# SpringBoot总结

1. **什么是SpringBoot**

1.通过一套默认的**配置,减少**了spring样板化的配置

2.通过集成大量常用的第三方库配置简化了项目的**依赖管理**简化了spring应用的初始化搭建及开发过程。

1. **SpringBoot的特性**

1、创建独立的Spring项目

2、内置Tomcat和Jetty容器

3、提供一个starter POMs来简化Maven配置

4、提供了一系列大型项目中常见的非功能性特性，如安全、指标，健康检测、外部配置等

5、完全没有代码生成和xml配置文件

1. **SpringBoot核心功能**

1、SpringBoot可以以jar包的方式独立运行项目（通过java-jar xx.jar）；

2、内嵌servlet容器（Tomcat、jetty），无需以war包形式部署；

3、提供starter简化Maven的Pom依赖配置；

4、SpringBoot会根据类路径中的jar包自动配置Bean

5、SpringBoot提供基于http,ssh,telnet对运行时的项目进行监控

6、spring4.x提倡使用Java配置和注解配置组合，而springboot不需要任何xml配置即可实现spring的所有配置。

1. **SpringBoot的优缺点**

**优点:**

**1、**快速构建项目

2、对主流框架无缝集成

3、项目可以独立运行，无需依赖外部servlet容器

4、提供运行时的应用监控

5、极大提高了开发，部署效率

6、与Docker容器等集成方便

**缺点:**

1、依赖太多,随便的一个Spring Boot 应用都有好几十M;

2、缺少服务的注册和发现等解决方案;

3、却少监控集成方案,安全管理方案

# SpringBoot常用注解

（1）@RestController和@Controller指定一个类，作为控制器的注解

（2）@RequestMapping方法级别的映射注解，这一个用过Spring MVC的小伙伴相信都很熟悉

（3）@EnableAutoConfiguration和@SpringBootApplication是类级别的注解，根据maven依赖的jar来自动猜测完成正确的spring的对应配置，只要引入了spring-boot-starter-web的依赖，默认会自动配置Spring MVC和tomcat容器

（4）@Configuration类级别的注解，一般这个注解，我们用来标识main方法所在的类,完成元数据bean的初始化。

（5）@ComponentScan类级别的注解，自动扫描加载所有的Spring组件包括Bean注入，一般用在main方法所在的类上

（6）@ImportResource类级别注解，当我们必须使用一个xml的配置时，使用@ImportResource和@Configuration来标识这个文件资源的类。

（7）@Autowired注解，一般结合@ComponentScan注解，来自动注入一个Service或Dao级别的Bean

（8）@Component类级别注解，用来标识一个组件，比如我自定了一个filter，则需要此注解标识之后，Spring Boot才会正确识别。