

# 谈谈你对SpringBoot的认识和理解

---

## 从Spring谈起

SpringBoot的底层是Spring，也是对Spring的封装。

Spring提供了一种简单的方法通过依赖注入和面向切面编程，但是虽然组件代码是轻量级的，但是配置确是重量级的，需要大量的xml，Spring2.5引入了基于注解的组件扫描，消除了大量针对程序自身组件的显式xml配置，Spring3.0引入基于Java的配置，这是一种类型安全的可重构配置方式，可以代替xml

尽管如此，我们依然没能逃脱配置的魔爪，开启某些Spring的特性的时候，依然需要显式的配置，比如事务管理，SpringMVC，还有一些第三方的配置，也需要显式的xml配置，比如thymeleaf的web视图，配置servlet和过滤器等，组件扫描减少了配置量，但是依然需要大量的配置

光是xml文件的就可能需要占用我们大量的时间，除此之外，相关库的依赖也非常让人头疼，不同库之间的版本冲突非常常见

## SpringBoot的认识

Spring Boot可以轻松创建独立的生产级基于Spring的应用程序,只要通过“just run”（可能是run ‘Application’或java -jar 或 tomcat 或 maven插件run 或 shell脚本）便可以运行项目。大部分Spring Boot项目只需要少量的配置即可

## 优点

1. 开发基于Spring的应用程序很容易
2. SpringBoot项目所需的开发或工程时间明显减少，通常会提高整体的生产力
3. SpringBoot不需要编写大量的样板代码，xml配置和注释
4. Spring引导应用程序可以容易地与Spring生态系统集成，SpringJDBC、SpringORM，SpringData，SpringSecurity等

5. 遵循默认配置，以减少开发工作，默认配置可以修改
6. 提供嵌入式的HTTP服务器，Tomcat和Jetty，可以轻松的开发和测试web应用程序
7. 提供了命令行接口工具（CLI），用于开发和测试springBoot应用程序，如Java或者Groovy
8. 提供和很多的插件，可以使用内置的工具（Maven或者Gradle）开发和测试SpringBoot应用程序

等下我写一篇文章谈谈你对springboot的认识给大家参考和学习？

其实就几个问题：

- 1: springboot怎么做到的零配置？
- 2: springboot是怎么加载starter的原理是什么？
- 3: springboot是怎么自定义starter的。流程和步骤是什么？
- 4: springboot为什么要条件注解？
- 5: springboot和spring的关系是什么？
- 6: SpringBoot的@Import机制解决什么问题？
- 7: SpringBoot的为什么可以通过main函数启动一个tomcat？