

快速开发框架 Spring Boot

知识点暨面试题总结

【Q-01】 对于 Spring Boot 来说，最为重要的注解应该就是@SpringBootApplication 了，请谈一下你对这个注解的认识。

【RA】 Spring Boot 中的@SpringBootApplication 是一个组合注解。除了基本的元注解外，它还组合了三个很重要的注解：

- @SpringBootConfiguration：其等价于@Configuration，表示当前类为一个配置类。
- @ComponentScan：该注解用于配置应用中 Bean 的扫描指令，但其并没有进行真正的扫描。
- @EnableAutoConfiguration：这是核心注解，其开启了自动配置。对内置自动配置类的加载及对自定义 Bean 的扫描都是由该注解完成的。

【Q-02】 Spring Boot 中有一个注解@ComponentScan，请谈一下你对这个注解的认识。

【RA】 Spring Boot 中的注解@ComponentScan 是@SpringBootApplication 注解的一个组成注解，用于配置使用@Configuration 定义的组件的扫描指令，并且支持同时使用 Spring 配置文件中的<context:component-scan/>标签。我查看过该注解的注释，注释上说，该注解仅仅就是配置了一下用于进行组件扫描的指令参数，并没有进行扫描，真正扫描并装配这些类是@EnableAutoConfiguration 完成的。而这些指令参数就是通过该注解的属性进行配置的，例如扫描哪里，扫描之前可以先进行怎样的过滤等。如果这些注解属性都没有配置，则默认会扫描当前注解所标注类所在的包及其子孙包。

不过，对于这点，Spring Boot1.x 版本中仅会扫描当前标注类所在包的子孙包，不会扫描标注类所在的包。但 Spring Boot2.x 版本中默认会扫描当前注解所标注类所在的包及其子孙包。

对于一个 Spring Boot 工程，与自动配置相关的 Bean 均是由 Spring 容器管理的。而这些 Bean 的类型根据创建者的不同可以分为两种：一种是由程序员自定义的组件类，例如我们自己定义的处理类、Service 类等；另一种是框架本身已经定义好的自动配置相关类。

自定义类由@ComponentScan 指定的扫描指令进行扫描，而框架自身的自动配置相关类在一个配置文件中存放，将来会被加载到内存。这两种类型的类都会由注解@EnableAutoConfiguration 交给 Spring 容器来管理。

【Q-03】 Spring Boot 中有一个注解@EnableAutoConfiguration，请谈一下你对这个注解的认识。

【RA】 首先我想谈一下我对@Enable 开头的这一类注解的认识，然后我再谈一下对 Spring Boot 中的注解@EnableAutoConfiguration 的认识。

首先，@Enable 开头这一类注解一般用于开启某一项功能，是为了简化代码的导入，即使用了该类注解，就会自动导入某些类。所以该类注解是组合注解，一般都会组合一个@Import 注解，用于导入指定的多个类。@EnableXxx 的功能主要体现在这些被导入的类上，而被导入的类一般有三种：配置类、可以实现动态选择的选择器，与可以完成动态注解的注册器。

然后，Spring Boot 中的注解@EnableAutoConfiguration 是@SpringBootApplication 注解的

一个组成注解，该注解用于完成自动配置相关的自定义类及内置类的加载。其本身也是一个组合注解。除了元注解外，还组合了 `@Import` 与 `@AutoConfigurationPackage` 两个注解。具体分工如下：

- `@Import`：用于加载 Spring Boot 中内置的及导入 starter 中 META-INF/spring.factory 配置中的自动配置类。
- `@AutoConfigurationPackage`：用于扫描、加载并注册自定义的组件类。

【Q-04】 META-INF/spring.factory 配置文件对于 Spring Boot 的自动配置很重要，为什么？

【RA】 无论是 Spring Boot 内置的 META-INF/spring.factory 配置文件，还是导入 Starter 依赖中的 META-INF/spring.factory 配置文件，对于 Spring Boot 自动配置来说很重要是因为，其包含了一个 key-value 对，key 为 `EnableAutoConfiguration` 的全限定性类名，而 value 则为当前应用中所有可用的自动配置类。所有可用的自动配置类都在这里声明，所以该文件对于 Spring Boot 自动配置来说很重要。

