

毕业设计(论文)

英 文 翻 译

学 院 通信与信息工程学院

专业及班级  电信1502班

姓 名  康旺

学 号  1507050227

指 导 教 师 黄健

日 期 2019年4月19日

# 第一部分.springboot文档

本节简要概述SpringBoot参考文档。它是其它部分的概述。

### 关于这篇文档

这个springBoot参考指南可以作为[HTML](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)、[PDF](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/pdf/spring-boot-reference.pdf" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)、[EPUB](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/epub/spring-boot-reference.epub" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)，最新版在复制在[docs.spring.io/spring-boot/docs/current/reference](https://docs.spring.io/spring-boot/docs/current/reference" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)上.这个的副本文档可以让你自己来用和分享给其它人，前提就是你不收取任何费用，并进一步规定每一个副本需要有以下内容，版权公告，无论是以印刷品或者是电子版本。

### **获取帮助**

如果在使用中遇到问题，我们可以获取帮助，尝试这个处理文档，它可以为一些常见的问题提供解决。学习spring基础，springboot构建在许多其它的spring项目上。查看spring.io网站，获取丰富的参考文档，如果从spring开始，那就尝试其中的一个指南。问一个问题，我们查看[stackoverflow.com](https://stackoverflow.com/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)上更多的springboot问题，报告springboot的bug可以在[github.com/spring-projects/spring-boot/issues](https://github.com/spring-projects/spring-boot/issues" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

|  |
| --- |
| IMG_256 |
| 所有的springboot开源，包括文档，如果你发现问题，如果你想改进它，请参与其中。 |

### **第一步**

一般的，如果你是开始使用springBoot或Spring,从以下主题开始，从头开始：概述|要求|安装

辅助：部分1|部分2

运行你的案例：部分1|部分2

### **让SpringBoot工作**

# 第二部分：开始

如果你开始使用springboot或者spring，就开始读这一部分，它基本回答了是什么？怎么样和为什么？它包括springboot的介绍，以及安装，然后我们通过构建你的第一个springboot应用程序，并讨论一些核心原则。

1. **介绍spring**

Springboot可以容易的单独构建以及生产级的基础可以运行的应用程序。我们可以对spring平台的看法和第三方库以至于你小题大做。许多的SpringBoot应用程序只需要非常小的spring配置。你可以使用springboot去创建一个java应用程序，通过java -jar或者传统的war部署。我们也提供了在线工具命令spring脚本运行。我们的目标是：为所有的spring提供一个极快、广泛的入门体验。提供一系列大型项目通用的非功能特性（如嵌入式服务器、安全性、度量、运行状况检查和外部化配置）。绝对没有代码生成和xml配置。

1. **系统要求**

Spring Boot 2.1.3.RELEASE要求java8并且兼容java 11，spring框架5.1.5.RELEASE或者更高的要求。以下构建工具显示提供显示的支持。

* 1. **Servlet容器**

您还可以将Spring引导应用程序部署到任何与servlet 3.1+兼容的容器中。

1. **安装SpringBoot**

Spring Bug可用于“经典”Java开发工具或安装为命令行工具。无论哪种方式，你都需要Java SDK V1.8或更高版本。在开始之前，您应该使用以下命令检查当前的Java安装：

$ java -version

如果你有新的java项目开发或者你想实验springBoot，你可以首先去尝试 [Spring Boot CLI](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-installing-the-cli" \o "10.2 Installing the Spring Boot CLI) ，否则，你就阅读这个经典的介绍说明。

3.1java开发者的安装说明

你可以使用与任何标准Java库相同的方式使用Spring Boot。这样做，包括合适的spring-boot-\*.jar文件在你的classpath.springBoot不要求任何特殊的工具整合.因此，你可以使用任何IDE或者文本编辑器，哪儿没有任何特殊关于springboot应用程序.因此，可以像任何其他Java程序一样运行和调试Spring启动应用程序。尽管你可以复制springboot jars，我们通常建议您使用支持依赖性管理的构建工具（如Maven或Gradle）。

* 1. **Maven安装**

SpringBoot与Apache Maven 3.3或更高版本兼容。如果你没有准备好一个Maven，你可以按照下面介绍通过[maven.apache.org](https://maven.apache.org/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

|  |
| --- |
| IMG_257 |
| 在许多的操作系统上，Maven可以安装通过包管理，如果你用 OSX Home brew,尝试brew安装maven，Ubuntu用户可以运行 sudo apt-get install maven.windows用户可以运行choco安装maven通过管理员运行choco install maven。 |

SpringBoot依赖通过使用org.springframework.boot 组，通常，您的maven pom文件继承自spring boot starter父项项目，并声明对一个或多个“starter”的依赖关系。SpringBoot还提供了一个可选的Maven插件来创建可执行JAR。

* 1. **Gradle安装**

springboot与Gradle 4.4及更高版本兼容。如果没有安装Gradle，可以按照gradle.org的说明进行操作。可以使用org.springframework.boot组来声明Spring引导依赖项。通常，您的项目声明依赖于一个或多个“starters”。SpringBoot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。

这个Gradle Wrapper提供了一个很好的方式获得Gradle，当你需要构建一个项目，它是一个小的脚本和库，它可以和代码一起发布，查看细节:[docs.gradle.org/4.2.1/userguide/gradle\_wrapper.html](https://docs.gradle.org/4.2.1/userguide/gradle_wrapper.html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) 。有关Spring Boot和Gradle入门的更多详细信息，请参阅Gradle插件参考指南的入门部分。

