

# 第九章：Spring Bean 生命周期

小马哥 • mercyblitz

# Spring Bean 生命周期

---

1. Spring Bean 元信息配置阶段
2. Spring Bean 元信息解析阶段
3. Spring Bean 注册阶段
4. Spring BeanDefinition 合并阶段
5. Spring Bean Class 加载阶段
6. Spring Bean 实例化前阶段
7. Spring Bean 实例化阶段
8. Spring Bean 实例化后阶段
9. Spring Bean 属性赋值前阶段
10. Spring Bean Aware接口回调阶段



# Spring Bean 生命周期

---

- 11. Spring Bean 初始化前阶段
- 12. Spring Bean 初始化阶段
- 13. Spring Bean 初始化后阶段
- 14. Spring Bean 初始化完成阶段
- 15. Spring Bean 销毁前阶段
- 16. Spring Bean 销毁阶段
- 17. Spring Bean 垃圾收集
- 18. 面试题



# Spring Bean 元信息配置阶段

- BeanDefinition 配置
  - 面向资源
    - XML 配置
    - Properties 资源配置
  - 面向注解
  - 面向 API

# Spring Bean 元信息解析阶段

- 面向资源 BeanDefinition 解析
  - BeanDefinitionReader
  - XML 解析器 - BeanDefinitionParser
- 面向注解 BeanDefinition 解析
  - AnnotatedBeanDefinitionReader

# Spring Bean 注册阶段

- BeanDefinition 注册接口
  - BeanDefinitionRegistry

# Spring BeanDefinition 合并阶段

- BeanDefinition 合并
  - 父子 BeanDefinition 合并
    - 当前 BeanFactory 查找
    - 层次性 BeanFactory 查找

# Spring Bean Class 加载阶段

- ClassLoader 类加载
- Java Security 安全控制
- ConfigurableBeanFactory 临时 ClassLoader



# Spring Bean 实例化阶段

- 实例化方式
  - 传统实例化方式
    - 实例化策略 - `InstantiationStrategy`
  - 构造器依赖注入

# Spring Bean 实例化前阶段

- 非主流生命周期 – Bean 实例化前阶段
  - `InstantiationAwareBeanPostProcessor#postProcessBeforeInstantiation`

# Spring Bean 实例化后阶段

- Bean 属性赋值 (Populate) 判断
  - `InstantiationAwareBeanPostProcessor#postProcessAfterInstantiation`

# Spring Bean 属性赋值前阶段

- Bean 属性值元信息
  - PropertyValues
- Bean 属性赋值前回调
  - Spring 1.2 - 5.0: `InstantiationAwareBeanPostProcessor#postProcessPropertyValues`
  - Spring 5.1: `InstantiationAwareBeanPostProcessor#postProcessProperties`

# Spring Bean Aware接口回调阶段

- Spring Aware 接口
  - BeanNameAware
  - BeanClassLoaderAware
  - BeanFactoryAware
  - EnvironmentAware
  - EmbeddedValueResolverAware
  - ResourceLoaderAware
  - ApplicationEventPublisherAware
  - MessageSourceAware
  - ApplicationContextAware

# Spring Bean 初始化前阶段

- 已完成
  - Bean 实例化
  - Bean 属性赋值
  - Bean Aware 接口回调
- 方法回调
  - `BeanPostProcessor#postProcessBeforeInitialization`

# Spring Bean 初始化阶段

- Bean 初始化 (Initialization)
  - `@PostConstruct` 标注方法
  - 实现 `InitializingBean` 接口的 `afterPropertiesSet()` 方法
  - 自定义初始化方法

# Spring Bean 初始化后阶段

- 方法回调
  - `BeanPostProcessor#postProcessAfterInitialization`



# Spring Bean 初始化完成阶段

- 方法回调
  - Spring 4.1 +: `SmartInitializingSingleton#afterSingletonsInstantiated`

# Spring Bean 销毁前阶段

- 方法回调
  - `DestructionAwareBeanPostProcessor#postProcessBeforeDestruction`

# Spring Bean 销毁阶段

- Bean 销毁 (Destroy)
  - `@PreDestroy` 标注方法
  - 实现 `DisposableBean` 接口的 `destroy()` 方法
  - 自定义销毁方法

# Spring Bean 垃圾收集

- Bean 垃圾回收 (GC)
  - 关闭 Spring 容器 (应用上下文)
  - 执行 GC
  - Spring Bean 覆盖的 `finalize()` 方法被回调

# 面试题

沙雕面试题 – BeanPostProcessor 的使用场景有哪些？



我真的没笑

答：BeanPostProcessor 提供 Spring Bean 初始化前和初始化后的生命周期回调，分别对应 `postProcessBeforeInitialization` 以及 `postProcessAfterInitialization` 方法，允许对关心的 Bean 进行扩展，甚至是替换。

加分项：其中，`ApplicationContext` 相关的 `Aware` 回调也是基于 `BeanPostProcessor` 实现，即 `ApplicationContextAwareProcessor`。

# 面试题



## 996 面试题 - BeanFactoryPostProcessor 与 BeanPostProcessor 的区别

答：BeanFactoryPostProcessor 是 Spring BeanFactory（实际为 ConfigurableListableBeanFactory）的后置处理器，用于扩展 BeanFactory，或通过 BeanFactory 进行依赖查找和依赖注入。

加分项：BeanFactoryPostProcessor 必须有 Spring ApplicationContext 执行，BeanFactory 无法与其直接交互。

而 BeanPostProcessor 则直接与BeanFactory 关联，属于 N 对 1 的关系。

# 面试题

## 劝退面试题 – BeanFactory 是怎样处理 Bean 生命周期?



答:

BeanFactory 的默认实现为 DefaultListableBeanFactory, 其中 Bean生命周期与方法映射如下:

- BeanDefinition 注册阶段 – registerBeanDefinition
- BeanDefinition 合并阶段 – getMergedBeanDefinition
- Bean 实例化前阶段 – resolveBeforeInstantiation
- Bean 实例化阶段 – createBeanInstance
- Bean 初始化后阶段 – populateBean
- Bean 属性赋值前阶段 – populateBean
- Bean 属性赋值阶段 – populateBean
- Bean Aware 接口回调阶段 – initializeBean
- Bean 初始化前阶段 – initializeBean
- Bean 初始化阶段 – initializeBean
- Bean 初始化后阶段 – initializeBean
- Bean 初始化完成阶段 – preInstantiateSingletons
- Bean 销毁前阶段 – destroyBean
- Bean 销毁阶段 – destroyBean