### 1、安装Spring Boot

可以与Java开发工具一起使用，也可以作为命令行工具安装。

#### 1.1Java开发人员的安装说明

建议使用支持依赖关系管理的构建工具（例如Maven或Gradle）。

1）Maven安装：

Spring Boot与Apache Maven 3.3+版本兼容。

许多操作系统上，Maven可以与程序包管理器一起安装。

如果使用OSX Homebrew，brew install maven。

Ubuntu用户可以运行sudo apt-get install maven。

具有Chocolatey的Windows用户使用choco install maven从提升的提示符下运行。

Spring Boot依赖项使用org.springframework.boot groupId。通常，Maven POM文件从spring-boot-starter-parent项目继承，并声明对一个或多个[“启动器”的](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html#using-boot-starter)依赖关系。Spring Boot还提供了一个可选的[Maven插件](https://docs.spring.io/spring-boot/docs/current/reference/html/build-tool-plugins-maven-plugin.html)来创建可执行jar。

2）Gradle安装：

Spring Boot与Gradle 4.4+兼容。

可以使用来声明Spring Boot依赖项org.springframework.boot group。通常，您的项目声明对一个或多个“启动器”的依赖。Spring Boot提供了一个有用的Gradle插件，可用于简化依赖项声明和创建可执行jar。

Gradle Wrapper

当您需要构建项目时，Gradle包装器提供了一种“获取”Gradle的好方法。这是一个小的脚本和库，您随代码一起提交以引导构建过程。有关详细信息，请参见[docs.gradle.org/4.2.1/userguide/gradle\_wrapper.html](https://docs.gradle.org/4.2.1/userguide/gradle_wrapper.html)。

#### 1.2安装Spring Boot CLI

Spring Boot CLI（命令行界面）是一个命令行工具，可用于快速使用Spring进行原型设计。它可以运行Groovy脚本，这意味着具有类似Java的熟悉语法，而没有太多样板代码。

无需使用CLI即可与Spring Boot一起使用，但这绝对是使Spring应用程序启动的最快方法。

### 2、从较早版本的Spring Boot升级

如果要从1.xSpring Boot发行版进行升级，请查看[项目Wiki上的“迁移指南”，](https://github.com/spring-projects/spring-boot/wiki/Spring-Boot-2.0-Migration-Guide)其中提供了详细的升级说明。还请检查[“发行说明”](https://github.com/spring-projects/spring-boot/wiki)以获取每个发行版的“新功能和值得注意的功能”列表。

升级到新功能版本时，某些属性可能已被重命名或删除。Spring Boot提供了一种在启动时分析应用程序环境并打印诊断的方法，而且还可以在运行时为您临时迁移属性。要启用该功能，请将以下依赖项添加到您的项目中：

<dependency>

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

<artifactId> spring-boot-properties-migrator </ artifactId>

<scope>运行时</ scope>

</ dependency>

较晚添加到环境的属性（例如使用时@PropertySource）将不被考虑。

[注意]

迁移完成后，请确保从项目的依赖项中删除此模块。