## Spring boot ioc

概念 IOC:inversion of control, 指将对象的依赖管理关系

Dependence injection是实现ioc的方式。

源码：

@SpringBootApplication: 多个注解的集成

1. @SpringBootConfiguration : ->表明这是一个spring boot的@Configuration, 所以可以在该类下使用@Bean
2. **@EnableAutoConfiguration**

启用spring boot context auto-configuration尽可能聪明地加载配置。@Import(AutoConfigurationImportSelector.class)。 @AutoConfigurationPackage：指定扫描包的路径功能类比@ComponentScan，当没有使用@ComponentScan时，相当于扫描启动类所在包的所在包及其子包路径。

执行阶段： 扫描META-INF/spring.factories中的配置定义（这里就是SPI），加载ApplicationContextInitializer和ApplicationListener。注册到容器，以待之后回调使用。 面向过程的， 在不同阶段调用以作处理。

--》ConfigurableApplicationContext（->**ApplicationContext** -> **BeanFactory**）。 实例化类为**AnnotationConfigApplicationContext**。

AnnotationCOnfigApplicationContext的初始化过程有两个关键成员负责对beanDefinition进行扫描AnnotationBeanDefinitionReader与ClassPathBeanDefinitionScanner。其中单例对象由**DefulatSingletonBeanRegistry**负责处理。

对于@Configuraion的类， 由ConfigurationClassPostProcessor处理。

核心方法： AbstractAutowireCapableBeanFactory.populateBean(…)方法

Bean的生命周期：实例化(反射) -> 依赖注入 -> …->BeanPostProcessor->initializingBean ->销毁

使用三级缓存来解决循环依赖（DefaultSingletonRegistry）， 当然， 通常设计下不应当出现循环依赖。

## 2. aop

## 3. 常见提问

1. bean的生命周期

2. 如何解决循环依赖

3. spring boot 事务的实现原理

## 4. 关键词

IOC: Bean,BeanDefinition, BeanDefinitionRegistry, BeanFactory, ApplicationContext,BeanPostProcessord