



链滴

翻译：Spring 框架概述

作者：[CoolReid](#)

原文链接：<https://ld246.com/article/1532418008147>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

翻译：Spring框架概述

原文链接：<https://docs.spring.io/spring/docs/5.0.x/spring-framework-reference/overview.htm#overview>

版本：5.0.7.RELEASE

Spring可以轻松创建Java企业应用程序。它提供了在企业环境中支持Java语言所需的一切，并支持Groovy和Kotlin作为JVM上的替代语言，而且可根据应用程序的需求灵活地创建多种体系结构。从Spring Framework 5.0开始，Spring需要JDK 8+ (Java SE 8+)，并且已经为JDK 9提供了开箱即用的支持。

Spring支持广泛的应用场景。在大型企业中，应用程序通常存在很长时间，并且必须在升级周期超出开发人员控制的JDK和应用程序服务器上运行。其他人可能会将嵌入式服务器作为单个jar运行，可能在环境中运行。而另一些可能是不需要服务器的独立应用程序（如批处理或集成工作负载）。

Spring是开源的。它拥有一个庞大而活跃的社区，可根据各种实际用例提供持续的反馈。这有助于Spring在很长一段时期内成功发展。

1. “Spring” 的含义

术语“Spring”的含义在不同的上下文语境中不尽相同。大部分情况下它指代spring框架本身，也是开始的含义。后来，出现了很多其他基于Spring框架构建的项目。但是大部分情况下当人们说spring时候，他们的意思是整个spring项目体系。本篇说明将重点放在spring框架本身。

spring框架分成不同模块，应用只需选择需要的模块。核心是core容器模块，包括一个配置模块和依赖注入机制。除此之外，spring框架为不同的应用架构提供基础支持，包括消息传递，事务数据和久化以及Web。它还包含基于Servlet的Spring MVC Web框架以及Spring WebFlux反应式Web框。

模块说明：spring框架的jar包部署到JDK9的组件路径("Jigsaw")。为了在启用Jigsaw的应用程序使用，Spring Framework 5 jar包带有“Automatic-Module-Name”词典，它定义了语言层面独立于jar名称架构的稳定的名称（即用“-”代替“.”，比如"spring-core" and "spring-context"）。此，Spring框架在JDK8和JDK9都能正常工作。

2.Spring和Spring框架的历史

Spring于2003年应运而生，以应对早期J2EE规范的复杂性。虽然有人认为Java EE和Spring竞争激烈但Spring实际上是对JavaEE的补充。Spring编程模型不支持Java EE平台规范；相反，它集成了从EE umbrella精心挑选的个别规格：

- Servlet API (JSR 340)
- WebSocket API (JSR 356)
- 并发实用程序 (JSR 236)
- JSON绑定API (JSR 367)
- Bean验证 (JSR 303)
- JPA (JSR 338)
- JMS (JSR 914)
- 以及必要时用于事务协调的JTA/JCA设置。

Spring框架同时也支持依赖注入(JSR330)和公共声明(JSR250)规范，应用程序开发人员可以选择使用这些规范，而不是Spring框架提供的Spring特定机制。

从Spring Framework 5.0开始，Spring至少需要Java EE7级别（例如Servlet3.1+，JPA 2.1+），同时在Java EE 8级别上提供运行时的开箱即用的新API集成（例如Servlet 4.0，JSON绑定API）在运行时到。这使Spring完全兼容Tomcat 8和9，WebSphere 9和JBoss EAP 7。

随着时间的推移，Java EE在应用程序开发中的作用也在不断发展。在Java EE和Spring的早期，创建应用程序以部署到应用程序服务器。今天，用Spring Boot应用程序以devops和易于云化的方式创建，时嵌入了Servlet容器并且可以轻松更改。从Spring Framework 5开始，WebFlux应用程序甚至不直接使用Servlet API，而是可以在不是Servlet容器的服务器（如Netty）上运行。

Spring不断创新并不断发展。除了Spring框架之外，还有其他项目，比如Spring Boot，Spring Security，Spring Data，Spring Cloud，Spring Batch等等。重要的是要记住每个项目都有自己的源代码库，issue 跟踪和发布节奏。有关Spring项目的完整列表，请参阅spring.io/projects。

3.设计哲学

当你学习一个框架的时候，知道它做了什么和重要，但更重要的了解他遵从的原则。下面是spring设计时的一些主要原则：

- 在各个层面提供选择。Spring允许您尽可能迟地推迟设计决策。例如，您可以通过配置切换持久化序，而无需更改代码。许多其他基础架构以及与第三方API的集成也是如此。
- 包容各种观点。spring拥抱灵活，并没有对如何完成事情持有自己的看法。它以不同的视角支持广的应用需求。
- 保持强大的后向兼容性。Spring的演变经过精心设计，可以在版本之间进行一些重大改变。Spring持精心挑选的JDK版本和第三方库，以便于维护依赖于Spring的应用程序和库。
- 关心API设计。Spring团队花费了大量的精力和时间来制作直观的API，并在多年里支持多种版本。

*为代码质量设定高标准。Spring Framework强调有意义，最新且准确的Javadoc。它是少数几个可声明干净的代码结构的程序之一，package之间没有循环依赖关系。

4.反馈和贡献

对于操作方法问题、诊断问题或调试问题，我们建议使用StackOverflow，我们有一个[问题页面](#)列出要使用的建议标签。如果您确信Spring框架存在问题或者想要推荐一个功能，请使用[JIRA issue tracker](#)。

如果你有一个解决方案或建议修复，你可以在[Github](#)上提交一个pull请求。但是请记住，除了最琐碎问题外，我们希望在issue tracker中提交，在那里进行讨论并留下记录以备将来参考。

有关更多详细信息，请参阅[贡献](#)顶级项目页面上的指南。

5.入门

如果您刚刚开始使用Spring，则可能需要创建一个基于Spring Boot的应用程序来开始使用Spring Framework。Spring Boot提供了一种快速（武断）的方式来创建一个生产就绪的基于Spring的应用程序。它基于Spring Framework，偏爱公共约定而不是配置，并且旨在让您尽快启动并运行。

您可以使用start.spring.io生成一个基本项目，或者按照[入门指南](#)之一，例如 [构建RESTful Web服务](#)。除了易于消化外，这些指南非常专注于业务，其中大部分都基于Spring Boot。他们还涵盖了您

解决特定问题时可能需要考虑的Spring组合中的其他项目。