* 1. **安装springboot CLI**

Springboot CLI（命令行界面）是一个命令行工具，您可以使用它快速生成Spring原型。它允许你运行Groovy脚本，这意味着你有一个熟悉的类似Java的语法，而没有太多的样板代码。在springboot你不需要使用CLI去工作，但是这个一个运行springboot的方法。

* 1. **Manual 安装**

你可以下载spring CLI从spring软件仓库：

[spring-boot-cli-2.1.3.RELEASE-bin.zip](https://repo.spring.io/release/org/springframework/boot/spring-boot-cli/2.1.3.RELEASE/spring-boot-cli-2.1.3.RELEASE-bin.zip" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)

[spring-boot-cli-2.1.3.RELEASE-bin.tar.gz](https://repo.spring.io/release/org/springframework/boot/spring-boot-cli/2.1.3.RELEASE/spring-boot-cli-2.1.3.RELEASE-bin.tar.gz" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)

还提供了快照分发。第一次下载，安装INSTALL.txt介绍通过未打包的类库。总之，在.zip文件的bin/目录中有一个spring脚本（spring.bat for windows）。或者，您可以使用javaJAR与.jar文件（脚本帮助您确保正确设置类路径）。

* 1. **安装通过SDKMAN**

SDKMAN！（软件开发工具包管理器）可用于管理各种二进制的多个版本的SDK，包括groovy和spring boot cli。获取SDKMAN！从sdkman.io使用以下命令安装Springboot：

|  |
| --- |
| $ sdk install springboot  $ spring --version  Spring Boot v2.1.3.RELEASE |

如果您为CLI开发功能并希望轻松访问构建的版本，请使用以下命令：

|  |
| --- |
| $sdk install springboot dev /path/to/spring-boot/spring-boot-cli/target/spring-boot-cli-2.1.3.RELEASE-bin/spring-2.1.3.RELEASE/  $ sdk default springboot dev  $ spring --version  Spring CLI v2.1.3.RELEASE |

前面的说明安装了一个为dev实例的spring本地实例。它指向您的目标构建位置，因此每次您重建Spring Boot时，Spring都是最新的。你可以看下面的命令：

|  |
| --- |
| $ sdk ls springboot  Available Springboot Versions  > + dev  \* 2.1.3.RELEASE  + - local version  \* - installed  > - currently in use |

* 1. **OSX Homebrew安装**

如果你在Mac可以使用 [Homebrew](https://brew.sh/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top),，你可以安装一个springboot CLI通过该执行以下命令：

|  |
| --- |
| $ brew tap pivotal/tap  $ brew install springboot  [Homebrew](https://brew.sh/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)安装在 /usr/local/bin. |

* 1. **MacPorts安装**

如果你在mac上使用MacPorts，你可以安装SpringBootCLI通过使用下面的命令。

|  |
| --- |
| $ sudo port install spring-boot-cli |

1. **开发你的第一个springboot应用**

这一部分介绍如何开发一个简单的Helloworld项目程序,突出显示Spring Boot的一些关键功能的Web应用程序。我们可以使用Maven构建这个项目，许多的IDES支持它。

|  |
| --- |
| $ java -version  java version "1.8.0\_102"  Java(TM) SE Runtime Environment (build 1.8.0\_102-b14)  Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)  $ mvn -v  Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T14:33:14-04:00)  Maven home: /usr/local/Cellar/maven/3.3.9/libexec  Java version: 1.8.0\_102, vendor: Oracle Corporation |

* 1. **Creating the POM**
     1. **创建一个POM**

我们需要创建一个Maven的pom.xml文件，这个pom.xml文件位我们来构建项目，打开你喜欢的文本编译工具并且加入下面：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.example</groupId>  <artifactId>myproject</artifactId>  <version>0.0.1-SNAPSHOT</version>  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.1.3.RELEASE</version>  </parent>  <!-- Additional lines to be added here... -->  </project> |

前面的清单提供一个有效的构建。你可以通过运行mvn包来测试它（警告：目前，您可以忽略“jar将为空-没有标记要包含的内容！”。

|  |
| --- |
| IMG_267 |
| 在这一点，你可以引入项目到你的IDE(许多的javaIDE支持maven构建。).为了方便起见，这个例子我们继续去使用一个文本编译器 |

* + 1. **增加classpath依赖**

Spring Boot提供了许多“Starters”让我们加入jars在你的classpath.

我们的示例应用程序已经使用了spring-boot-starter-parent在这个父类的POM部分。spring boot starter父级是一个特殊的启动程序，提供有用的maven默认值。它也提供了我们使用的默认部分。它也提供了一个依赖管理部分，以至于我们可以免去版本管理来“表示”依赖关系。

其它的“Starters”提供了可能需要的依赖，当我们在开发特定类型的应用程序时。因为我们在开发一个web应用程序，我们需要加入spring-boot-starter-web依赖，在这个之前，我们需要运行一下来查看运行状态：

|  |
| --- |
| $ mvn dependency:tree |

[INFO] com.example:myproject:jar:0.0.1-SNAPSHOT

这个mvn依赖：tree命令打印你的项目依赖项的表示。你可以看spring-boot-starter-parent提供的它自己未依赖项。修改你的pom.xml并且加入spring-boot-starter-web依赖项如下：

|  |
| --- |
| <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  </dependencies> |

如果你再次运行了mvn依赖：tree，你可以看到哪儿有一下刚加入的依赖，包括tomcat web服务器和springboot它自己。

* + 1. **写代码**

去完成我们自己的应用程序，我们需要去创建一个单例的java文件，maven从src/main/java下编译源码，以至于你需要去创建一个目录结构然后加一个文件名字为/src/main/java/Exception去验证你的代码：

|  |
| --- |
| Import org.springframework.boot.\*;  import org.springframework.boot.autoconfigure.\*;  import org.springframework.web.bind.annotation.\*;  @RestController  @EnableAutoConfiguration  public class Example {  @RequestMapping("/")  String home() {  return "Hello World!";  }  public static void main(String[] args) {  SpringApplication.run(Example.class, args);  }  } |