【Q-05】 你了解 Spring Boot 官方给出的 Starter 工程的命名规范吗？具体是什么？

【RA】 Spring Boot 官方给出的 Starter 工程的命名需要遵循如下规范：

- Spring 官方定义的 Starter 格式为：spring-boot-starter-{name}，如 spring-boot-starter-web。
- 非官方 Starter 命名格式为：{name}-spring-boot-starter，如 dubbo-spring-boot-starter。

【Q-06】 我们知道 `@Configuration` 所注解的类称为配置类，其中的 `@Bean` 方法可以创建相应实例。请问这些 `@Bean` 方法实例的创建顺序什么？

【RA】 `@Configuration` 所注解的类中的 `@Bean` 方法实例的创建顺序即这些 `@Bean` 方法的执行顺序。首先可以为这些方法的执行添加执行条件，例如使用以 `@ConditionalOn` 开头的条件注解。在这些条件都满足的情况下，这些方法的执行顺序即为其在 `@Configuration` 注解类中的定义顺序，先定义者先执行，其对应实例先创建。

【Q-07】 你曾定义过 Starter 吗？简单说一下定义的大体步骤。

【RA】 对于自定义 Starter，其工程命名格式为 {name}-spring-boot-starterConfiguration。工程需要导入配置处理器依赖。然工程中需要定义如下的类与配置：

- 定义核心业务类，这是该 Starter 存在的意义。
- 定义自动配置类，其完成对核心业务类的实例化。
- 若核心业务类中需要从配置文件获取配置数据，还需要定义一个用于封装配置文件中相关属性的类。
- 定义 META-INF/spring.factories 配置文件，用于对自动配置类进行注册。

【Q-08】 在自动配置类中一般会涉及到很多的条件注解，简单介绍几个你了解的条件注解。

【RA】 在自动配置类定义中的确会用到很多的条件注解，条件注解一般都是以 `@ConditionalOn` 开头的，例如：

- `@ConditionalOnClass()`：条件判断注解，其可以标注在类与方法上，表示当参数指定的类在类路径下存在时才会创建当前类的 Bean，或当前方法指定的 Bean。
- `@ConditionalOnMissingBean`：条件判断注解，其可以标注在类与方法上，表示当容器中不存在当前类或方法类型的对象时，将去创建一个相应的 Bean。
- `@ConditionalOnBean()`：条件判断注解，其可以标注在类与方法上，表示当在容器中存在指定类的实例时才会创建当前类的 Bean，或当前方法指定的 Bean。

【Q-09】请简单谈一下你对 Spring Boot 启动过程的了解。

【RA】Spring Boot 的启动从大的方面来说需要经过以下两大阶段：加载 Spring Boot 配置文件 application.yml，与完成自动配置。

而自动配置又包含以下两个大的步骤：加载自动配置相关的类，与扫描并注册自定义的组件类。

加载 Spring Boot 配置文件是在启动类的 run()方法执行时加载的。其大体要经历运行环境准备、为环境准备过程添加监听、发布环境准备事件等 6 步。

自动配置则是由组合注解 @SpringBootApplication 所包含的子注解 @EnableAutoConfiguration 完成。其不仅加载了 META-INF/spring.factories 中内置的自动配置相关类，还完成了自定义类的加载与注册。若存在第三方 Starter，则其会将该 Starter 中 META-INF/spring.factories 中的自动配置相关类加载并装配。

【Q-10】请简单谈一下你对 Spring Boot 自动配置的理解。

【RA】Spring Boot 与 SSM 传统开发相比，其最大的特点是自动配置，不用再在 xml 文件中做大量的配置了。自动配置的实现主要是通过自动配置类来完成的，自动配置类存在于两类位置：一个是 Spring Boot 框架中内置的，一是从外部引入的 Starter 中。

具体来说，自动配置类的作用就是根据条件创建核心业务类的实例到 Spring 容器中，以备该 Starter 的引用工程类中注入。当然，自动配置类还有一个作用：若创建核心业务类时需要获取配置文件中的相关属性值，其也会将这些属性值封装为一个属性实例，以备核心业务类使用。当然，自动配置类需要在 META-INF/spring.factories 中注册。

所以自动配置其实就是能够自动获取配置文件中的属性并创建相应核心业务类实例。