## 概述：

历史，设计理念，反馈，入门。

Spring可以轻松创建Java企业应用程序。它提供了在企业环境中使用Java语言所需的一切，支持Groovy和Kotlin作为JVM上的替代语言，并可根据应用程序的需要灵活地创建多种体系结构。从Spring Framework 5.1开始，Spring需要JDK 8+（Java SE 8+），并为JDK 11 LTS提供开箱即用的支持。

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，并且不得不在升级周期超出开发人员控制范围的JDK和应用程序服务器上运行。其他人可能在嵌入服务器的情况下作为单个jar运行，可能在云环境中运行。还有一些可能是不需要服务器的独立应用程序（例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，可根据各种各样的实际用例提供持续的反馈。这有助于Spring在很长一段时间内成功发展。

## 1.“Spring”是什么意思？

术语“Spring”在不同的背景下意味着不同的东西。它可以用来引用Spring Framework项目本身，它就是一切开始的地方。随着时间的推移，其他Spring项目已经构建在Spring Framework之上。大多数情况下，当人们说“春天”时，他们就意味着整个项目家族。本参考文档侧重于基础：Spring Framework本身。

Spring框架分为几个模块。应用程序可以选择所需的模块。core container(核心容器)是这些模块的核心。包括配置模型和依赖注入机制。除此之外，Spring Framework还为不同的应用程序体系结构提供了基础支持，包括消息传递，事务数据和持久性以及Web。它还包括基于Servlet的Spring MVC Web框架，以及Spring WebFlux响应式Web框架。

关于模块的说明：Spring的框架jar允许部署到JDK 9的模块路径（“Jigsaw”）。为了在支持Jigsaw的应用程序中使用，Spring Framework 5 jar带有“Automatic-Module-Name”清单条目，它们定义了独立于jar工件的稳定语言级模块名称（“spring.core”，“spring.context”等）名称（jar使用相同的命名模式:用“-”代替“.“。例如“spring-core”和“spring-context”）。当然，Spring的框架jar在JDK 8和9+上的类路径上都能正常工作。

## 2. Spring的历史和Spring框架

Spring于2003年成立，是对早期[J2EE](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)规范复杂性的回应 。虽然有些人认为Java EE和Spring处于竞争中，但Spring实际上是对Java EE的补充。Spring编程模型不包含Java EE平台规范; 相反，它集成了EE结构中精心挑选的个别规范：

* Servlet API（[JSR 340](https://jcp.org/en/jsr/detail?id=340)）
* WebSocket API（[JSR 356](https://www.jcp.org/en/jsr/detail?id=356)）
* 并发实用程序（[JSR 236](https://www.jcp.org/en/jsr/detail?id=236)）
* JSON绑定API（[JSR 367](https://jcp.org/en/jsr/detail?id=367)）
* Bean验证（[JSR 303](https://jcp.org/en/jsr/detail?id=303)）
* JPA（[JSR 338](https://jcp.org/en/jsr/detail?id=338)）
* JMS（[JSR 914](https://jcp.org/en/jsr/detail?id=914)）
* 以及必要时用于事务协调的JTA / JCA设置。

说明：

Servlet：Servlet（Server Applet）是[Java](https://baike.baidu.com/item/Java/85979" \t "_blank) Servlet的简称，称为小服务程序或服务连接器，用Java编写的[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8/100571" \t "_blank)端程序，具有独立于平台和[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE/13020269)的特性，主要功能在于交互式地浏览和生成数据，生成动态[Web](https://baike.baidu.com/item/Web/150564)内容。

狭义的Servlet是指Java语言实现的一个接口，广义的Servlet是指任何实现了这个Servlet接口的类，一般情况下，人们将Servlet理解为后者。Servlet运行于支持Java的应用服务器中。从原理上讲，Servlet可以响应任何类型的请求，但绝大多数情况下Servlet只用来扩展基于HTTP协议的Web服务器。

WebSocket :**WebSockets** 是一种先进的技术。它可以在用户的浏览器和服务器之间打开交互式通信会话。使用此API，您可以向服务器发送消息并接收事件驱动的响应，而无需通过轮询服务器的方式以获得响应。