尽管哪里没有许多的代码，发生了许多事情，我们在下面的几个部分介绍重要的部分。

* + 1. **这个@RestController和@RequestMapping**

这个类的案例的第一个注解是@RestController，这个被称为原型注解，它对读代码的人和spring提供了一个特殊的规则，在这种情况下，我们的类是web的@Controller，以至于spring在web请求的时候会进入它。

这个@RequestMapping注解提供了一个“路由”信息，它告诉spring任何的/的Http请求都会匹配它的Home方法，这个@RestController注解告诉spring将数据的返回结果给调用者。

|  |
| --- |
| IMG_268 |
| 这个@RestController和@RequestMapping注解是Spring MVC的注解（它们不是springboot特殊的），有关详细的部分，查看spring的参考文档MVC部分 |

* + 1. **@EnableAutoConfiguration注解**

第二个基本的注解是@EnableAutoConfiguration。这个注解告诉springBoot去猜，你如何去配置Spring，你需要加入的基础jar依赖。自从spring-boot-starter-web加入Tomcat和springMVC,这个自动的配置假定你正在开发一个web应用程序和设置了spring

启动和自动配置

自动配置被设计的与“启动器”一起工作良好，但这两个概念并没有直接联系。您可以在初学者之外自由选择JAR依赖项。SpringBoot仍然尽力自动配置应用程序。

* + 1. **这个“main”方法**

这个最后一个部分是我们应用程序的主方法，这个仅仅是按照java约定的应用程序入口的一个标准方法，我们通过调用run方法去委托springboot的SpringApplication类。SpringApplication启动我们的应用程序。启动spring，然后自动配置我们的Tomcat服务器，我们通过一个案例，class作为参数去执行方法告诉SpringApplication哪一个是我们的主的spring组件，这个参数数组该可以通过任何的命令行传入。

* + 1. **运行案例**

在这个部分，你的应用程序可以运行，从你室友这个spring-boot-starter-parentPOM，你可以使用这个启动的Application运行达到可以运行的这个目标，mvn类的spring-boot:run来自于跟目录去启动这个应用程序。你可以看到类似于下面的输出：

|  |
| --- |
| $ mvn spring-boot:run  . \_\_\_\_ \_ \_\_ \_ \_  /\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \  ( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \  \\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )  ' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /  =========|\_|==============|\_\_\_/=/\_/\_/\_/  :: Spring Boot :: (v2.1.3.RELEASE)  ....... . . .  ....... . . . (log output here)  ....... . . .  ........ Started Example in 2.222 seconds (JVM running for 6.514) |

如果你打开一个浏览器去执行localhost：8080，你可以看到这个输出：Hello World!

优雅的退出这个应用程序，你可以按下ctrl-c

* + 1. **创建一个外部的jar。**

我们完成我们简单的通过创建一个完全独立的可执行JAR文件来完成我们的示例，该文件可以在生产环境中运行。可执行的jars（有时候叫“far jars”）包含了代码运行的所有jar依赖。

执行jars和java

Java不仅仅提供了一个标准的方式可以加载嵌套的jar文件（jar文件它自己包含在jar中）。这个可能会有问题如果你分发一个自包含的应用程序。

为了解决这个问题，许多的开发者使用“user”jars，一个uber jar的包包含所有的classes来自于这个应用程序的依赖的单个存档。这个方法的问题是，我们很难看出你的应用程序中包含哪些包。它也存在一个问题，如果你使用相同的包（但是又是不同的内容）。

springBoot提供了一个不同的方法，并且可以嵌套jar。

去创建一个可以执行的jar,我们需要增加一些spring-boot-maveb-plugin在我们的pom.xml中，这样做，插入下面划线的在dependenciessection：

|  |
| --- |
| <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build> |

|  |
| --- |
| IMG_269 |
| 这个spring-boot-starter-parentPOM包括<executions>配置去绑定这个包围目标，如果你没使用pom父类。你需要自己声明这个配置，有关详细信息，请查看文档。 |

保存pom.xml和运行mvn包在你的命令行中，如下：

|  |
| --- |
| $ mvn package  [INFO] Scanning for projects...  [INFO]  [INFO] --------------------------------------------  [INFO] Building myproject 0.0.1-SNAPSHOT  [INFO] --------------------------------------------  [INFO] .... ..  [INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ myproject ---  [INFO] Building jar: /Users/developer/example/spring-boot-example/target/myproject-0.0.1-SNAPSHOT.jar  [INFO]---spring-boot-maven-plugin:2.1.3.RELEASE:repackage (default) @ myproject ---  [INFO] --------------------------------------------  [INFO] BUILD SUCCESS  [INFO] -------------------------------------------- |

如果你查看target目录，你可以看到myproject-0.0.1-SNAPSHOT.jar.这个文件大概10M左右，如果你想要查看内部，你可以使用jar tvf，如下：

$ jar tvf target/myproject-0.0.1-SNAPSHOT.jar

你也可以看到一个更小的文件名字是myproject-0.0.1-SNAPSHOT.jar.original在target目录中.这个源文件的jar文件在springBoot还没有打包之前

