**第2章**

开发必备工具

通过第1章对Spring Boot有了感性的认识后，本章对本书所用的常用工具进行简单快速地介绍。介绍的目的当然不是深入地学习相关工具，这超出了本书的范围，建议读者专门去学习在本章中涉及的每个工具。

2.1 IntelliJ IDEA

IntelliJ IDEA是Java开发的最优秀的IDE之一。它有两个版本：社区版（Community Edition）和旗舰版（Ultimate Edition），但只有旗舰版支持Spring相关的开发，本小节以IntelliJ IDEA旗舰版进行演示（旗舰版可试用30天）。

2.1.1 下载安装

IntelliJ IDEA的下载地址为https://www.jetbrains.com/idea/download，请按照对应的操作系统选择下载，这里选择Ultimate Edition，如图2-1所示。

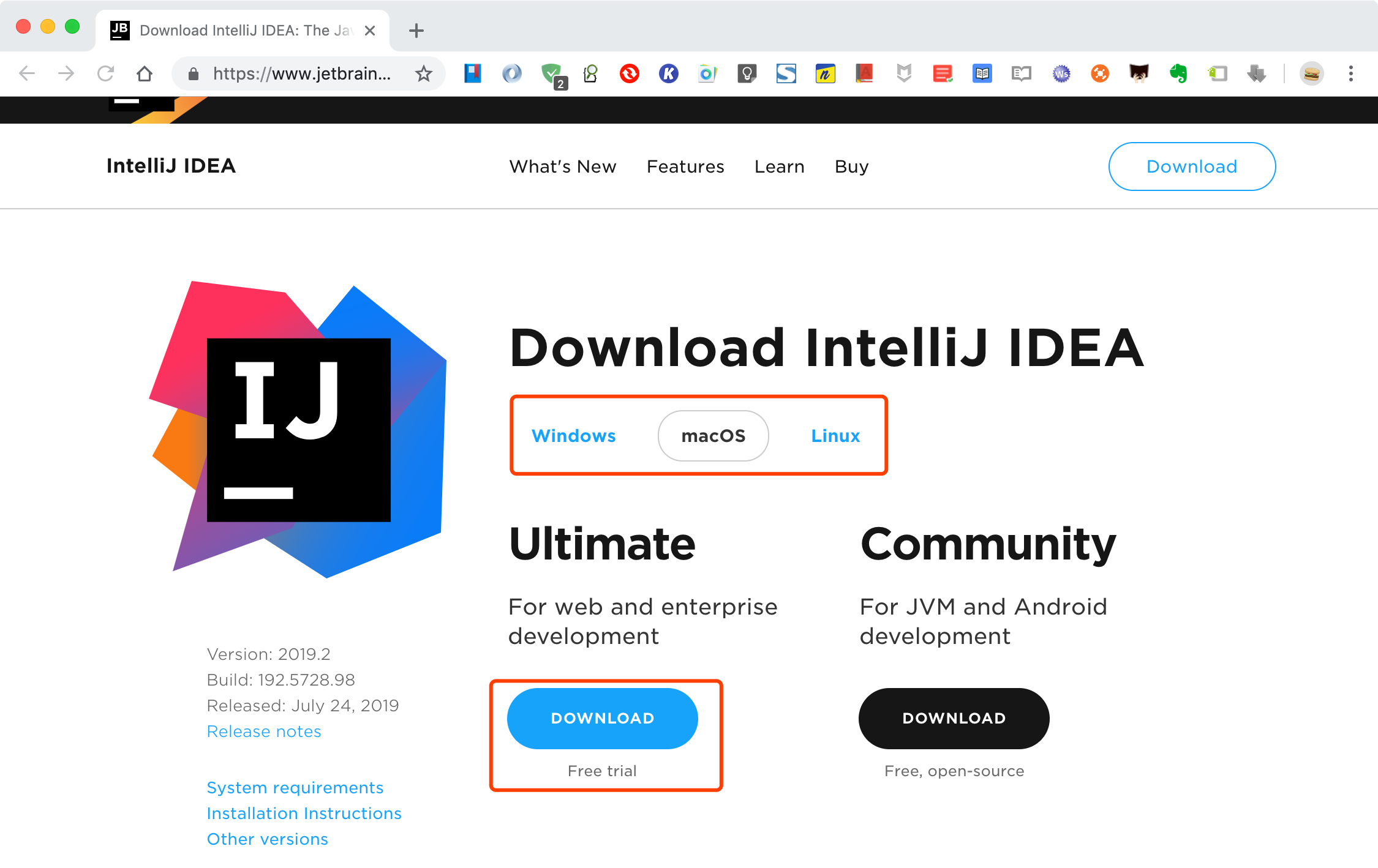


图2-1

2.1.2 界面功能

1．欢迎界面

