1. **ImportSelector和DeferredImportSelector的区别：**

DeferredImportSelector是ImportSelector的一个扩展

ImportSelector实例的selectImports方法的执行时机，是在@Configguration注解中的其他逻辑被处理\*\*之前\*\*，所谓的其他逻辑，包括对@ImportResource、@Bean这些注解的处理（**注意，这里只是对@Bean修饰的方法的处理，并不是立即调用@Bean修饰的方法，这个区别很重要！**）；

DeferredImportSelector实例的selectImports方法的执行时机，是在@Configguration注解中的其他逻辑被处理\*\*完毕之后\*\*

DeferredImportSelector的实现类可以用Order注解，或者实现Ordered接口来对selectImports的执行顺序排序（ImportSelector不支持）

ImportSelector是Spring3.1提供的，DeferredImportSelector是Spring4.0提供的

Spring Boot的自动配置功能就是通过DeferredImportSelector接口的实现类EnableAutoConfigurationImportSelector做到的（**因为自动配置必须在我们自定义配置后执行才行**）

源码详情：<https://cloud.tencent.com/developer/article/1497795>

1. @Configuration full 模式和 lite模式

full和lite模式

@Configuration注解标记的类是配置类，BeanDefinition被标记为full 类型

注解@Component @ComponentScan @Import @ImportResource 标注的类是配置类，BeanDefinition被标记为lite 类型

类中的方法被@Bean注解标记的类也是配置类，Bean定义信息被标记为lite 类型

lite模式

1.该模式下，配置类本身不会被CGLIB增强，放进IoC容器内的就是类本身

2. 该模式下，对于内部类是没有限制的：可以是Full模式或者Lite模式

3. 该模式下，配置类内部不能通过方法调用来处理依赖，否则每次生成的都是一个新实例而并非IoC容器内的单例

4. 该模式下，配置类就是一普通类，所以@Bean方法可以使用private/final等进行修饰

Full 和lite 源码解析ConfigurationClassPostProcessor：https://blog.csdn.net/f641385712/article/details/88095165

区别小结：https://zhuanlan.zhihu.com/p/346542455

1. SpringBoot 自动装配

就是通过SpringBootApplication注解 里面的@EnableAutoConfiguration注解，通过Import注解加载AutoConfigurationImportSelector类中的，这个类实现了DeferredSelectImport接口，里面有个selectImports方法，会扫描并筛选MATE-INF下的spring.factories文件下的自动配置类，，selectImport方法的执行在AbstractApplicationContext.refresh 方法里面有个invokeBeanFactoryPostProcessors（激活各种BeanFactory处理器）这里会执行ConfigurationClassPostProcessor 里面的一个postProcessBeanFactory方法，通过递归解析注册beanDefinition，把这个自动配置类以注解@Configuration 相同的处理方式装配到IOC容器里面

SpringBoot中Condition下常用条件依赖注解详细介绍及案例使用

@ConditionalOnClass() Spring中存在指定class对象时，注入指定配置

@ConditionalOnMissingClass() Spring中不存在指定class对象时，注入指定配置

@ConditionalOnBean() 根据ioc中判断有没有此实例对象,有则加载指定配置

ConditionalOnMissingBean() 根据ioc中判断有没有此实例对象,没有则加载指定配置

@ConditionalOnProperty() 配置文件中配置的指定参数值符合要求时，加载此配置

@ConditionalOnExpression() 配置文件中配置的指定参数值与我给的值比较为true时，加载此配置

@ConditionalOnResource() 指定的资源文件出现在classpath中生效(就是编译后target里classes里的路径名是否存在)