去运行这个应用程序，使用java -jar命令，如下：

|  |
| --- |
| $ java -jar target/myproject-0.0.1-SNAPSHOT.jar  . \_\_\_\_ \_ \_\_ \_ \_  /\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \  ( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \  \\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )  ' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /  =========|\_|==============|\_\_\_/=/\_/\_/\_/  :: Spring Boot :: (v2.1.3.RELEASE)  ....... . . .  ....... . . . (log output here)  ....... . . .  ........ Started Example in 2.536 seconds (JVM running for 2.864) |

和前面一样，我们推出应用程序，按下ctrl+c。

1. **接下来读什么**

这一部分提供基础的spring知识帮助你可以写出你自己的应用程序。如果您是一个面向任务的开发人员，那么您可能希望跳到spring.io并查看一些入门指南，这些指南解决了特定的“我如何使用spring实现这一点？”“问题,我们还提供了特定于Spring引导的“如何操作”参考文档。

这个springBoot仓库也有一些可以运行的简单案例，这些简单的案例独立于代码部分（也就是说，你不需要构建其他的代码或者来运行这些实例）。

否则，下一个合乎逻辑的步骤是阅读第三部分，“使用springBoot”，如果你真的有耐心，你可以跳过并且阅读关于springBoot的特性。

第III部分. 使用springBoot

这一部分就开始详细的关于你如何使用springBoot，它包含了主题，包括构建系统，自动配置和怎样去运行你的应用程序，我们也介绍了一些springBoot的最佳实践。尽管哪儿没有任何特殊的东西关于springboot（它仅仅使得你使用了一个库），哪儿有一些建议，可以使你开发更加容易点。如果你是从springBoot开始，你在越大这个之前，不可能会阅读入门指南。

1. **构建系统**

这里强烈建议选择一个可以支持依赖管理并可以发布到“Maven Central”存储仓库中的构建系统。我们建议你选择maven或者gradle，它让springBoot与其他构建系统（比如ant）一起工作是可能的，但是他们并不是特别受支持的

* 1. **依赖管理**

每一个稳定版本springBoot提供了当前依赖支持列表，在练习中，你可以不需要提供任何版本和这些依赖在你构建的编译版本中，你可以管理springBoot。升级SpringBoot本身时，这些依赖项也会以一致的方式升级。

|  |
| --- |
| IMG_270 |
| 当你需要的时候，你仍然可以提供版本和覆盖springBoot  的推荐. |

这个计划包含了可以在springBoot中使用的所有的springBoot模块，以及完整的第三方库的列表。这个列表中所有的清单提供，可以与Maven和Gradle一起使用。

|  |
| --- |
| IMG_271 |
| 每一个发行版本的springBoot都和spring框架的一个基础版本相关联。 |

* 1. **Maven**

Maven使用者可以继承spring-boot-starter-parent项目，获得合理的默认值。父类项目提供了如下功能：

Java1.8作为默认的编译器级别。

源码是UTF-8

一个依赖管理部分，继承来自于spring-boot-dependencies pom文件，这个管理版本是一个共同的依赖，这种依赖关系管理允许您在自己的pom中使用省略的那些依赖项的<version>标记。

带有重新打包执行id的重打包目标的执行。

合理的资源拦截。

合理的插件配置（执行插件，git提交ID）

合理资源拦截在application.properties和application.yml包括配置文件（比如，application-dev.properties andapplication-dev.yml）

注意：由于applicy.properties和application.yml文件接受Spring样式占位符(${…)(​})，Maven筛选被更改为使用@.@占位符。(您可以通过设置一个Maven来覆盖它)属性。)

* + 1. **继承启动父类**

配置你的项目去继承这个spring-boot-starter-parent,将父类设置如下：

|  |
| --- |
| <!-- Inherit defaults from Spring Boot -->  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.1.3.RELEASE</version>  </parent> |

|  |
| --- |
| IMG_272 |
| 你仅仅需要其指定springBoot版本数在你的依赖上，如果你引入其他启动器，可以完全的省略版本号。 |

有了那个设置，你也可以覆盖自己的依赖去覆盖在自己的项目中重写属性。比如，去更新其他的spring数据练习，你可以增加下面在你的pom.xml。

|  |
| --- |
| <properties>  <spring-data-releasetrain.version>Fowler-SR2</spring-data-releasetrain.version>  </properties> |

|  |
| --- |
| IMG_273 |
| 检查spring-boot-dependencies以支持的属性列表。 |

# **Part I. Spring Boot Documentation**

This section provides a brief overview of Spring Boot reference documentation. It serves as a map for the rest of the document.

### **1. About the Documentation**

The Spring Boot reference guide is available as [HTML](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) [PDF](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/pdf/spring-boot-reference.pdf" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) [EPUB](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/epub/spring-boot-reference.epub" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)

The latest copy is available at [docs.spring.io/spring-boot/docs/current/reference](https://docs.spring.io/spring-boot/docs/current/reference" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

Copies of this document may be made for your own use and for distribution to others,

provided

Provided that you do not charge any fee for such copies and further provided that each copy

contains this Copyright Notice, whether distributed in print or electronically.

### **2. Getting Help**

If you have trouble with Spring Boot, we would like to help.

Try the [How-to documents](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "howto" \o "Part IX. ‘How-to’ guides). They provide solutions to the most common questions.