第一次打开软件，按照默认设置，看到的是IntelliJ IDEA的欢迎界面，如图2-2所示。



图2-2

（1）Create New Project：创建新的项目。

（2）Import Project：导入项目，如导入Eclipse项目。

（3）Open：打开项目或文件夹。

（4）Check out from Version Control：从代码控制库（Git、SVN）导入代码 。

2．创建一个Spring Boot项目

首先单击“Create New Project”选项，在左侧选择“Spring Initializr”选项；然后单击“Next”按钮，填写项目信息，如图2-3所示。

* Group：top.wisely
* Artifact：intellij-spring-boot

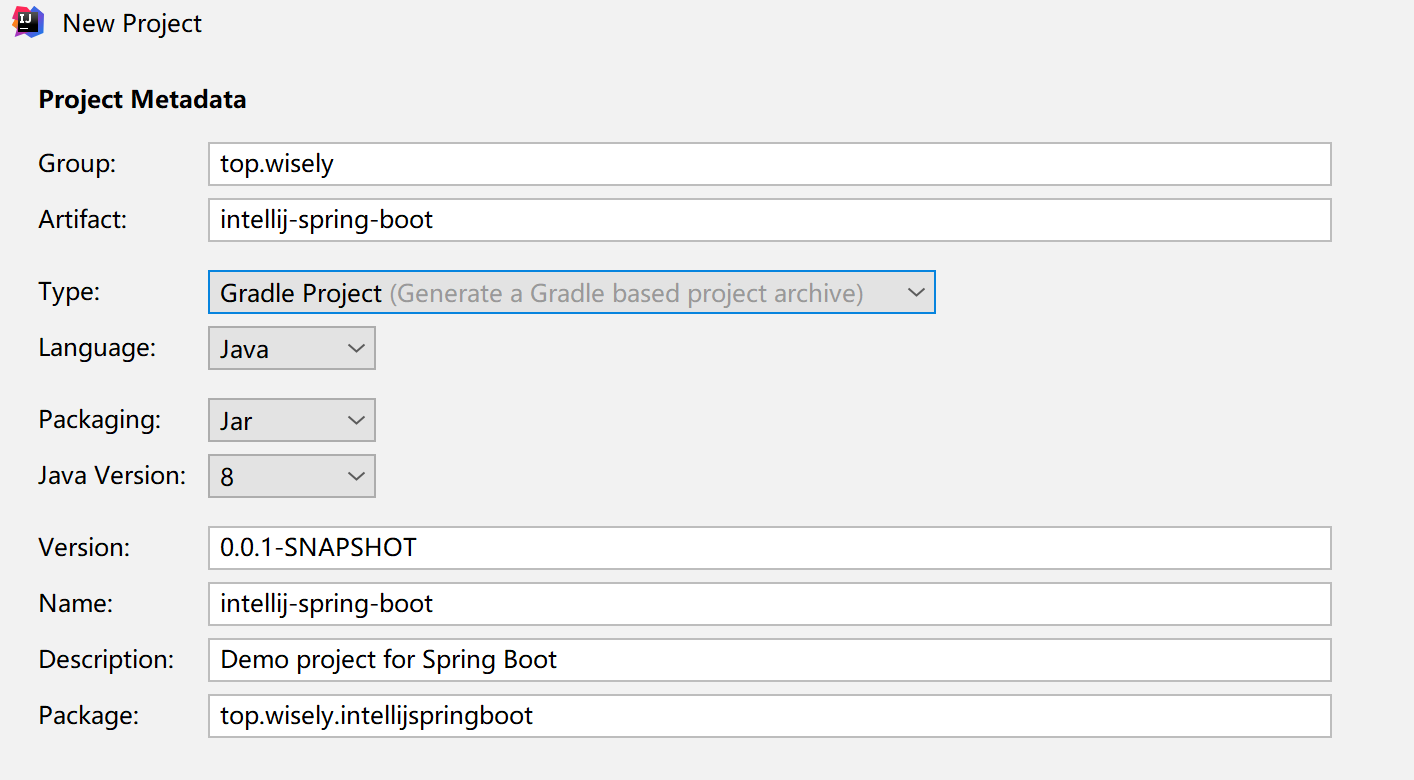


图2-3

填写好信息后，单击“Next”按钮搜索选择Spring Boot项目的依赖和版本。依赖选择Spring Web Starter和Lombok，版本依赖可选择当前任意版本，如图2-4所示。

