很容易的理解。EJB要严格地实现各种不同类型的接口，类似的或者重复的代码大量存在。而配置也是复杂和单调，同样使用JNDI进行对象查找的代码也是单调而枯燥。虽然有一些开发工作随着xdoclet的出现，而有所缓解，但是学习EJB的高昂代价，和极低的开发效率，极高的资源消耗，都造成了EJB的使用困难。而Spring出现的初衷就是为了解决类似的这些问题。

Spring的一个最大的目的就是使JAVA EE开发更加容易。同时，Spring之所以与Struts、Hibernate等单层框架不同，是因为Spring致力于提供一个以统一的、高效的方式构造整个应用，并且可以将单层框架以最佳的组合揉和在一起建立一个连贯的体系。可以说Spring是一个提供了更完善开发环境的一个框架，可以为POJO(Plain Ordinary Java Object)对象提供企业级的服务。

Spring的形成，最初来自Rod Jahnson所著的一本很有影响力的书籍《Expert One-on-One J2EE Design and Development》，就是在这本书中第一次出现了Spring的一些核心思想，该书出版于2002年。另外一本书《Expert One-on-One J2EE Development without EJB》，更进一步阐述了在不使用EJB开发JAVA EE企业级应用的一些设计思想和具体的做法。有时间了可以详细的研读一下。

Spring的初衷：

- 1、JAVA EE开发应该更加简单。
- 2、使用接口而不是使用类，是更好的编程习惯。Spring将使用接口的复杂度几乎降低到了零。

- 3、为JavaBean提供了一个更好的应用配置框架。
- 4、更多地强调[面向对象](#)的设计，而不是现行的技术如JAVA EE。
- 5、尽量减少不必要的异常捕捉。
- 6、使应用程序更加容易测试。

Spring的目标：

- 1、可以令人方便愉快的使用Spring。
- 2、应用程序代码并不依赖于Spring APIs。
- 3、Spring不和现有的解决方案竞争，而是致力于将它们融合在一起。

Spring的基本组成：

- 1、最完善的轻量级核心框架。
- 2、通用的事务管理抽象层。
- 3、JDBC抽象层。
- 4、集成了Toplink, Hibernate, JDO, and iBATIS SQL Maps。
- 5、AOP功能。
- 6、灵活的MVC Web应用框架。

优点

- ◆JAVA EE应该更加容易使用。
- ◆面向对象的设计比任何实现技术（比如JAVA EE）都重要。
- ◆面向接口编程，而不是针对类编程。Spring将使用接口的复杂度降低到零。（面向接口编程有哪些复杂度？）
- ◆代码应该易于测试。Spring框架会帮助你，使代码的测试更加简单。
- ◆JavaBean提供了应用程序配置的最好方法。
- ◆在Java中，已检查异常（Checked exception）被过度使用。框架不应该迫使你捕获不能恢复的异常。

简介

Spring：

Spring是一个开源框架，它由[Rod Johnson](#)创建。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。

Spring是一个轻量级的控制反转(IOC)和面向切面(AOP)的容器框架。

轻量——从大小与开销两方面而言Spring都是轻量的。完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布。并且Spring所需的处理开销也是微不足道的。此外，Spring是非侵入式的：典型地，Spring应用中的对象不依赖于Spring的特定类。

控制反转——Spring通过一种称作控制反转（IoC）的技术促进了松耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为IoC与JNDI相反——不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

面向切面——Spring提供了[面向切面编程](#)的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和事务（transaction）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

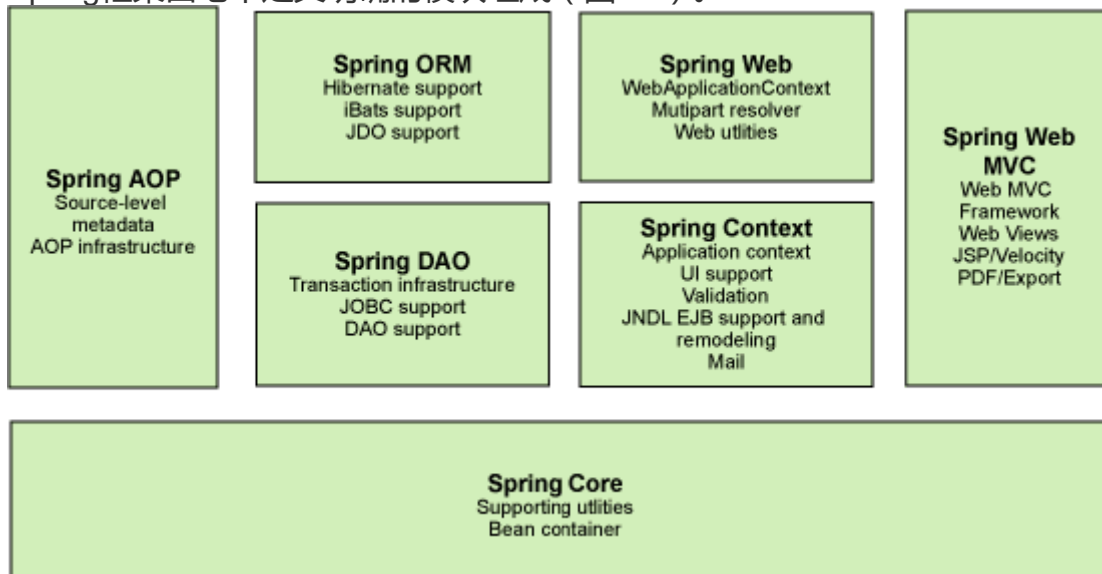
容器——Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个bean如何被创建——基于一个可配置原型（prototype），你的bean可以创建一个单独的实例或者每次需要时都生成一个新的实例——以及它们是如何相互关联的。然而，Spring不应该被混同于传统的重量级的EJB容器，它们经常是庞大与笨重的，难以使用。

