建议选择一个支持依赖关系管理并且可以使用发布到“Maven Central”存储库的工件的构建系统。建议选择Maven或Gradle。可以使用Spring Boot与其他构建系统（例如，Ant）一起使用，但是它们并没有得到很好的支持。

### 1、依赖管理

每个Spring Boot版本都提供了它所支持的依赖关系的精选列表。实际上，我们不需要为构建配置中的所有这些依赖项提供版本，因为Spring Boot会为我们管理该版本。当升级Spring Boot本身时，这些依赖项也会以一致方式升级。

注意：仍然可以指定版本，并在需要时覆盖Spring Boot的建议。

精选列表包含可与Spring Boot一起使用的所有spring模块以及完善的第三方库列表。该列表作为可与Maven和Gradle一起使用的标准物料清单（spring-boot-dependencies）提供。

警告：Spring Boot的每个发行版本都与Spring Framework的基本版本相关联。

建议不要指定其他版本。

### 2、Maven

Maven用户可以从spring-boot-starter-parent项目继承来获得合理的默认值。

父项目提供以下特性：

* Java 1.8是默认的编译器级别。
* UTF-8 source encoding。
* 从spring-boot-dependencies pom继承的Dependency Management部分，它管理公共依赖项的版本。当在自己的pom中使用这些依赖关系时，可以为这些依赖关系省略<version>标记。
* 使用重新打包执行ID执行重新打包目标。
* 合理的资源过滤。
* 合理的插件配置（exec插件，Git commit ID和shade）。
* 对于application.xml和application.yml进行合理的资源过滤，包括特定于配置文件的文件（例如，application-dev.properties和application-dev.yml）。

注意：由于application.properties和application.yml文件接收Spring样式的占

位符（${...}），因此Maven过滤已更改为使用@..@占位符。可以通过设置名为resource.delimiter的Maven属性来覆盖它。

#### 2.1继承入门级父项目

要将项目配置为从spring-boot-starter-parent继承，请设置父项：

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

<parent>

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

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

<version>2.1.8.RELEASE</version>

</parent>

注意：只需要为此依赖项指定Spring Boot版本号。如果导入其他启动器，则可以安全地省略版本号。

使用该设置还可以通过覆盖自己项目中的属性来覆盖各个依赖项。例如，要升级到另一个Spring Data发布系列，可以将以下内容添加到pom.xml中：

<properties>

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

</properties>

提示：检查spring-boot-dependencies pom获得支持的属性列表。

#### 2.2在没有父POM的情况下使用Spring Boot

如果不想使用spring-boot-starter-parent，仍然可以通过使用scope=import依赖项来保持依赖管理（而不是插件管理）的优势，如下所示：

<dependencyManagement>

<dependencies>

<dependency>

<!-- Import dependency management from Spring Boot -->

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

<artifactId>spring-boot-dependencies</artifactId>

<version>2.1.8.RELEASE</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

如上所述，前面的示例设置不允许您使用属性来覆盖各个依赖项。为了获得相同的结果，需要在spring-boot-dependencies条目之前的项目的dependencyManagement中添加一个条目。例如，要升级到另一个Spring Data发布系列，可以将以下元素添加到pom.xml中：

<dependencyManagement>

<dependencies>

<!—覆盖Spring Boot提供的Spring Data -->

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-releasetrain</artifactId>

<version>Fowler-SR2</version>

<type>pom</type>

<scope>import</scope>

</dependency>

<dependency>

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

<artifactId>spring-boot-dependencies</artifactId>

<version>2.1.8.RELEASE</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

注意：在前面的示例中，我们指定了一个BOM（物料清单），但是任何依赖类型都可以以相同的方式覆盖。

#### 2.3使用Spring Boot Maven Plugin

Spring Boot包含一个Maven插件，可以将项目打包为可执行jar。如果要使用插件，请将其添加到<plugins>部分。

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

注意：如果使用Spring Boot 启动器的父pom，则只需要添加插件，无需对其进行配置。

### 3、Gradle

要了解有关将Spring Boot与Gradle结合使用的信息，请参阅Spring Boot的Gradle插件文档：

* Reference ([HTML](https://docs.spring.io/spring-boot/docs/2.1.8.RELEASE/gradle-plugin/reference/html) and [PDF](https://docs.spring.io/spring-boot/docs/2.1.8.RELEASE/gradle-plugin/reference/pdf/spring-boot-gradle-plugin-reference.pdf))
* [API](https://docs.spring.io/spring-boot/docs/2.1.8.RELEASE/gradle-plugin/reference/html)

### 4、Ant

可以使用Apache Ant + lvy构建Spring Boot项目。Spring-boot-antlib“AntLib”模块也可用于帮助Ant创建可执行jar。