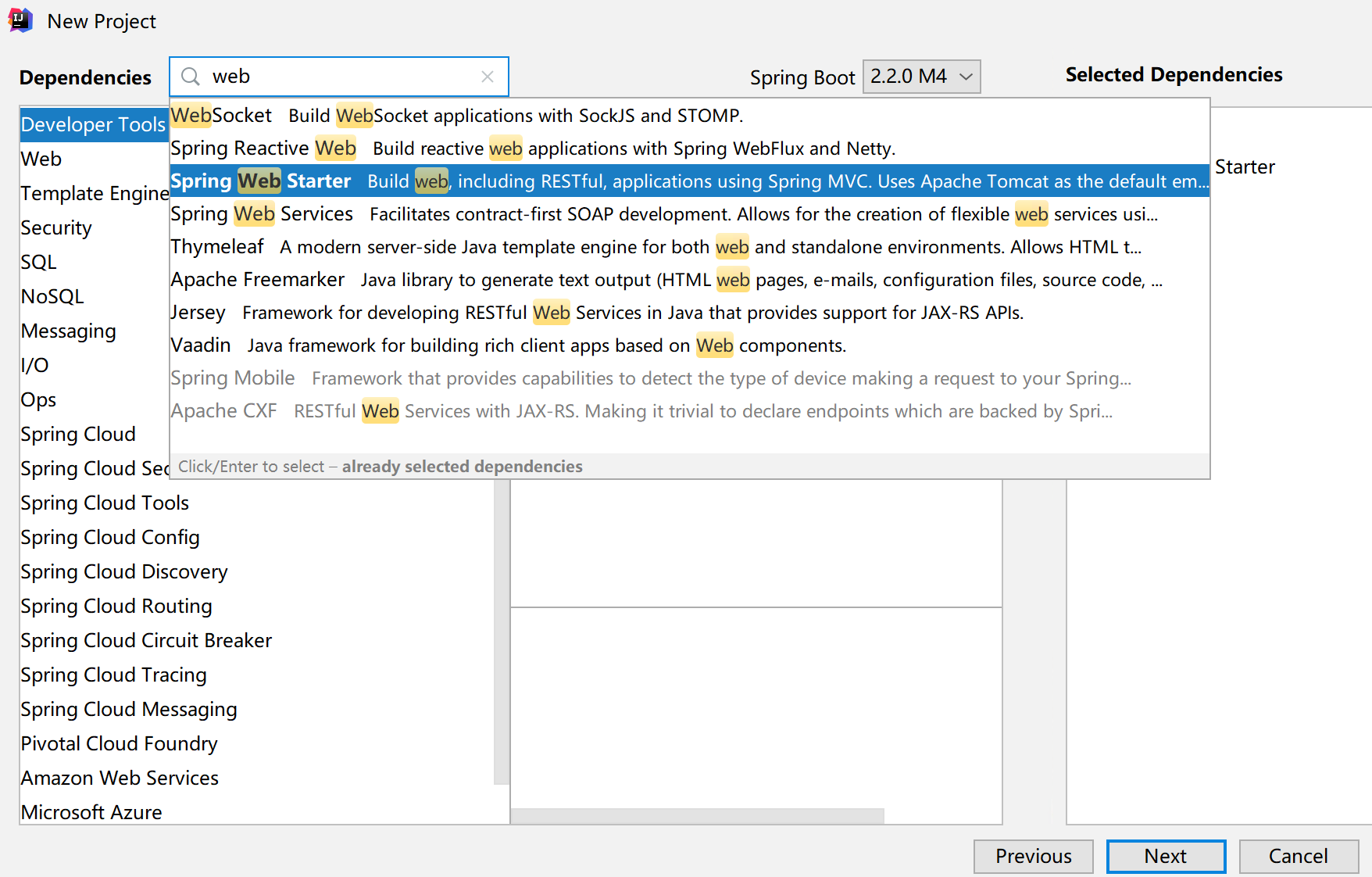


图2-4

单击“Next”按钮后，单击“Finish”按钮完成新建项目。

3．主页面

在完成新建项目后，IntelliJ IDEA会自动下载Spring Boot项目的依赖，主页面的右下角显示依赖下载的进度，下载完成的主页面如图2-5所示。

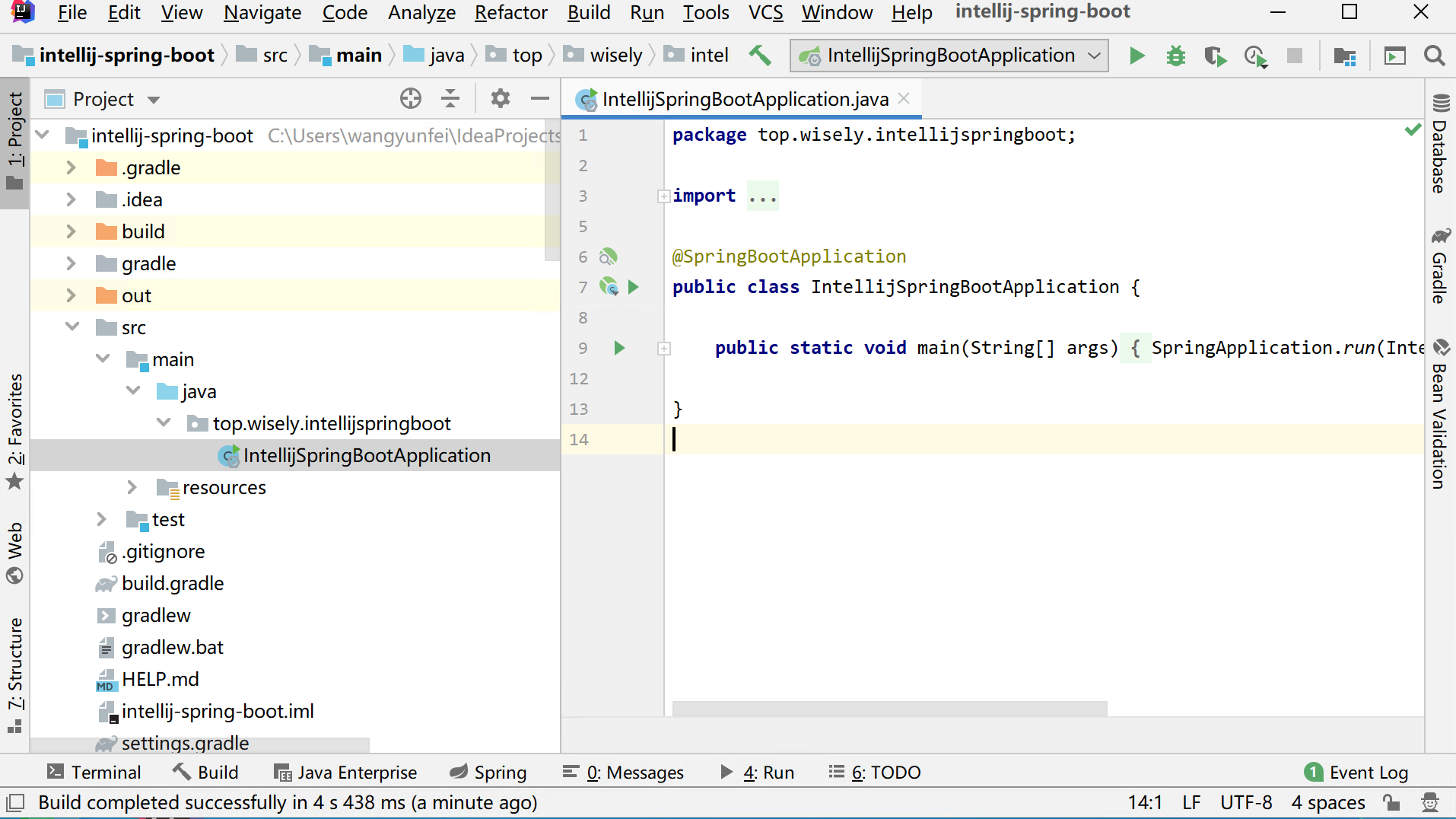


图2-5

（5）启动应用：单击右上角工具栏三角符号启动应用。

（6）调试应用：单击右上角工具栏虫子符号调试应用。

2.1.3 主要功能和快捷键

Linux系统的快捷键与Windows通用，因而不再单独列出。若按快捷键执行后没有效果，则很可能是因为和当前系统的某些软件的快捷键有冲突。

1．基础检索

（1）当前页检索：检索当前文件的字符。

* Windows：Ctrl + F。
* macOS：Command + F。

（2）在路径中查询：检索指定路径或范围。

* Windows：Ctrl + Shift + F（可能会有和其他快捷键额冲突，可单击菜单Edit -> Find -> Find in Path直接调用）。
* macOS：Command + Shift + F。

