1. 从spring-boot-starter-parent 项目继承，并将依赖项声明为一个或多个“Starters”
2. @SpringBootApplication annotation通常放在你的主类上，它隐式地为某些项目定义了一个基本的“搜索包”
3. 如果您不想使用 @SpringBootApplication ，则 @EnableAutoConfiguration 和 @ComponentScan 注释将定义该行为，以便您也可以使用该注释
4. Spring Boot支持基于Java的配置。 虽然可以将SpringApplication 与XML源一起使用，但我们通常建议您的主要源是一个@Configuration 类。 通常，定义main 方法的类作为主要的@Configuration 是一个很好的候选@Configuration 。
5. 不需要的特定自动配置类正在应用，则可以使用 @EnableAutoConfiguration 的exclude属性来禁用它们@EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class})
6. @SpringBootApplication 注释是相当于使用 @Configuration ， @EnableAutoConfiguration 和 @ComponentScan 与他们的默认属性
7. 则可以使用 java -jar 运行应用程序，如以下示例所示：$ java -jar target/myapplication-0.0.1-SNAPSHOT.jar
8. 也可以在启用远程调试支持的情况下运行打包的应用程序。

$ java -Xdebug -Xrunjdwp:server=y,transport=dt\_socket,address=8000,suspend=n \

-jar target/myapplication-0.0.1-SNAPSHOT.jar

1. Spring Boot Maven插件包含一个可用于快速编译和运行应用程序的run 目标。mvn spring-boot:run
2. 每当类路径上的文件发生更改时，使用 spring-boot-devtools 自动重新启动
3. Spri n g Boot提供的重启技术通过使用两个类加载器来工作。 不改变的类（例如来自第三方j ar的类）被加载到基类加载器中。 您正在开发的类将加载到重启类加载器中。 当应用程序重新启动时， 重新启动类加载器将被丢弃并创建一个新类。 这种方法意味着应用程序重启通常比“冷启动”快得多
4. 某些资源不一定需要在更改时触发重新启动。 例如，可以就地编辑T h y mel eaf模板。 默认情况下，在改变资源 /META-INF/maven ， /META-INF/resources ，/resources ， /static ， /public ，或 /templates 不会触发重启但并触发l i v e rel oad 。 如果您想自定义这些排除项，可以使用 spring.devtools.restart.exclude 属性，例如，要仅排除 /static 和 /public 您可以设置以下属性
5. spring.devtools.restart.exclude=static/\*\*,public/\*\*
6. 如果您想保留这些默认设置并 添加其他排除项，请改为使用 spring.devtools.restart.additional-exclude 属性