Learn the Spring basics. Spring Boot builds on many other Spring projects. Check

the [spring.io](https://spring.io/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) web-site for a wealth of reference documentation. If you are starting out with Spring, try one of the [guides](https://spring.io/guides" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

Ask a question. We monitor [stackoverflow.com](https://stackoverflow.com/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) for questions tagged with [spring-boot](https://stackoverflow.com/tags/spring-boot" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

Report bugs with Spring Boot at [github.com/spring-projects/spring-boot/issues](https://github.com/spring-projects/spring-boot/issues" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

|  |
| --- |
| IMG_256 |
| All of Spring Boot is open source, including the documentation. If you find problems  所有的springboot开源，包括文档，如果你发现问题  with the docs or if you want to improve them, please [get involved](https://github.com/spring-projects/spring-boot/tree/v2.1.3.RELEASE" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).  如果你想改进它，请参与其中 |

### **3. First Steps**

If you are getting started with Spring Boot or 'Spring' in general, start with [the following topics](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started" \o "Part II. Getting Started):

From scratch: [Overview](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-introducing-spring-boot" \o "8. Introducing Spring Boot) | [Requirements](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-system-requirements" \o "9. System Requirements) | [Installation](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-installing-spring-boot" \o "10. Installing Spring Boot)

Tutorial: [Part 1](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-first-application" \o "11. Developing Your First Spring Boot Application) | [Part 2](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-first-application-code" \o "11.3 Writing the Code)

Running your example: [Part 1](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-first-application-run" \o "11.4 Running the Example) | [Part 2](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-first-application-executable-jar" \o "11.5 Creating an Executable Jar)

### **4. Working with Spring Boot**

## Part II. Getting Started

If you are getting started with Spring Boot, or “Spring” in general, start by reading this section. It

answers the basic “what?”, “how?” and “why?” questions. It includes an introduction to Spring Boot, along with installation instructions. We then walk you through building your first Spring Boot application, discussing some core principles as we go.

### 1. Introducing Spring Boot

Spring Boot makes it easy to create stand-alone, production-grade Spring-based Applications that

you can run. We take an opinionated view of the Spring platform and third-party libraries, so that

you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

You can use Spring Boot to create Java applications that can be started by using java -jar or more

traditional war deployments. We also provide a command line tool that runs “spring scripts”.

Our primary goals are:

Provide a radically faster and widely accessible getting-started experience for all Spring development.

Provide a range of non-functional features that are common to large classes of projects (such as embedded servers, security, metrics, health checks, and externalized configuration).

Absolutely no code generation and no requirement for XML configuration.

### 2. System Requirements

Spring Boot 2.1.3.RELEASE requires [Java 8](https://www.java.com/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) and is compatible up to Java 11 (included). [Spring Framework 5.1.5.RELEASE](https://docs.spring.io/spring/docs/5.1.5.RELEASE/spring-framework-reference/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) or above is also required.

Explicit build support is provided for the following build tools:

#### 2.1 Servlet Containers

You can also deploy Spring Boot applications to any Servlet 3.1+ compatible container.

### 3. Installing Spring Boot

Spring Boot can be used with “classic” Java development tools or installed as a command line tool. Either way, you need [Java SDK v1.8](https://www.java.com/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) or higher. Before you begin, you should check your current Java installation by using the following command:

$ java -version

If you are new to Java development or if you want to experiment with Spring Boot, you might want to try the [Spring Boot CLI](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-installing-the-cli" \o "10.2 Installing the Spring Boot CLI) (Command Line Interface) first. Otherwise, read on for “classic” installation instructions.

3.1 Installation Instructions for the Java Developer

You can use Spring Boot in the same way as any standard Java library. To do so, include the

appropriate spring-boot-\*.jar files on your classpath. Spring Boot does not require any special

tools integration, so you can use any IDE or text editor. Also, there is nothing special about a

Spring Boot application, so you can run and debug a Spring Boot application as you would any other Java program.

Although you could copy Spring Boot jars, we generally recommend that you use a build tool that supports dependency management (such as Maven or Gradle).

##### 3.1 Maven Installation

Spring Boot is compatible with Apache Maven 3.3 or above. If you do not already have Maven installed, you can follow the instructions at [maven.apache.org](https://maven.apache.org/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

|  |
| --- |
| IMG_257 |
| On many operating systems, Maven can be installed with a package manager. If you use OSX Home brew, try brew install maven. Ubuntu users can run sudo apt-get install maven. Windows users with [Chocolatey](https://chocolatey.org/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) can run choco install maven from an elevated (administrator) prompt. |

Spring Boot dependencies use the org.springframework.boot groupId. Typically, your Maven POM file inherits from the spring-boot-starter-parent project and declares dependencies to one or more [“Starters”](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-starter" \o "13.5 Starters). Spring Boot also provides an optional [Maven plugin](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "build-tool-plugins-maven-plugin" \o "71. Spring Boot Maven Plugin) to create executable jars.

##### **3.2 Gradle Installation**

Spring Boot is compatible with Gradle 4.4 and later. If you do not already have Gradle installed, you can follow the instructions at [gradle.org](https://gradle.org/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

Spring Boot dependencies can be declared by using the org.springframework.boot group. Typically, your project declares dependencies to one or more [“Starters”](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-starter" \o "13.5 Starters). Spring Boot provides a useful [Gradle plugin](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "build-tool-plugins-gradle-plugin" \o "72. Spring Boot Gradle Plugin) that can be used to simplify dependency declarations and to create executable jars.

The Gradle Wrapper provides a nice way of “obtaining” Gradle when you need to build a project. It is a small script and library that you commit alongside your code to bootstrap the build process. See [docs.gradle.org/4.2.1/userguide/gradle\_wrapper.html](https://docs.gradle.org/4.2.1/userguide/gradle_wrapper.html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) for details.

More details on getting started with Spring Boot and Gradle can be found in the [Getting Started section](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/gradle-plugin/reference/html/" \l "getting-started" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) of the Gradle plugin’s reference guide.