2．检索所有

检索所有文件可检索类、文件、Symbol、Action，Windows和macOS的快捷键都是双击**Shift**键盘，检索时可以使用检索对象名称中的任意字符按顺序检索，如图2-6所示。

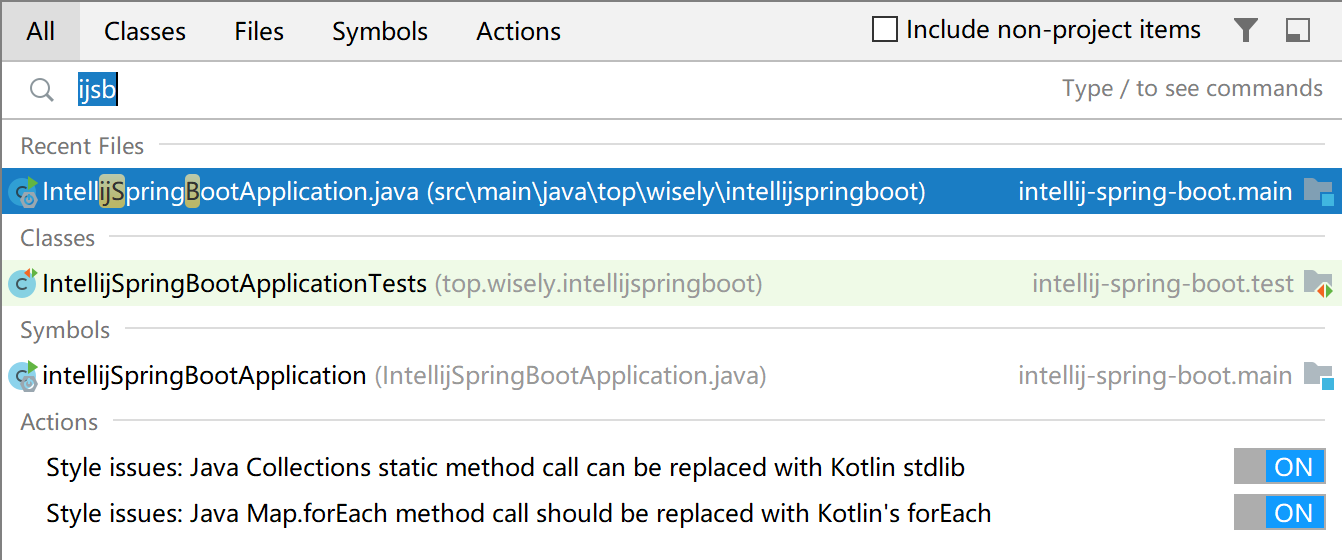


图2-6

3．检索类

（1）检索类：只检索类，属于“检索所有”的一个页签。

* Windows：Ctrl + N。
* macOS：Command + O。

（2）查看类的定义和使用：用光标选中某个类或接口，查看它的定义，第三方的源码也可查看。

* Windows：Ctrl + B。
* macOS：Command + B。

（3）查看父类或接口：用光标选中某个类或接口，查看其父类或接口。

* Windows： Ctrl + U。
* macOS：Command + U。

（4）查看子类或实现类：用光标选中某个类或接口，查看其子类或实现类。

* Windows：Ctrl + Alt + B。
* macOS：Command + Option + B。

4．快速修复

使用快速修复可修复代码中的错误，如导包、实现接口方法等。

* Windows：Alt + Enter。
* macOS：Option + Enter。

5．修改名称

* Windows：Shift + F6。
* macOS：Shift + F6。

6．生成代码

（7）生成代码。

可通过生成代码生成构造器、getter、setter、覆盖方法等。

* Windows：Alt + Insert。
* macOS：Command + N。

（8）覆盖父类方法。

* Windows：Ctrl + O。
* macOS：Ctrl + O。

（9）实现接口方法。

* Windows：Ctrl + I。
* macOS：Ctrl + I。

7．优化导入

优化导入可去除无用的类的import。

* Windows：Ctrl + Alt + O。
* macOS：Ctrl + Option + O。

8．格式化代码

* Windows：Ctrl + Alt + L。
* macOS：Command + Option + L。

9．展开代码和折叠代码

（1）展开代码。

* Windows：Ctrl + “+”。
* macOS：Command + “+”。

（2）折叠代码。

* Windows：Ctrl + “-”。
* macOS：Command + “-”。

10．代码注释

（1）行注释。

* Windows：Ctrl + /。
* macOS：Command + /。

（2）块注释。

