### 1、什么是Spring

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。

### 2、什么是SpringBoot

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starters的简化构建配置
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

### 3、从配置分析

#### 3.1 Maven依赖

使用Spring创建Web应用程序所需的最小依赖项：

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>5.1.0.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.1.0.RELEASE</version>

</dependency>

SpringBoot只需要一个依赖来启动和运行Web应用程序：

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<version>2.0.6.RELEASE</version>

</dependency>

在构建期间，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

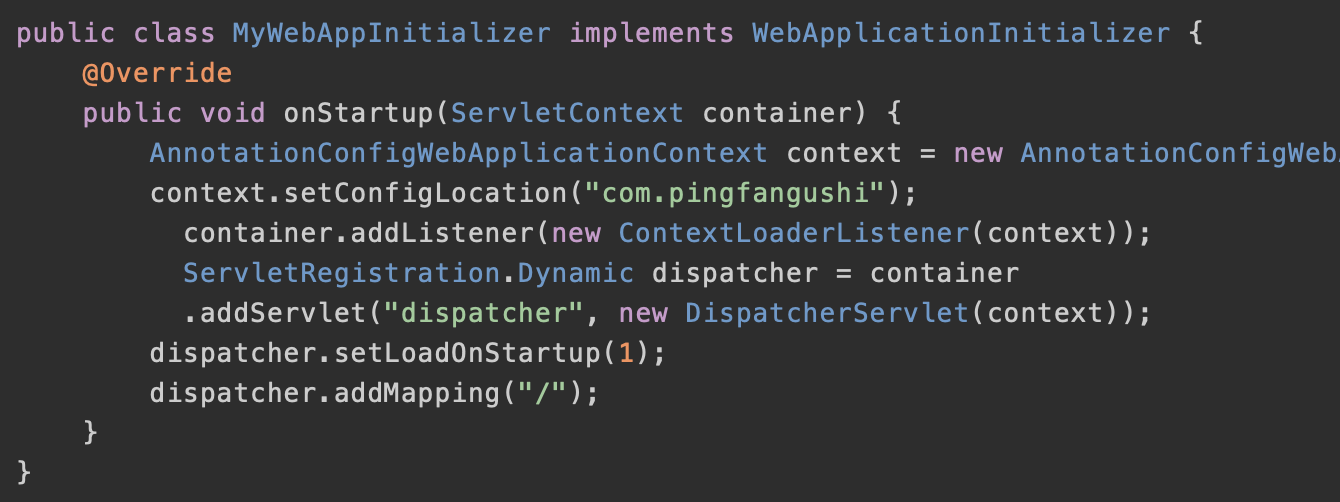
spring-boot-starter-web

spring-boot-starter-thymeleaf

#### 3.2 MVC配置

看一下Spring和SpringBoot创建JSP Web应用程序所需的配置。

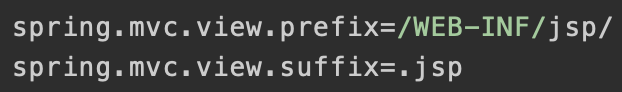
Spring需要定义调度程序servlet，映射和其他支持配置。我们可以使用web.xml文件或Initializer类来完成此操作：



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 Bootweb 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。

