https://mp.weixin.qq.com/s/SgaW-cE-I6iGq-gcVqBlsw

# Spring 和 Spring Boot 的区别是什么

## **概述**

对于 Spring和 SpringBoot到底有什么区别，我听到了很多答案，刚开始迈入学习 SpringBoot的我当时也是一头雾水，随着经验的积累、我慢慢理解了这两个框架到底有什么区别，相信对于用了 SpringBoot很久的同学来说，还不是很理解 SpringBoot到底和 Spring有什么区别，看完文章中的比较，或许你有了不同的答案和看法！

## **什么是Spring**

作为 Java开发人员，大家都 Spring都不陌生，简而言之， Spring框架为开发 Java应用程序提供了全面的基础架构支持。它包含一些很好的功能，如依赖注入和开箱即用的模块，如：

SpringJDBC、SpringMVC、SpringSecurity、SpringAOP、SpringORM、SpringTest，这些模块缩短应用程序的开发时间，提高了应用开发的效率例如，在 JavaWeb开发的早期阶段，我们需要编写大量的代码来将记录插入到数据库中。但是通过使用 SpringJDBC模块的 JDBCTemplate，我们可以将操作简化为几行代码。

## **什么是Spring Boot**

SpringBoot基本上是 Spring框架的扩展，它消除了设置 Spring应用程序所需的 XML配置，为更快，更高效的开发生态系统铺平了道路。

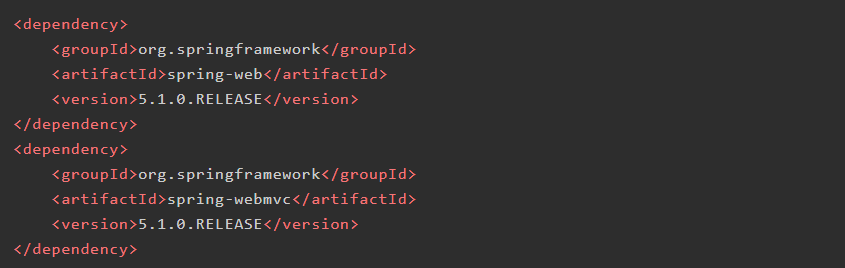
SpringBoot**中的一些特征：**

**1、**创建独立的 Spring应用。  
**2、**嵌入式 Tomcat、 Jetty、 Undertow容器（无需部署war文件）。  
**3、**提供的 starters 简化构建配置  
**4、**尽可能自动配置 spring应用。  
**5、**提供生产指标,例如指标、健壮检查和外部化配置  
**6、**完全没有代码生成和 XML配置要求

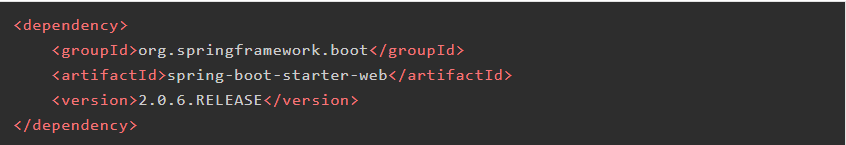
## **从配置分析**

### **Maven依赖**

**首先，让我们看一下使用Spring创建Web应用程序所需的最小依赖项**



**与Spring不同，Spring Boot只需要一个依赖项来启动和运行Web应用程序：**



**在进行构建期间，所有其他依赖项将自动添加到项目中。**

另一个很好的例子就是测试库。我们通常使用 SpringTest， JUnit， Hamcrest和 Mockito库。在 Spring项目中，我们应该将所有这些库添加为依赖项。但是在 SpringBoot中，我们只需要添加 spring-boot-starter-test依赖项来自动包含这些库。

