# 框架篇

# Spring Framework

Provides core support for dependency injection, transaction management, web apps, data access, messaging and more. 【官方介绍】

为**依赖注入**、**事务管理**、**web应用程序**、**数据访问**、**消息传递**等提供**核心**支持。

## 简述

### 概述

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform.

A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

### 结构

* Core technologies: dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, AOP.
* Testing: mock objects, TestContext framework, Spring MVC Test, WebTestClient.
* Data Access: transactions, DAO support, JDBC, ORM, Marshalling XML.
* Spring MVC and Spring WebFlux web frameworks.
* Integration: remoting, JMS, JCA, JMX, email, tasks, scheduling, cache.
* Languages: Kotlin, Groovy, dynamic languages.

### 最低配置环境

JDK 8+ for Spring Framework 5.x

JDK 6+ for Spring Framework 4.x

## 系统学习 【版本5.0.8】

### Spring Framework官方介绍

<https://docs.spring.io/spring/docs/5.0.8.RELEASE/spring-framework-reference/overview.html#overview-spring>

#### 设计理念

* Provide choice at every level. Spring lets you defer design decisions as late as possible. For example, you can switch persistence providers through configuration without changing your code. The same is true for many other infrastructure concerns and integration with third-party APIs.

**在每个层次提供选择**。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数据持久化提供者，而无需更改代码。对于许多其他基础设施关注点和与第三方API的集成也是如此。

* Accommodate diverse perspectives. Spring embraces flexibility and is not opinionated about how things should be done. It supports a wide range of application needs with different perspectives.

**容纳不同的观点**。Spring支持灵活性，对于应该如何做事不固执己见。它以不同的角度支持广泛的应用程序需求。

* Maintain strong backward compatibility. Spring’s evolution has been carefully managed to force few breaking changes between versions. Spring supports a carefully chosen range of JDK versions and third-party libraries to facilitate maintenance of applications and libraries that depend on Spring.

**保持强烈的向后兼容性**。Spring的进化被小心翼翼地控制在版本之间，几乎没有发生重大变化。Spring支持精心选择的JDK版本和第三方库，以方便维护依赖于Spring的应用程序和库。

* Care about API design. The Spring team puts a lot of thought and time into making APIs that are intuitive and that hold up across many versions and many years.

**关心API设计**。Spring团队投入了大量的精力和时间来制作直观的api，这些api可以在许多版本和多年的时间内使用。

* Set high standards for code quality. The Spring Framework puts a strong emphasis on meaningful, current, and accurate Javadoc. It is one of very few projects that can claim clean code structure with no circular dependencies between packages.

**为代码质量设定高标准**。Spring框架非常强调有意义的、最新的和准确的Javadoc。它是为数不多的项目之一，可以声明代码结构干净，而包之间没有循环依赖关系。

### Core

* IoC container
* Events
* Resources
* i18n
* Validation
* Data Binding
* Type Conversion
* SpEL
* AOP

#### IoC container 【控制反转容器】

##### 理解

IoC：the Inversion of Control 【控制反转】

DI：Dependency Injection 【依赖注入】

对象的创建和依赖（包括对象和各种资源）注入全部交由IoC容器完成，反转的含义是：IoC替代了“我们”，本由“我们”主动的去创建和注入这些对象，现在由IoC负责，“我们”变得被动。

##### IoC容器的实现

1. 基础包

org.springframework.beans 🡪 BeanFactory

org.springframework.context 🡪 ApplicationContext

1. 两个接口

BeanFactory：提供了配置框架和基本功能

ApplicationContext：继承BeanFactory，集成更多企业级特定功能，如AOP、国际化、资源处理、事件分发、应用程序层上下文WebApplicationContext

1. spring中的bean

beans：构成应用程序主干并由Spring IoC容器管理的所有对象

bean：一个实例化的、组装的并且被Spring IoC容器管理的对象

beans及依赖关系：反映在容器使用的配置元数据中