为了声明依赖关系，一个典型的ivy.xml文件看起来类似于以下示例：

<ivy-module version="2.0">

<info organisation="org.springframework.boot" module="spring-boot-sample-ant" />

<configurations>

<conf name="compile" description="everything needed to compile this module" />

<conf name="runtime" extends="compile" description="everything needed to run this module" />

</configurations>

<dependencies>

<dependency org="org.springframework.boot" name="spring-boot-starter"

rev="${spring-boot.version}" conf="compile" />

</dependencies>

</ivy-module>

一个典型的build.xml类似于以下示例：

<project

xmlns:ivy="antlib:org.apache.ivy.ant"

xmlns:spring-boot="antlib:org.springframework.boot.ant"

name="myapp" default="build">

<property name="spring-boot.version" value="2.1.8.RELEASE" />

<target name="resolve" description="--> retrieve dependencies with ivy">

<ivy:retrieve pattern="lib/[conf]/[artifact]-[type]-[revision].[ext]" />

</target>

<target name="classpaths" depends="resolve">

<path id="compile.classpath">

<fileset dir="lib/compile" includes="\*.jar" />

</path>

</target>

<target name="init" depends="classpaths">

<mkdir dir="build/classes" />

</target>

<target name="compile" depends="init" description="compile">

<javac srcdir="src/main/java" destdir="build/classes" classpathref="compile.classpath" />

</target>

<target name="build" depends="compile">

<spring-boot:exejar destfile="build/myapp.jar" classes="build/classes">

<spring-boot:lib>

<fileset dir="lib/runtime" />

</spring-boot:lib>

</spring-boot:exejar>

</target>

</project>

### 5、Starters（起步依赖列表）

起步依赖列表是一组包含在应用程序内的依赖项描述符。可以一站式购买所需的Spring和相关技术，不必搜索同样的代码复制粘贴依赖项描述符。例如，如果要开始使用Spring和JPA进行数据库访问，请在项目中包括spring-boot-starter-data-jpa依赖项。

起步依赖包含许多启动项目（并快速运行）所需的依赖项，以及一组一致且受支持的托管传递依赖项。

起步依赖的名字叫什么？

Spring-boot-starter-\*，其中\*是特定类型的应用程序。这种命名结构旨在帮助我们寻找起步依赖。许多IDE中集成的Maven使我们可以按名称搜索依赖项。

如“创建自己的起步依赖”部分所述，第三方起步依赖不应以spring-boot开头，因为它是保留给Spring Boot官方构件的。第三方起步依赖通常以项目名称开头。例如，一个名为thirdpartyproject通常会被命名为thirdpartyproject-spring-boot-starter.