**Spring Boot为不同的Spring模块提供了许多依赖项。一些最常用的是：**

spring-boot-starter-data-jpa

spring-boot-starter-security

spring-boot-starter-test

spring-boot-starter-web

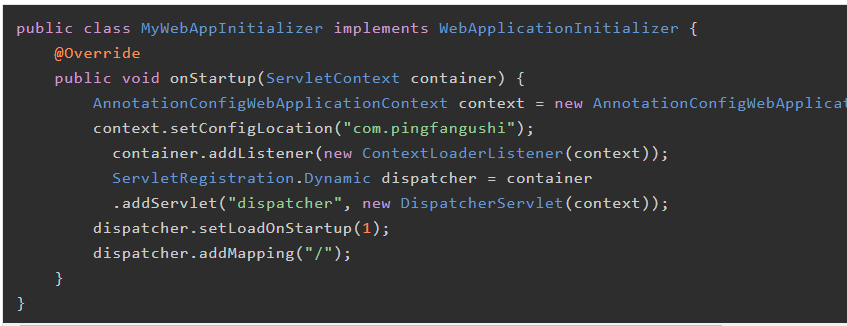
spring-boot-starter-thymeleaf

有关 starter的完整列表，请查看Spring文档。

### **MVC配置**

**让我们来看一下** Spring**和** SpringBoot**创建** JSP Web**应用程序所需的配置。**

Spring需要定义调度程序 servlet，映射和其他支持配置。我们可以使用 web.xml 文件或 Initializer类来完成此操作：



还需要将 @EnableWebMvc注释添加到 @Configuration类，并定义一个视图解析器来解析从控制器返回的视图：



再来看 SpringBoot一旦我们添加了 Web启动程序， SpringBoot只需要在 application配置文件中配置几个属性来完成如上操作：



**上面的所有Spring配置都是通过一个名为auto-configuration的过程添加** Boot web starter**来自动包含的。**

这意味着 SpringBoot将查看应用程序中存在的依赖项，属性和 bean，并根据这些依赖项，对属性和 bean进行配置。当然，如果我们想要添加自己的自定义配置，那么 SpringBoot自动配置将会退回。

### **配置模板引擎**

**现在我们来看下如何在Spring和Spring Boot中配置Thymeleaf模板引擎。**

在 Spring中，我们需要为视图解析器添加 thymeleaf-spring5依赖项和一些配置：

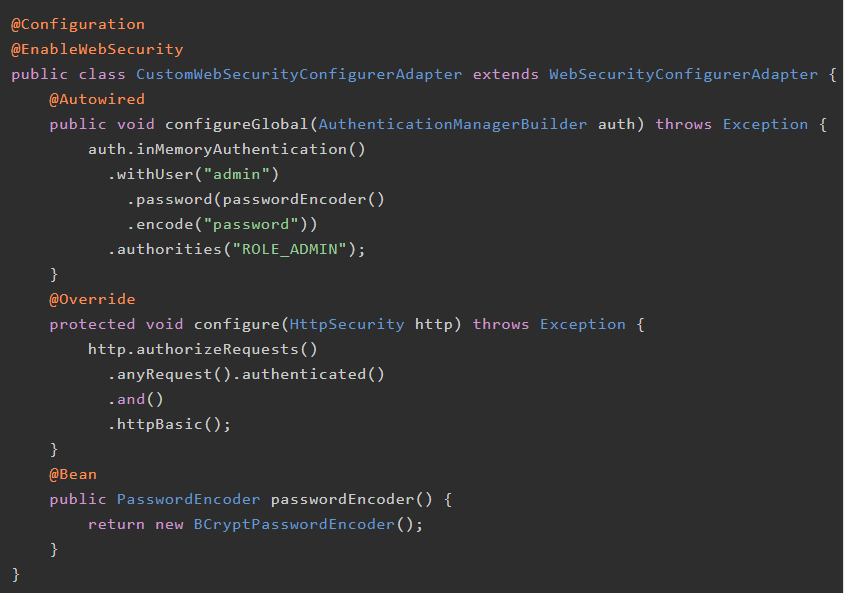


SpringBoot1X只需要 spring-boot-starter-thymeleaf的依赖项来启用 Web应用程序中的 Thymeleaf支持。   但是由于 Thymeleaf3.0中的新功能，我们必须将 thymeleaf-layout-dialect 添加为 SpringBoot2XWeb应用程序中的依赖项。配置好依赖，我们就可以将模板添加到 src/main/resources/templates文件夹中， SpringBoot将自动显示它们。

### **Spring Security 配置**

为简单起见，我们使用框架默认的 HTTPBasic身份验证。让我们首先看一下使用 Spring启用 Security所需的依赖关系和配置。

Spring首先需要依赖 spring-security-web和 spring-security-config 模块。接下来， 我们需要添加一个扩展 WebSecurityConfigurerAdapter的类，并使用 @EnableWebSecurity注解：



这里我们使用 inMemoryAuthentication来设置身份验证。同样， SpringBoot也需要这些依赖项才能使其工作。但是我们只需要定义 spring-boot-starter-security的依赖关系，因为这会自动将所有相关的依赖项添加到类路径中。