#### 3.3 Installing the Spring Boot CLI

The Spring Boot CLI (Command Line Interface) is a command line tool that you can use to quickly prototype with Spring. It lets you run [Groovy](http://groovy-lang.org/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) scripts, which means that you have a familiar Java-like syntax without so much boilerplate code.

You do not need to use the CLI to work with Spring Boot, but it is definitely the quickest way to get a Spring application off the ground.

#### **3.4 Manual Installation**

You can download the Spring CLI distribution from the Spring software repository:

[spring-boot-cli-2.1.3.RELEASE-bin.zip](https://repo.spring.io/release/org/springframework/boot/spring-boot-cli/2.1.3.RELEASE/spring-boot-cli-2.1.3.RELEASE-bin.zip" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)

[spring-boot-cli-2.1.3.RELEASE-bin.tar.gz](https://repo.spring.io/release/org/springframework/boot/spring-boot-cli/2.1.3.RELEASE/spring-boot-cli-2.1.3.RELEASE-bin.tar.gz" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)

Cutting edge [snapshot distributions](https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-cli/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) are also available.

Once downloaded, follow the [INSTALL.txt](https://raw.github.com/spring-projects/spring-boot/v2.1.3.RELEASE/spring-boot-project/spring-boot-cli/src/main/content/INSTALL.txt" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) instructions from the unpacked archive. In summary, there is a spring script (spring.bat for Windows) in a bin/ directory in the .zip file. Alternatively, you can use java -jar with the .jar file (the script helps you to be sure that the classpath is set correctly).

#### **3.5 Installation with SDKMAN!**

SDKMAN! (The Software Development Kit Manager) can be used for managing multiple versions of various binary SDKs, including Groovy and the Spring Boot CLI. Get SDKMAN! from [sdkman.io](http://sdkman.io/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) and install Spring Boot by using the following commands:

$ sdk install springboot

$ spring --version

Spring Boot v2.1.3.RELEASE

If you develop features for the CLI and want easy access to the version you built, use the following commands:

$sdk install springboot dev /path/to/spring-boot/spring-boot-cli/target/spring-boot-cli-2.1.3.RELEASE-bin/spring-2.1.3.RELEASE/

$ sdk default springboot dev

$ spring --version

Spring CLI v2.1.3.RELEASE

The preceding instructions install a local instance of spring called the dev instance. It points at your target build location, so every time you rebuild Spring Boot,spring is up-to-date.

You can see it by running the following command:

$ sdk ls springboot

Available Springboot Versions

> + dev

\* 2.1.3.RELEASE

+ - local version

\* - installed

> - currently in use

================================================================================

#### **3.6 OSX Homebrew Installation**

If you are on a Mac and use [Homebrew](https://brew.sh/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top), you can install the Spring Boot CLI by using the following commands:

$ brew tap pivotal/tap

$ brew install springboot

Homebrew installs spring to /usr/local/bin.

#### **3.7 MacPorts Installation**

If you are on a Mac and use [MacPorts](https://www.macports.org/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top), you can install the Spring Boot CLI by using the following command:

$ sudo port install spring-boot-cli

### 4. Developing Your First Spring Boot Application

This section describes how to develop a simple “Hello World!” web application that highlights

some of Spring Boot’s key features. We use Maven to build this project, since most IDEs support it.

|  |
| --- |
| IMG_265 |
| The [spring.io](https://spring.io/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) web site contains many “Getting Started” [guides](https://spring.io/guides" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) that use Spring Boot. If you need to solve a specific problem, check there first.  You can shortcut the steps below by going to [start.spring.io](https://start.spring.io/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) and choosing the "Web" starter from the dependencies searcher. Doing so generates a new project structure so that you can [start coding right away](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started-first-application-code" \o "11.3 Writing the Code). Check the [Spring Initializr documentation](https://docs.spring.io/initializr/docs/current/reference/htmlsingle/" \l "user-guide" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) for more details. |

Before we begin, open a terminal and run the following commands to ensure that you have valid versions of Java and Maven installed:

$ java -version

java version "1.8.0\_102"

Java(TM) SE Runtime Environment (build 1.8.0\_102-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)

$ mvn -v

Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T14:33:14-04:00)

Maven home: /usr/local/Cellar/maven/3.3.9/libexec

Java version: 1.8.0\_102, vendor: Oracle Corporation

|  |
| --- |
| IMG_266 |
| This sample needs to be created in its own folder. Subsequent instructions assume that you have created a suitable folder and that it is your current directory. |

#### 4.1 Creating the POM

We need to start by creating a Maven pom.xml file. The pom.xml is the recipe that is used to build your project. Open your favorite text editor and add the following:

<?xml version="1.0" encoding="UTF-8"?><project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>myproject</artifactId>

<version>0.0.1-SNAPSHOT</version>

<parent>

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

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

<version>2.1.3.RELEASE</version>

</parent>

<!-- Additional lines to be added here... -->

</project>

The preceding listing should give you a working build. You can test it by running mvn package (for now, you can ignore the “jar will be empty - no content was marked for inclusion!” warning).

|  |
| --- |
| IMG_267 |
| At this point, you could import the project into an IDE (most modern Java IDEs include built-in support for Maven). For simplicity, we continue to use a plain text editor for this example. |

#### **4.2 Adding Classpath Dependencies**

Spring Boot provides a number of “Starters” that let you add jars to your classpath. Our sample application has already used spring-boot-starter-parent in the parent section of the POM. The spring-boot-starter-parent is a special starter that provides useful Maven defaults. It also provides a [dependency-management](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-dependency-management" \o "13.1 Dependency Management)section so that you can omit version tags for “blessed” dependencies.