#### 3.3 配置模版引擎

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：



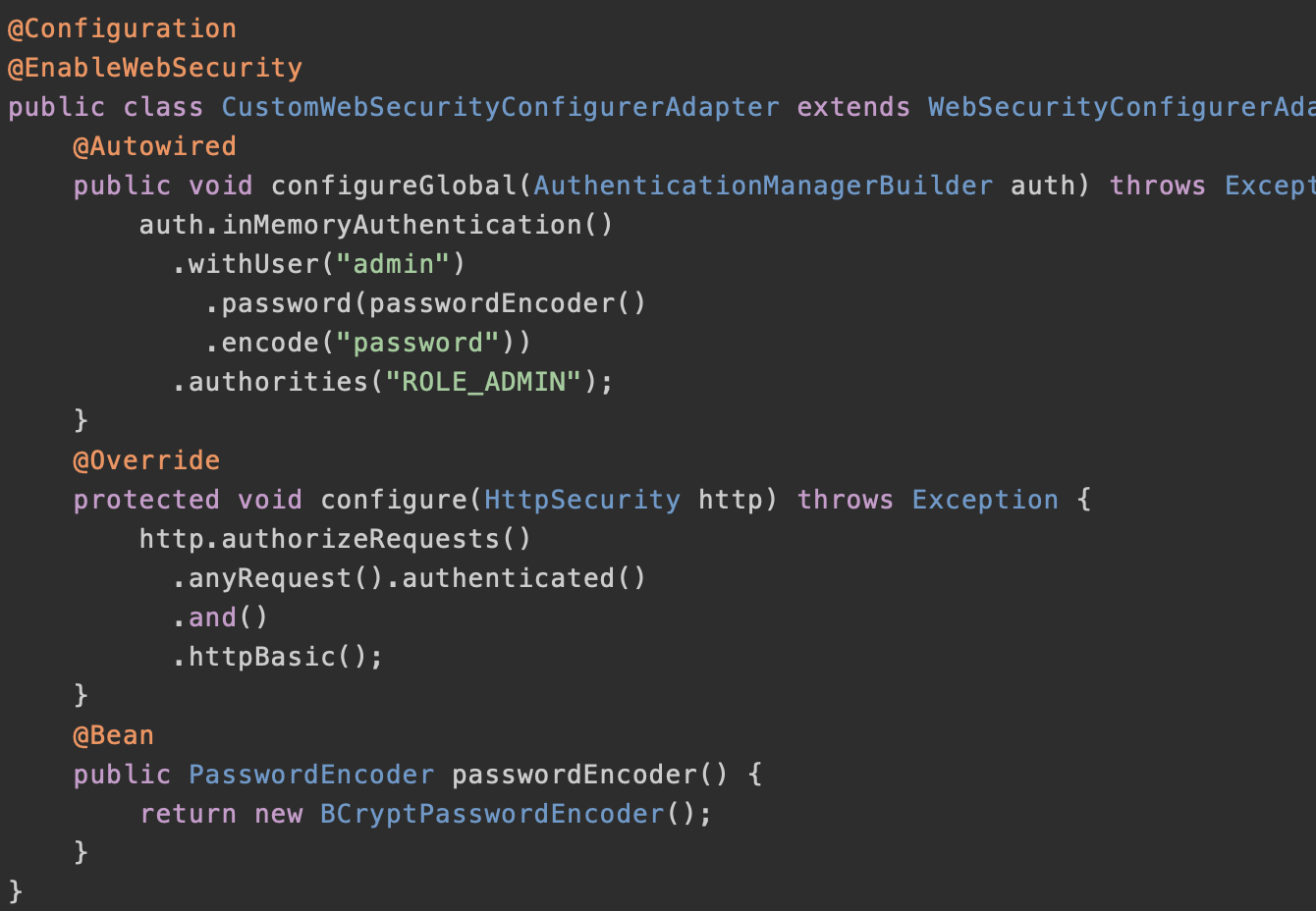
SpringBoot1X只需要 spring-boot-starter-thymeleaf的依赖项来启用 Web应用程序中的 Thymeleaf支持。

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。

#### 3.4 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 。

#### 3.5 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：

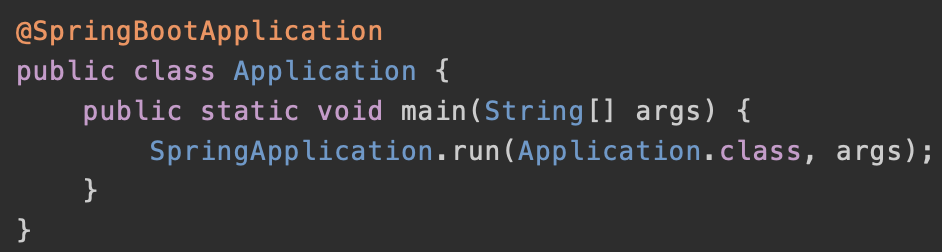
* Servlet容器（服务器）读取 web.xml
* web.xml中定义的 DispatcherServlet由容器实例化
* DispatcherServlet通过读取 WEB-INF/{servletName}-servlet.xml来创建 WebApplicationContext。最后， DispatcherServlet注册在应用程序上下文中定义的 bean

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：

SpringBoot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public static void main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：

*/\*\**

*\* War部署  
 \*/*public class ServletInitializer extends SpringBootServletInitializer {  
  
 @Override  
 protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {  
 return application.sources(Application.class);  
 }  
  
 @Override  
 public void onStartup(ServletContext servletContext) throws ServletException {  
 super.onStartup(servletContext);  
 servletContext.addListener(new HttpSessionEventPublisher());  
 }  
}

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。

#### 3.6 打包和部署

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