

# 第七章：Spring IoC 依赖来源

小马哥 · mercyblitz



扫码试看/订阅

《小马哥讲 Spring 核心编程思想》视频课程

# Spring IoC 依赖来源

---

1. 依赖查找的来源
2. 依赖注入的来源
3. Spring 容器管理和游离对象
4. Spring BeanDefinition 作为依赖来源
5. 单例对象作为依赖来源
6. 非 Spring 容器管理对象作为依赖来源  
游离对象
7. 外部化配置作为依赖来源
8. 面试题精选



## 依赖查找的来源

- 查找来源

来源	配置元数据
Spring BeanDefinition	<code>&lt;bean id="user" class="org.geekbang...User"&gt;</code> xml
	<code>@Bean public User user(){...}</code> 注解
	BeanDefinitionBuilder API
单例对象	API 实现

## 依赖查找的来源

- Spring 內建 BeanDefintion

Bean 名称	Bean 实例	使用场景
org.springframework.context.annotation.internalConfigurationAnnotationProcessor	ConfigurationClassPostProcessor 对象	处理 Spring 配置类
org.springframework.context.annotation.internalAutowiredAnnotationProcessor	AutowiredAnnotationBeanPostProcessor 对象	处理 @Autowired 以及 @Value 注解 + @Inject注解
org.springframework.context.annotation.internalCommonAnnotationProcessor	CommonAnnotationBeanPostProcessor 对象	(条件激活) 处理 JSR-250 注解, 如 @PostConstruct 等
org.springframework.context.event.internalEventListenerProcessor	EventListenerMethodProcessor 对象	处理标注 @EventListener 的 Spring 事件监听方法

## 依赖查找的来源

- Spring 内建单例对象

Bean 名称	Bean 实例	使用场景
environment	Environment 对象	外部化配置以及 Profiles
systemProperties	java.util.Properties 对象	Java 系统属性
systemEnvironment	java.util.Map 对象	操作系统环境变量
messageSource	MessageSource 对象	国际化文案
lifecycleProcessor	LifecycleProcessor 对象	Lifecycle Bean 处理器
applicationEventMulticaster	ApplicationEventMulticaster 对象	Spring 事件广播器

## 依赖注入的来源

- 注入来源

来源	配置元数据
Spring BeanDefinition	<code>&lt;bean id="user" class="org.geekbang...User"&gt;</code>
	<code>@Bean public User user(){...}</code>
	BeanDefinitionBuilder
单例对象	API 实现
非 Spring 容器管理对象	游离对象(非容器的管理对象)

# Spring 容器管理和游离对象

- 依赖对象

来源	Spring Bean 对象	生命周期管理	配置元信息	使用场景
Spring BeanDefinition	是	是	有 <a href="#">BeanDefinition</a>	依赖查找、依赖注入
单体对象	是	否	无	依赖查找、依赖注入
Resolvable Dependency	否	否	无	依赖注入

[非 Spring 容器管理对象](#)



# Spring BeanDefinition 作为依赖来源

- 要素
  - 元数据: BeanDefinition
  - 注册: BeanDefinitionRegistry#registerBeanDefinition
  - 类型: 延迟和非延迟
  - 顺序: Bean 生命周期顺序按照注册顺序
    - 对应变量的 beanDefinitionNames (ArrayList)

## 单例对象作为依赖来源

- 要素
  - 来源：外部普通 Java 对象（不一定是 POJO）
  - 注册：SingletonBeanRegistry#registerSingleton
- 限制
  - 无生命周期管理
  - 无法实现延迟初始化 Bean

## 非 Spring 容器管理对象作为依赖来源

- 要素
  - 注册: `ConfigurableListableBeanFactory#registerResolvableDependency`
- 限制
  - 无生命周期管理
  - 无法实现延迟初始化 Bean
  - 无法通过依赖查找 只能依赖注入

## 外部化配置作为依赖来源

- 要素
  - 类型：非常规 Spring 对象依赖来源
- 限制
  - 无生命周期管理
  - 无法实现延迟初始化 Bean
  - 无法通过依赖查找

## 面试题

**沙雕面试题** - 注入和查找的依赖来源是否相同？

答：否，依赖查找的来源仅限于 Spring BeanDefinition 以及单例对象，而依赖注入的来源还包括 Resolvable Dependency 以及 @Value 所标注的外部化配置



我真的没笑

## 面试题

**996 面试题** - 单例对象能在 IoC 容器启动后注册吗？



答：可以的，单例对象的注册与 BeanDefinition 不同，BeanDefinition 会被 ConfigurableListableBeanFactory#freezeConfiguration() 方法影响，从而冻结注册，单例对象则没有这个限制。

# 面试题

**劝退面试题** - Spring 依赖注入的来源有哪些？



答：

Spring BeanDefinition

单例对象

Resolvable Dependency

@Value 外部化配置



扫码试看/订阅

《小马哥讲 Spring 核心编程思想》视频课程