

# Spring Boot 最核心的 25 个注解，都是干货！

原创：栈长 Java技术栈 2月26日



Java技术栈

www.javastack.cn

优秀的Java技术公众号

学习和应用 Spring Boot 有一些时间了，你们对 Spring Boot 注解了解有多少呢？今天栈长我给大家整理了 Spring Boot 最核心的 25 个注解，都是干货！

## 你所需具备的基础

- [什么是 Spring Boot?](#)
- [Spring Boot 核心配置文件详解](#)
- [Spring Boot 开启的 2 种方式](#)
- [Spring Boot 自动配置原理、实战](#)
- [Spring Boot 2.x 启动全过程源码分析](#)
- [Java 必须掌握的 12 种 Spring 常用注解！](#)

更多请在Java技术栈微信公众号后台回复关键字：**boot**。

## Spring Boot 最核心的 25 个注解

### 1、@SpringBootApplication

这是 Spring Boot 最最最核心的注解，用在 Spring Boot 主类上，标识这是一个 Spring Boot 应用，用来开启 Spring Boot 的各项能力。

其实这个注解就是

`@SpringBootApplication`、`@EnableAutoConfiguration`、`@ComponentScan` 这三个注

解的组合，也可以用这三个注解来代替 `@SpringBootApplication` 注解。

## 2、@EnableAutoConfiguration

允许 Spring Boot 自动配置注解，开启这个注解之后，Spring Boot 就能根据当前类路径下的包或者类来配置 Spring Bean。

如：当前类路径下有 Mybatis 这个 JAR 包，`MybatisAutoConfiguration` 注解就能根据相关参数来配置 Mybatis 的各个 Spring Bean。

## 3、@Configuration

这是 Spring 3.0 添加的一个注解，用来代替 applicationContext.xml 配置文件，所有这个配置文件里面能做到的事情都可以通过这个注解所在类来进行注册。

## 4、@SpringBootConfiguration

这个注解就是 `@Configuration` 注解的变体，只是用来修饰是 Spring Boot 配置而已，或者可利于 Spring Boot 后续的扩展。

## 5、@ComponentScan

这是 Spring 3.1 添加的一个注解，用来代替配置文件中的 component-scan 配置，开启组件扫描，即自动扫描包路径下的 `@Component` 注解进行注册 bean 实例到 context 中。

前面 5 个注解可以在这篇文章《[Spring Boot 最核心的 3 个注解详解](#)》中了解更多细节的。

## 6、@Conditional

这是 Spring 4.0 添加的新注解，用来标识一个 Spring Bean 或者 Configuration 配置文件，当满足指定的条件才开启配置。

## 7、@ConditionalOnBean

组合 `@Conditional` 注解，当容器中有指定的 Bean 才开启配置。

## 8、@ConditionalOnMissingBean

组合 `@Conditional` 注解，和 `@ConditionalOnBean` 注解相反，当容器中没有指定的

Bean 才开启配置。

## 9、@ConditionalOnClass

组合 `@Conditional` 注解，当容器中有指定的 Class 才开启配置。

## 10、@ConditionalOnMissingClass

组合 `@Conditional` 注解，和 `@ConditionalOnMissingClass` 注解相反，当容器中没有指定的 Class 才开启配置。

## 11、@ConditionalOnWebApplication

组合 `@Conditional` 注解，当前项目类型是 WEB 项目才开启配置。

当前项目有以下 3 种类型。

```
enum Type {  
  
    /**  
     * Any web application will match.  
     */  
    ANY,  
  
    /**  
     * Only servlet-based web application will match.  
     */  
    SERVLET,  
  
    /**  
     * Only reactive-based web application will match.  
     */  
    REACTIVE  
  
}
```

## 12、@ConditionalOnNotWebApplication

组合 `@Conditional` 注解，和 `@ConditionalOnWebApplication` 注解相反，当前项目类型不是 WEB 项目才开启配置。

## 13、@ConditionalOnProperty

组合 `@Conditional` 注解，当指定的属性有指定的值时才开启配置。

## 14、@ConditionalOnExpression

组合 `@Conditional` 注解，当 SpEL 表达式为 true 时才开启配置。

## 15、@ConditionalOnJava

组合 `@Conditional` 注解，当运行的 Java JVM 在指定的版本范围时才开启配置。

## 16、@ConditionalOnResource

组合 `@Conditional` 注解，当类路径下有指定的资源才开启配置。

## 17、@ConditionalOnJndi

组合 `@Conditional` 注解，当指定的 JNDI 存在时才开启配置。

## 18、@ConditionalOnCloudPlatform

组合 `@Conditional` 注解，当指定的云平台激活时才开启配置。

## 19、@ConditionalOnSingleCandidate

组合 `@Conditional` 注解，当指定的 class 在容器中只有一个 Bean，或者同时有多个但为首选时才开启配置。

## 20、@ConfigurationProperties

用来加载额外的配置（如 .properties 文件），可用在 `@Configuration` 注解类，或者 `@Bean` 注解方法上面。

关于这个注解的用法可以参考《[Spring Boot读取配置的几种方式](#)》这篇文章。

## 21、@EnableConfigurationProperties

一般要配合 `@ConfigurationProperties` 注解使用，用来开启对 `@ConfigurationProperties` 注解配置 Bean 的支持。

## 22、@AutoConfigureAfter

用在自动配置类上面，表示该自动配置类需要在另外指定的自动配置类配置完之后。

如 Mybatis 的自动配置类，需要在数据源自动配置类之后。

```
@AutoConfigureAfter(DataSourceAutoConfiguration.class)
public class MybatisAutoConfiguration {
```

## 23、@AutoConfigureBefore

这个和 `@AutoConfigureAfter` 注解使用相反，表示该自动配置类需要在另外指定的自动配置类配置之前。

## 24、@Import

这是 Spring 3.0 添加的新注解，用来导入一个或者多个 `@Configuration` 注解修饰的类，这在 Spring Boot 里面应用很多。

## 25、@ImportResource

这是 Spring 3.0 添加的新注解，用来导入一个或者多个 Spring 配置文件，这对 Spring Boot 兼容老项目非常有用，因为有些配置无法通过 Java Config 的形式来配置就只能用这个注解来导入。

好了，终于总结完了，花了栈长我 2 个小时。。。另外，关注微信公众号Java技术栈，在后台回复关键字：**boot**，可以获取栈长整理的更多 Spring Boot 系列教程文章。

对以上有疑问或者对 Spring Boot 有不懂的都可以点击文末[阅读原文](#)链接加入栈长的知识星球，和栈长及更多 Java 技术人一起学习。

..... END .....

推荐大家关注栈长的另外一个公众号：**Java大后端**，专注分享Java主流后端技术。

关注并在里面回复：**大大**，可以领取一份我收藏的架构师学习资料，要的话就速度了，过阵子不一定还能领取，哈哈！