Other “Starters” provide dependencies that you are likely to need when developing a specific type of application. Since we are developing a web application, we add aspring-boot-starter-web dependency. Before that, we can look at what we currently have by running the following command:

$ mvn dependency:tree

[INFO] com.example:myproject:jar:0.0.1-SNAPSHOT

The mvn dependency:tree command prints a tree representation of your project dependencies. You can see that spring-boot-starter-parent provides no dependencies by itself. To add the necessary dependencies, edit your pom.xml and add the spring-boot-starter-web dependency immediately below the parentsection:

<dependencies>

<dependency>

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

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

</dependency></dependencies>

If you run mvn dependency:tree again, you see that there are now a number of additional dependencies, including the Tomcat web server and Spring Boot itself.

#### 4.3 Writing the Code

To finish our application, we need to create a single Java file. By default, Maven compiles sources from src/main/java, so you need to create that folder structure and then add a file named src/main/java/Example.java to contain the following code:

import org.springframework.boot.\*;import org.springframework.boot.autoconfigure.\*;import org.springframework.web.bind.annotation.\*;

@RestController@EnableAutoConfigurationpublic class Example {

@RequestMapping("/")

String home() {

return "Hello World!";

}

public static void main(String[] args) {

SpringApplication.run(Example.class, args);

}

}

Although there is not much code here, quite a lot is going on. We step through the important parts in the next few sections.

#### 4.4 The @RestController and @RequestMapping Annotations

The first annotation on our Example class is @RestController. This is known as a stereotype annotation. It provides hints for people reading the code and for Spring that the class plays a specific role. In this case, our class is a web @Controller, so Spring considers it when handling incoming web requests.

The @RequestMapping annotation provides “routing” information. It tells Spring that any HTTP request with the / path should be mapped to the home method. The@RestController annotation tells Spring to render the resulting string directly back to the caller.

|  |
| --- |
| IMG_268 |
| The @RestController and @RequestMapping annotations are Spring MVC annotations. (They are not specific to Spring Boot.) See the [MVC section](https://docs.spring.io/spring/docs/5.1.5.RELEASE/spring-framework-reference/web.html" \l "mvc" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) in the Spring Reference Documentation for more details. |

#### 4.5 The @EnableAutoConfiguration Annotation

The second class-level annotation is @EnableAutoConfiguration. This annotation tells Spring Boot to “guess” how you want to configure Spring, based on the jar dependencies that you have added. Since spring-boot-starter-web added Tomcat and Spring MVC, the auto-configuration assumes that you are developing a web application and sets up Spring accordingly.

Starters and Auto-configuration

Auto-configuration is designed to work well with “Starters”, but the two concepts are not directly tied. You are free to pick and choose jar dependencies outside of the starters. Spring Boot still does its best to auto-configure your application.

#### 4.6 The “main” Method

The final part of our application is the main method. This is just a standard method that follows the Java convention for an application entry point. Our main method delegates to Spring Boot’s SpringApplication class by calling run. SpringApplication bootstraps our application, starting Spring, which, in turn, starts the auto-configured Tomcat web server. We need to pass Example.class as an argument to the run method to tell SpringApplication which is the primary Spring component. The args array is also passed through to expose any command-line arguments.

#### 4.7 Running the Example

At this point, your application should work. Since you used the spring-boot-starter-parent POM, you have a useful run goal that you can use to start the application. Type mvn spring-boot:run from the root project directory to start the application. You should see output similar to the following:

$ mvn spring-boot:run

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v2.1.3.RELEASE)

....... . . .

....... . . . (log output here)

....... . . .

........ Started Example in 2.222 seconds (JVM running for 6.514)

If you open a web browser to [localhost:8080](http://localhost:8080/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top), you should see the following output:

Hello World!

To gracefully exit the application, press ctrl-c.

#### 4.8 Creating an Executable Jar

We finish our example by creating a completely self-contained executable jar file that we could run in production. Executable jars (sometimes called “fat jars”) are archives containing your compiled classes along with all of the jar dependencies that your code needs to run.

Executable jars and Java

Java does not provide a standard way to load nested jar files (jar files that are themselves contained within a jar). This can be problematic if you are looking to distribute a self-contained application.

To solve this problem, many developers use “uber” jars. An uber jar packages all the classes from all the application’s dependencies into a single archive. The problem with this approach is that it becomes hard to see which libraries are in your application. It can also be problematic if the same filename is used (but with different content) in multiple jars.

Spring Boot takes a [different approach](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "executable-jar" \o "Appendix E. The Executable Jar Format) and lets you actually nest jars directly.

To create an executable jar, we need to add the spring-boot-maven-plugin to our pom.xml. To do so, insert the following lines just below the dependenciessection:

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins></build>

|  |
| --- |
| IMG_269 |
| The spring-boot-starter-parent POM includes <executions> configuration to bind the repackage goal. If you do not use the parent POM, you need to declare this configuration yourself. See the [plugin documentation](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/maven-plugin/usage.html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) for details. |

Save your pom.xml and run mvn package from the command line, as follows:

$ mvn package

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building myproject 0.0.1-SNAPSHOT

[INFO] ------------------------------------------------------------------------

[INFO] .... ..

[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ myproject ---

[INFO] Building jar: /Users/developer/example/spring-boot-example/target/myproject-0.0.1-SNAPSHOT.jar

[INFO]

[INFO] --- spring-boot-maven-plugin:2.1.3.RELEASE:repackage (default) @ myproject ---

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

If you look in the target directory, you should see myproject-0.0.1-SNAPSHOT.jar. The file should be around 10 MB in size. If you want to peek inside, you can use jar tvf, as follows:

$ jar tvf target/myproject-0.0.1-SNAPSHOT.jar

You should also see a much smaller file named myproject-0.0.1-SNAPSHOT.jar.original in the target directory. This is the original jar file that Maven created before it was repackaged by Spring Boot.