Spring Boot在org.springframework.boot组下提供了以下应用程序starters：

| **名称** | **描述** |
| --- | --- |
| spring-boot-starter | 核心入门工具，包括自动配置支持，日志记录和YAML |
| spring-boot-starter-activemq | 使用Apache ActiveMQ的JMS消息传递入门 |
| spring-boot-starter-amqp | 使用Spring AMQP和Rabbit MQ的入门 |
| spring-boot-starter-aop | 使用Spring AOP和AspectJ进行面向方面编程的入门 |
| spring-boot-starter-artemis | 使用Apache Artemis的JMS消息传递入门 |
| spring-boot-starter-batch | 使用Spring Batch的入门 |
| spring-boot-starter-cache | 开始使用Spring Framework的缓存支持 |
| spring-boot-starter-cloud-connectors | 使用Spring Cloud Connectors的入门程序，可简化与Cloud Foundry和Heroku等云平台中服务的连接 |
| spring-boot-starter-data-cassandra | 使用Cassandra分布式数据库和Spring Data Cassandra的入门者 |
| spring-boot-starter-data-cassandra-reactive | 使用Cassandra分布式数据库和Spring Data Cassandra Reactive的入门 |
| spring-boot-starter-data-couchbase | 使用Couchbase面向文档的数据库和Spring Data Couchbase的入门 |
| spring-boot-starter-data-couchbase-reactive | 使用Couchbase面向文档的数据库和Spring Data Couchbase Reactive的入门 |
| spring-boot-starter-data-elasticsearch | 使用Elasticsearch搜索和分析引擎以及Spring Data Elasticsearch的入门者 |
| spring-boot-starter-data-jdbc | 使用Spring Data JDBC的入门 |
| spring-boot-starter-data-jpa | 将Spring Data JPA与Hibernate结合使用的入门 |
| spring-boot-starter-data-ldap | 使用Spring Data LDAP的入门 |
| spring-boot-starter-data-mongodb | 使用MongoDB面向文档的数据库和Spring Data MongoDB的入门 |
| spring-boot-starter-data-mongodb-reactive | 使用MongoDB面向文档的数据库和Spring Data MongoDB Reactive的入门 |
| spring-boot-starter-data-neo4j | 使用Neo4j图形数据库和Spring Data Neo4j的入门 |
| spring-boot-starter-data-redis | 将Redis键值数据存储与Spring Data Redis和Lettuce客户端一起使用的入门 |
| spring-boot-starter-data-redis-reactive | 将Redis键值数据存储与Spring Data Redis Reacting和Lettuce客户端一起使用的入门 |
| spring-boot-starter-data-rest | 使用Spring Data REST在REST上公开Spring Data存储库的入门 |
| spring-boot-starter-data-solr | 结合使用Apache Solr搜索平台和Spring Data Solr的入门者 |
| spring-boot-starter-freemarker | 使用FreeMarker视图构建MVC Web应用程序的入门 |
| spring-boot-starter-groovy-templates | 使用Groovy模板视图构建MVC Web应用程序的入门 |
| spring-boot-starter-hateoas | 使用Spring MVC和Spring HATEOAS构建基于超媒体的RESTful Web应用程序的入门者 |
| spring-boot-starter-integration | 使用Spring Integration的入门 |
| spring-boot-starter-jdbc | 通过HikariCP连接池使用JDBC的入门 |
| spring-boot-starter-jersey | 使用JAX-RS和Jersey构建RESTful Web应用程序的入门。的替代品[spring-boot-starter-web](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html" \l "spring-boot-starter-web) |
| spring-boot-starter-jooq | 使用jOOQ访问SQL数据库的入门。替代[spring-boot-starter-data-jpa](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html" \l "spring-boot-starter-data-jpa)或[spring-boot-starter-jdbc](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html" \l "spring-boot-starter-jdbc) |
| spring-boot-starter-json | 读写JSON入门 |
| spring-boot-starter-jta-atomikos | 使用Atomikos的JTA交易入门 |
| spring-boot-starter-jta-bitronix | 使用Bitronix的JTA交易入门 |
| spring-boot-starter-mail | 开始使用Java Mail和Spring Framework的电子邮件发送支持 |
| spring-boot-starter-mustache | 使用Mustache视图构建Web应用程序的入门 |
| spring-boot-starter-oauth2-client | 使用Spring Security的OAuth2 / OpenID Connect客户端功能的入门 |
| spring-boot-starter-oauth2-resource-server | 使用Spring Security的OAuth2资源服务器功能的入门 |
| spring-boot-starter-quartz | 入门使用Quartz Scheduler |
| spring-boot-starter-security | 使用Spring Security的入门 |
| spring-boot-starter-test | 用于使用包括JUnit，Hamcrest和Mockito在内的库测试Spring Boot应用程序的入门程序 |
| spring-boot-starter-thymeleaf | 使用Thymeleaf视图构建MVC Web应用程序的入门 |
| spring-boot-starter-validation | 通过Hibernate Validator使用Java Bean验证的入门 |
| spring-boot-starter-web | 使用Spring MVC构建Web（包括RESTful）应用程序的入门程序。使用Tomcat作为默认的嵌入式容器 |
| spring-boot-starter-web-services | 使用Spring Web Services的入门 |
| spring-boot-starter-webflux | 使用Spring Framework的反应式Web支持构建WebFlux应用程序的入门者 |
| spring-boot-starter-websocket | 使用Spring Framework的WebSocket支持构建WebSocket应用程序的入门 |

除了应用程序starters，，如果想拍出或交换特定的技术方面，可以使用这些启动程序：

| **名称** | **描述** |
| --- | --- |
| spring-boot-starter-jetty | 使用Jetty作为嵌入式servlet容器的入门者。[spring-boot-starter-tomcat](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html" \l "spring-boot-starter-tomcat)的替代品 |
| spring-boot-starter-log4j2 | 使用Log4j2进行日志记录的启动器。[spring-boot-starter-logging](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html" \l "spring-boot-starter-logging)的替代品 |
| spring-boot-starter-logging | 使用Logback进行日志记录的启动器。默认记录启动器 |
| spring-boot-starter-reactor-netty | 使用Reactor Netty作为嵌入式反应式HTTP服务器的入门者。 |
| spring-boot-starter-tomcat | 入门，用于将Tomcat用作嵌入式servlet容器。默认使用的servlet容器启动器[spring-boot-starter-web](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html#spring-boot-starter-web) |
| spring-boot-starter-undertow | 使用Undertow作为嵌入式servlet容器的入门。的替代品[spring-boot-starter-tomcat](https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html#spring-boot-starter-tomcat) |