第五章：Spring IoC 依赖查找

小马哥 • mercyblitz

Spring IoC 依赖查找

1. 依赖查找的今世前生
2. 单一类型依赖查找
3. 集合类型依赖查找
4. 层次性依赖查找
5. 延迟依赖查找
6. 安全依赖查找
7. 内建可查找的依赖
8. 依赖查找中的经典异常
9. 面试题精选

依赖查找的今世前生

- 单一类型依赖查找
  - JNDI - javax.naming.Context#lookup(javax.naming.Name)
  - JavaBeans - java.beans.beancontext.BeanContext
- 集合类型依赖查找
  - java.beans.beancontext.BeanContext
- 层次性依赖查找
  - java.beans.beancontext.BeanContext

单一类型依赖查找

- 单一类型依赖查找接口 - BeanFactory
  - 根据 Bean 名称查找
    - getBean(String)
    - Spring 2.5 覆盖默认参数：getBean(String,Object...)
  - 根据 Bean 类型查找
    - Bean 实时查找
      - Spring 3.0 getBean(Class)
      - Spring 4.1 覆盖默认参数：getBean(Class,Object...)
    - Spring 5.1 Bean 延迟查找
      - getBeanProvider(Class)
      - getBeanProvider(ResolvableType)
  - 根据 Bean 名称 + 类型查找：getBean(String,Class)

集合类型依赖查找

- 集合类型依赖查找接口 - ListableBeanFactory
  - 根据 Bean 类型查找
    - 获取同类型 Bean 名称列表
      - getBeanNamesForType(Class)
      - Spring 4.2 getBeanNamesForType(ResolvableType)
    - 获取同类型 Bean 实例列表
      - getBeansOfType(Class) 以及重载方法
  - 通过注解类型查找
    - Spring 3.0 获取标注类型 Bean 名称列表
      - getBeanNamesForAnnotation(Class<? extends Annotation>)
    - Spring 3.0 获取标注类型 Bean 实例列表
      - getBeansWithAnnotation(Class<? extends Annotation>)
    - Spring 3.0 获取指定名称 + 标注类型 Bean 实例
      - findAnnotationOnBean(String,Class<? extends Annotation>)

层次性依赖查找

- 层次性依赖查找接口 - HierarchicalBeanFactory
  - 双亲 BeanFactory：getParentBeanFactory()
  - 层次性查找
    - 根据 Bean 名称查找
      - 基于 containsLocalBean 方法实现
    - 根据 Bean 类型查找实例列表
      - 单一类型：BeanFactoryUtils#beanOfType
      - 集合类型：BeanFactoryUtils#beansOfTypeIncludingAncestors
    - 根据 Java 注解查找名称列表
      - BeanFactoryUtils#beanNamesForTypeIncludingAncestors

延迟依赖查找

- Bean 延迟依赖查找接口
  - org.springframework.beans.factory.ObjectFactory
  - org.springframework.beans.factory.ObjectProvider
    - Spring 5 对 Java 8 特性扩展
      - 函数式接口
        - getIfAvailable(Supplier)
        - ifAvailable(Consumer)
      - Stream 扩展 - stream()

安全依赖查找

- 依赖查找安全性对比

依赖查找类型	代表实现	是否安全
单一类型查找	BeanFactory#getBean	否
	ObjectFactory#getObject	否
	ObjectProvider#getIfAvailable	是
集合类型查找	ListableBeanFactory#getBeansOfType	是
	ObjectProvider#stream	是

注意：层次性依赖查找的安全性取决于其扩展的单一或集合类型的 BeanFactory 接口

内建可查找的依赖

- AbstractApplicationContext 内建可查找的依赖

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

内建可查找的依赖

- 注解驱动 Spring 应用上下文内建可查找的依赖（部分）

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
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

内建可查找的依赖

- 注解驱动 Spring 应用上下文内建可查找的依赖（续）

Bean 名称	Bean 实例	使用场景
org.springframework.context.event.internalEventListenerFactory	DefaultEventListenerFactory 对象	@EventListener 事件监听方法适配为 ApplicationListener
org.springframework.context.annotation.internalPersistenceAnnotationProcessor	PersistenceAnnotationBeanPostProcessor 对象	（条件激活）处理 JPA 注解场景

依赖查找中的经典异常

- BeansException 子类型

异常类型	触发条件（举例）	场景举例
NoSuchBeanDefinitionException	当查找 Bean 不存在于 IoC 容器时	BeanFactory#getBean ObjectFactory#getObject
NoUniqueBeanDefinitionException	类型依赖查找时，IoC 容器存在多个 Bean 实例	BeanFactory#getBean(Class)
BeanInstantiationException	当 Bean 所对应的类型非具体类时	BeanFactory#getBean
BeanCreationException	当 Bean 初始化过程中	Bean 初始化方法执行异常时
BeanDefinitionStoreException	当 BeanDefinition 配置元信息非法时	XML 配置资源无法打开时

面试题

沙雕面试题 - ObjectFactory 与 BeanFactory 的区别？

答：ObjectFactory 与 BeanFactory 均提供依赖查找的能力。

不过 ObjectFactory 仅关注一个或一种类型的 Bean 依赖查找，并且自身不具备依赖查找的能力，能力则由 BeanFactory 输出。

BeanFactory 则提供了单一类型、集合类型以及层次性等多种依赖查找方式。

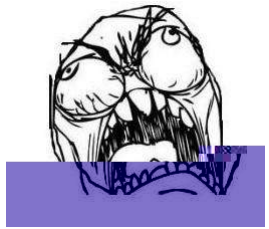
面试题

996 面试题 - BeanFactory.getBean 操作是否线程安全？

答：BeanFactory.getBean 方法的执行是线程安全的，操作过程中会增加互斥锁

面试题

劝退面试题 - Spring ？



答：答案将《Spring IoC依赖注入》以及《Spring IoC依赖来源》章节中继续讨论。