SpringBoot**中的安全配置与上面的相同** 。

## **应用程序启动引导配置**

Spring和 SpringBoot中应用程序引导的基本区别在于 servlet。Spring使用 web.xml 或 SpringServletContainerInitializer作为其引导入口点。SpringBoot仅使用 Servlet3功能来引导应用程序，下面让我们详细来了解下

### **Spring 引导配置**

Spring支持传统的 web.xml引导方式以及最新的 Servlet3+方法。

**配置** web.xml**方法启动的步骤**

Servlet容器（服务器）读取 web.xml

web.xml中定义的 DispatcherServlet由容器实例化

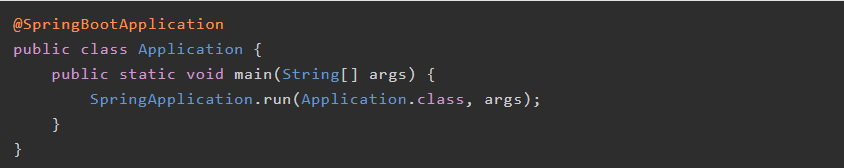
DispatcherServlet通过读取 WEB-INF/{servletName}-servlet.xml来创建 WebApplicationContext。最后， DispatcherServlet注册在应用程序上下文中定义的 bean

**使用** Servlet3+**方法的** Spring**启动步骤**

容器搜索实现 ServletContainerInitializer的类并执行 SpringServletContainerInitializer找到实现所有类 WebApplicationInitializer``WebApplicationInitializer创建具有XML或上下文 @Configuration类 WebApplicationInitializer创建 DispatcherServlet与先前创建的上下文。

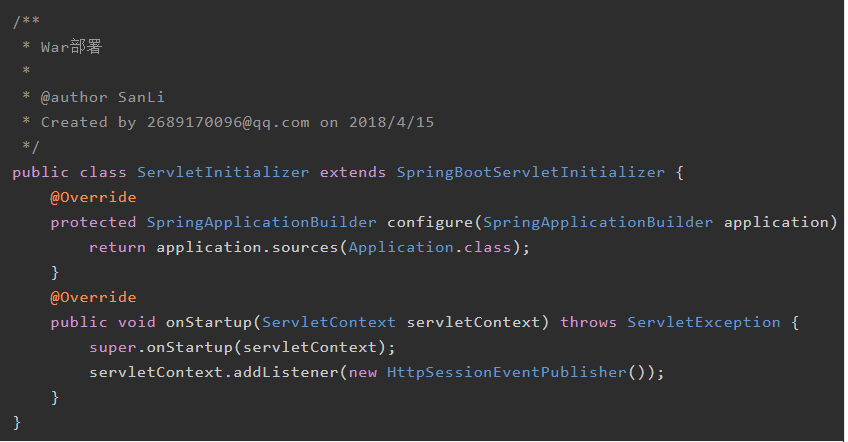
### **SpringBoot 引导配置**

**Spring Boot应用程序的入口点是使用@SpringBootApplication注释的类**



默认情况下， SpringBoot使用嵌入式容器来运行应用程序。在这种情况下， SpringBoot使用 publicstaticvoidmain入口点来启动嵌入式 Web服务器。此外，它还负责将 Servlet， Filter和 ServletContextInitializerbean从应用程序上下文绑定到嵌入式 servlet容器。SpringBoot的另一个特性是它会自动扫描同一个包中的所有类或 Main类的子包中的组件。

SpringBoot提供了将其部署到外部容器的方式。我们只需要扩展 SpringBootServletInitializer即可：



这里外部 servlet容器查找在war包下的 META-INF文件夹下MANIFEST.MF文件中定义的 Main-class， SpringBootServletInitializer将负责绑定 Servlet， Filter和 ServletContextInitializer。

## **打包和部署**

最后，让我们看看如何打包和部署应用程序。这两个框架都支持 Maven和 Gradle等通用包管理技术。但是在部署方面，这些框架差异很大。例如，Spring Boot Maven插件在 Maven中提供 SpringBoot支持。它还允许打包可执行 jar或 war包并 就地运行应用程序。

在部署环境中 SpringBoot 对比 Spring的一些优点包括：

**1、**提供嵌入式容器支持  
**2、**使用命令*java -jar*独立运行jar  
**3、**在外部容器中部署时，可以选择排除依赖关系以避免潜在的jar冲突  
**4、**部署时灵活指定配置文件的选项  
**5、**用于集成测试的随机端口生成

## **结论**

简而言之，我们可以说 SpringBoot只是 Spring本身的扩展，使开发，测试和部署更加方便。