To run that application, use the java -jar command, as follows:

$ java -jar target/myproject-0.0.1-SNAPSHOT.jar

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v2.1.3.RELEASE)

....... . . .

....... . . . (log output here)

....... . . .

........ Started Example in 2.536 seconds (JVM running for 2.864)

As before, to exit the application, press ctrl-c.

### 5. What to Read Next

Hopefully, this section provided some of the Spring Boot basics and got you on your way to writing your own applications. If you are a task-oriented type of developer, you might want to jump over to [spring.io](https://spring.io/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) and check out some of the [getting started](https://spring.io/guides/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) guides that solve specific “How do I do that with Spring?” problems. We also have Spring Boot-specific “[How-to](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "howto" \o "Part IX. ‘How-to’ guides)” reference documentation.

The [Spring Boot repository](https://github.com/spring-projects/spring-boot" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) also has a [bunch of samples](https://github.com/spring-projects/spring-boot/tree/v2.1.3.RELEASE/spring-boot-samples" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) you can run. The samples are independent of the rest of the code (that is, you do not need to build the rest to run or use the samples).

Otherwise, the next logical step is to read [Part III, “Using Spring Boot”](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot" \o "Part III. Using Spring Boot). If you are really impatient, you could also jump ahead and read about [Spring Boot features](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "boot-features" \o "Part IV. Spring Boot features).

Part III. Using Spring Boot

This section goes into more detail about how you should use Spring Boot. It covers topics such as build systems, auto-configuration, and how to run your applications. We also cover some Spring Boot best practices. Although there is nothing particularly special about Spring Boot (it is just another library that you can consume), there are a few recommendations that, when followed, make your development process a little easier.

If you are starting out with Spring Boot, you should probably read the [Getting Started](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "getting-started" \o "Part II. Getting Started) guide before diving into this section.

### 6. Build Systems

It is strongly recommended that you choose a build system that supports [dependency management](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-dependency-management" \o "13.1 Dependency Management) and that can consume artifacts published to the “Maven Central” repository. We would recommend that you choose Maven or Gradle. It is possible to get Spring Boot to work with other build systems (Ant, for example), but they are not particularly well supported.

#### 6.1 Dependency Management

Each release of Spring Boot provides a curated list of dependencies that it supports. In practice, you do not need to provide a version for any of these dependencies in your build configuration, as Spring Boot manages that for you. When you upgrade Spring Boot itself, these dependencies are upgraded as well in a consistent way.

|  |
| --- |
| IMG_270 |
| You can still specify a version and override Spring Boot’s recommendations if you need to do so. |

The curated list contains all the spring modules that you can use with Spring Boot as well as a refined list of third party libraries. The list is available as a standard [Bills of Materials (spring-boot-dependencies)](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-maven-without-a-parent" \o "13.2.2 Using Spring Boot without the Parent POM) that can be used with both [Maven](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-maven-parent-pom" \o "13.2.1 Inheriting the Starter Parent) and [Gradle](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-gradle" \o "13.3 Gradle).

|  |
| --- |
| IMG_271 |
| Each release of Spring Boot is associated with a base version of the Spring Framework. We highly recommend that you not specify its version. |

#### **6.2 Maven**

Maven users can inherit from the spring-boot-starter-parent project to obtain sensible defaults. The parent project provides the following features:

Java 1.8 as the default compiler level.

UTF-8 source encoding.

A [Dependency Management section](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/" \l "using-boot-dependency-management" \o "13.1 Dependency Management), inherited from the spring-boot-dependencies pom, that manages the versions of common dependencies. This dependency management lets you omit <version> tags for those dependencies when used in your own pom.

An execution of the [repackage goal](https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/maven-plugin/repackage-mojo.html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) with a repackage execution id.

Sensible [resource filtering](https://maven.apache.org/plugins/maven-resources-plugin/examples/filter.html" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top).

Sensible plugin configuration ([exec plugin](http://www.mojohaus.org/exec-maven-plugin/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top), [Git commit ID](https://github.com/ktoso/maven-git-commit-id-plugin" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top), and [shade](https://maven.apache.org/plugins/maven-shade-plugin/" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top)).

Sensible resource filtering for application.properties and application.yml including profile-specific files (for example, application-dev.properties andapplication-dev.yml)

Note that, since the application.properties and application.yml files accept Spring style placeholders (${…​}), the Maven filtering is changed to use @..@placeholders. (You can override that by setting a Maven property called resource.delimiter.)

#### 6.2.1 Inheriting the Starter Parent

To configure your project to inherit from the spring-boot-starter-parent, set the parent as follows:

<!-- Inherit defaults from Spring Boot --><parent>

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

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

<version>2.1.3.RELEASE</version></parent>

|  |
| --- |
| IMG_272 |
| You should need to specify only the Spring Boot version number on this dependency. If you import additional starters, you can safely omit the version number. |

With that setup, you can also override individual dependencies by overriding a property in your own project. For instance, to upgrade to another Spring Data release train, you would add the following to your pom.xml:

<properties>

<spring-data-releasetrain.version>Fowler-SR2</spring-data-releasetrain.version></properties>

|  |
| --- |
| IMG_273 |
| Check the [spring-boot-dependencies pom](https://github.com/spring-projects/spring-boot/tree/v2.1.3.RELEASE/spring-boot-project/spring-boot-dependencies/pom.xml" \t "https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/_top) for a list of supported properties. |