JPA:JPA是Java Persistence API的简称，中文名Java持久层API，是JDK 5.0注解或XML描述对象－关系表的映射关系，并将运行期的实体[对象持久化](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1%E6%8C%81%E4%B9%85%E5%8C%96/7316192)到数据库中。

JMS: JMS即[Java消息服务](https://baike.baidu.com/item/Java%E6%B6%88%E6%81%AF%E6%9C%8D%E5%8A%A1" \t "_blank)（Java Message Service）应用程序接口，是一个[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0)中关于面向[消息中间件](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E4%B8%AD%E9%97%B4%E4%BB%B6/5899771)（MOM）的[API](https://baike.baidu.com/item/API/10154" \t "_blank)，用于在两个应用程序之间，或[分布式系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F/4905336)中发送消息，进行异步通信。Java消息服务是一个与具体平台无关的API，绝大多数MOM提供商都对JMS提供支持。

Spring Framework还支持依赖注入（[JSR 330](https://www.jcp.org/en/jsr/detail?id=330)）和常用注解（[JSR 250](https://jcp.org/en/jsr/detail?id=250)）规范，应用程序开发人员可以选择使用这些规范，来代替Spring Framework提供的Spring特定机制。

从Spring Framework 5.0开始，Spring至少需要Java EE 7级别（例如Servlet 3.1 +，JPA 2.1+） - 同时在运行Java EE 8级别提供与新API的开箱即用集成（例如，Servlet 4.0，JSON绑定API）。这使Spring与Tomcat 8和9，WebSphere 9和JBoss EAP 7完全兼容。

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，创建了应用程序以部署到应用程序服务器。今天，在Spring Boot的帮助下，应用程序以开发运营和云支持的方式创建，Servlet容器嵌入变得微不足道。从Spring Framework 5开始，WebFlux应用程序甚至不直接使用Servlet API，并且可以在不是Servlet容器的服务器（例如Netty）上运行。

## 3.设计理念

当您了解框架时，重要的是不仅要知道它的作用，还要了解它遵循的原则。以下是Spring Framework的指导原则：

* 提供各个层面的选择。Spring允许您尽可能晚地推迟设计决策。例如，您可以通过配置切换持久性提供程序，而无需更改代码。许多其他基础架构问题以及与第三方API的集成也是如此。
* 适应不同的观点。Spring拥抱灵活性，它并不是固执己见的阐述事情应该怎么做。它以应用程序不同的视角支持广泛的应用需求。
* 保持强大的向后兼容性。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支持精心挑选的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团队花了很多心思和时间来制作直观的API，这些API在很多版本和多年中都有用。
* 为代码质量设定高标准。Spring Framework非常强调有意义的，最新的和准确的javadoc。它是极少数项目之一，可以声称干净的代码结构，包之间没有循环依赖。

## 4.反馈和贡献

对于操作方法问题或诊断或调试问题，我们建议使用StackOverflow，我们有一个[问题页面](https://spring.io/questions)列出了要使用的建议标签。如果您非常确定Spring Framework中存在问题或想要建议功能，请使用[GitHub Issues](https://github.com/spring-projects/spring-framework/issues)。

如果您有解决方案或建议的修复，您可以在[Github](https://github.com/spring-projects/spring-framework)上提交拉取请求 。但是，请记住，对于除了最微不足道的问题之外的所有问题，我们希望在问题跟踪器中提交一个标签，进行讨论并留下记录以供将来参考。

有关更多详细信息，请参阅“ [贡献](https://github.com/spring-projects/spring-framework/blob/master/CONTRIBUTING.md) ”顶级项目页面上的指南。

## 5.入门

如果您刚刚开始使用Spring，您可能希望通过创建基于[Spring Boot](https://projects.spring.io/spring-boot/)的应用程序来开始使用Spring Framework 。Spring Boot提供了一种快速（和固执己见）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，支持约定优于配置，旨在帮助您尽快启动和运行。

您可以使用[start.spring.io](https://start.spring.io/)生成基本项目，也可以按照[“入门”指南之一进行操作](https://spring.io/guides)，例如“ [入门构建RESTful Web服务”](https://spring.io/guides/gs/rest-service/)。除了更容易理解之外，这些指南非常注重任务，而且大多数都基于Spring Boot。它们还涵盖了Spring组合中您在解决特定问题时可能需要考虑的其他项目。