Spring Framework

# 概观

## 概述

Spring于2003年成立，是对早期J2EE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保护伞中精心挑选的个别规格：

* Servlet API
* 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 Framework还支持依赖注入和Common Annotations规范，应用程序开发人员可以选择使用这些规范，而不是Spring Framework提供的Spring特定机制。

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devops和云友好的方式创建，Servlet容器嵌入并且变得微不足道。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。

## 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，并不认为应该如何做。它以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团队投入了大量的思考和时间来制作直观的API，并且可以支持多个版本和多年。
* 为代码质量设定高标准。Spring框架强调有意义的，当前的和准确的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。

# 核心

## IoC容器

控制反转（IoC）也被称为依赖注入，指的是将创建对象（new）的工作交给IoC容器去完成，而在编程的时候只需要将需要的对象依赖

# 数据存储

# 单元测试

# Web Servlet

# Web Reactive

# Java EE（及相关）技术的集成。

# 语言支持