框架——Spring可以将简单的组件配置、组合成为复杂的应用。在Spring中，应用对象被声明式地组合，典型地是在一个XML文件里。Spring也提供了很多基础功能（事务管理、持久化框架集成等等），将应用逻辑的开发留给了你。

所有Spring的这些特征使你能够编写更干净、更可管理、并且更易于测试的代码。它们也为Spring中的各种模块提供了基础支持。

模块

Spring框架由七个定义明确的模块组成（图1.1）。



（Spring框架概览图1.1）

如果作为一个整体，这些模块为你提供了开发企业应用所需的一切。但你不必将应用完全基于Spring框架。你可以自由地挑选适合你的应用的模块而忽略其余的模块。就像你所看到的，所有的Spring模块都是在核心容器之上构建的。容器定义了Bean是如何创建、配置和管理的——更多的Spring细节。当你配置你的应用时，你会潜在地使用这些

类。但是作为一名开发者，你最可能对影响容器所提供的服务的其它模块感兴趣。这些模块将会为你提供用于构建应用服务的框架，例如AOP和持久性。

核心容器

这是Spring框架最基础的部分，它提供了依赖注入（DependencyInjection）特征来实现容器对Bean的管理。这里最基本的概念是BeanFactory，它是任何Spring应用的核心。BeanFactory是工厂模式的一个实现，它使用IoC将应用配置和依赖说明从实际的应用代码中分离出来。

应用上下文（Context）模块

核心模块的BeanFactory使Spring成为一个容器，而上下文模块使它成为一个框架。这个模块扩展了BeanFactory的概念，增加了对国际化（I18N）消息、事件传播以及验证的支持。

另外，这个模块提供了许多企业服务，例如电子邮件、JNDI访问、EJB集成、远程以及时序调度（scheduling）服务。也包括了对模版框架例如Velocity和FreeMarker集成的支持。

Spring的AOP模块

Spring在它的AOP模块中提供了对面向切面编程的丰富支持。这个模块是在Spring应用中实现切面编程的基础。为了确保Spring与其它AOP框架的互用性，Spring的AOP支持基于AOP联盟定义的API。AOP联盟是一个开源项目，它的目标是通过定义一组共同的接口和组件来促进AOP的使用以及不同的AOP实现之间的互用性。通过访问他们的站点，你可以找到关于AOP联盟的更多内容。

Spring的AOP模块也将元数据编程引入了Spring。使用Spring的元数据支持，你可以为你的源代码增加注释，指示Spring在何处以及如何应用切面函数。

JDBC抽象和DAO模块

使用JDBC经常导致大量的重复代码，取得连接、创建语句、处理结果集，然后关闭连接。Spring的JDBC和DAO模块抽取了这些重复代码，因此你可以保持你的数据库访问代码干净简洁，并且可以防止因关闭数据库资源失败而引起的问题。

这个模块还在几种数据库服务器给出的错误消息之上建立了一个有意义的异常层。使你不用再试图破译神秘的私有的SQL错误消息！

另外，这个模块还使用了Spring的AOP模块为Spring应用中的对象提供了事务管理服务。

对象/关系映射集成模块

对那些更喜欢使用对象/关系映射工具而不是直接使用JDBC的人，Spring提供了ORM模块。Spring并不试图实现它自己的ORM解决方案，而是为几种流行的ORM框架提供了集成方案，包括Hibernate、JDO和iBATIS SQL映射。Spring的事务管理支持这些ORM框架中的每一个也包括JDBC。

Spring的Web模块

Web上下文模块建立于应用上下文模块之上，提供了一个适合于Web应用的上下文。另外，这个模块还提供了一些面向服务支持。例如：实现文件上传的multipart请求，它也提供了Spring和其它Web框架的集成，比如Struts、WebWork。

Spring的MVC框架

Spring为构建Web应用提供了一个功能全面的MVC框架。虽然Spring可以很容易地与其它MVC框架集成，例如Struts，但Spring的MVC框架使用IoC对控制逻辑和业务对象提供了完全的分离。

它也允许你声明性地将请求参数绑定到你的业务对象中，此外，Spring的MVC框架还可以利用Spring的任何其它服务，例如国际化信息与验证。

Spring框架Web页面乱码问题

在做java Web 项目时，乱码问题时常都会出现，解决方法也不尽相同，有简单也有复杂的；如果加入了Spring框架之后就不一样了，可以采用Spring框架自带的过滤器CharacterEncodingFilter，这样可以大大减轻了我们的工作量，即简单方便又容易理解，配置方式如下：在web.xml文件中filter的位置加上如下内容：

```
1      <filter>
2          <filter-name>encodingFilter</filter-name>
3          <filter-class>
4              org.springframework.web.filter.CharacterEncodingFilter
5          </filter-class>
6          <init-param>
7              <param-name>encoding</param-name>
8              <param-value>UTF-8</param-value>
9          </init-param>
10         <init-param>
11             <param-name>forceEncoding</param-name>
12             <param-value>true</param-value>
13         </init-param>
14     </filter>
15     <filter-mapping>
16         <filter-name>encodingFilter</filter-name>
17         <url-pattern>*</url-pattern>
18     </filter-mapping>
```