* Windows：Ctrl + Shift + /。
* macOS：Command + Option + /。

11．删除代码

（1）删除行。

* Windows：Ctrl + Y。
* macOS：Command + delete。

（2）删除到词尾。

* Windows：Ctrl + delete。
* macOS：Option + delete + fn（笔记本键盘无向右删除键，可结合fn键让向左删除键具备向右删除的功能）。

（3）删除到词头。

* Windows：Ctrl + Backspace。
* macOS：Option + delete。

更多快捷键可在设置中查看“Keymap”，如图2-7所示。



图2-7

或者下载Intellij IDEA的快捷键参考：https://www.jetbrains.com/idea/docs/IntelliJIDEA\_ ReferenceCard.pdf。

2.2 Gradle

Gradle是一个开源通用的项目构建管理系统，它的主要功能是依赖包的管理和构建任务的执行。它基于Groovy语言的DSL（Domain Specific Language，领域特定语言），而不是XML，使用上更加灵活，构建速度更快。在大部分场景下，Gradle的构建速度是Maven的两倍；在某些情况下，甚至是Maven构建速度的100倍。它建立在Maven、ANT的优点之上，是Android开发的默认编译工具，也是Spring Boot支持的主要编译工具。鉴于以上优点，本书以Gradle作为构建工具。

2.2.1 安装

一般来说，在开发人员的工作电脑上的操作系统大多为Windows和macOS，下面分别使用Windows下的Chocolatey，以及MacOS下的Homebrew这样的软件包管理工具来安装Gradle。这些软件包管理工具还可以用来安装其他软件。

1．Windows

（1）“以管理员身份运行”打开“命令提示符“（cmd），如图2-8所示。

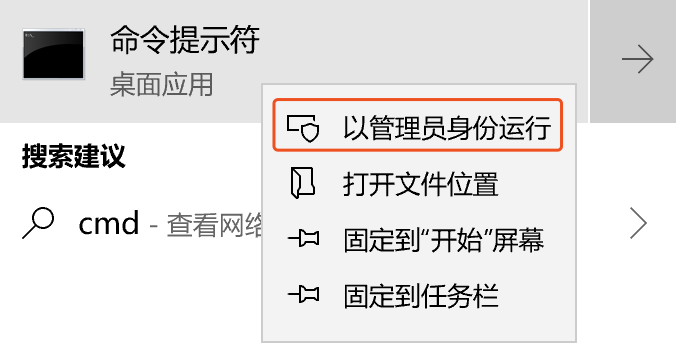


图2-8

（2）确认Java安装：java –version，如图2-9所示。

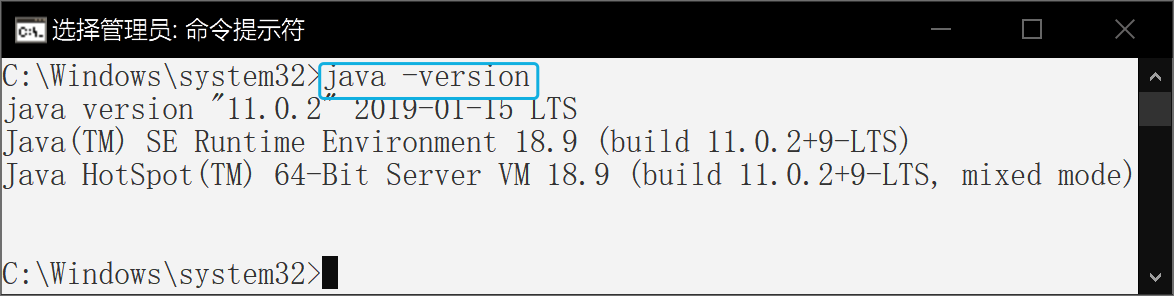


图2-9

（3）安装Windows下的软件包管理工具Chocolatey，安装地址为<https://chocolatey.org/install>。安装命令如下。

‌@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

安装完成后重新打开cmd，如图2-10所示。

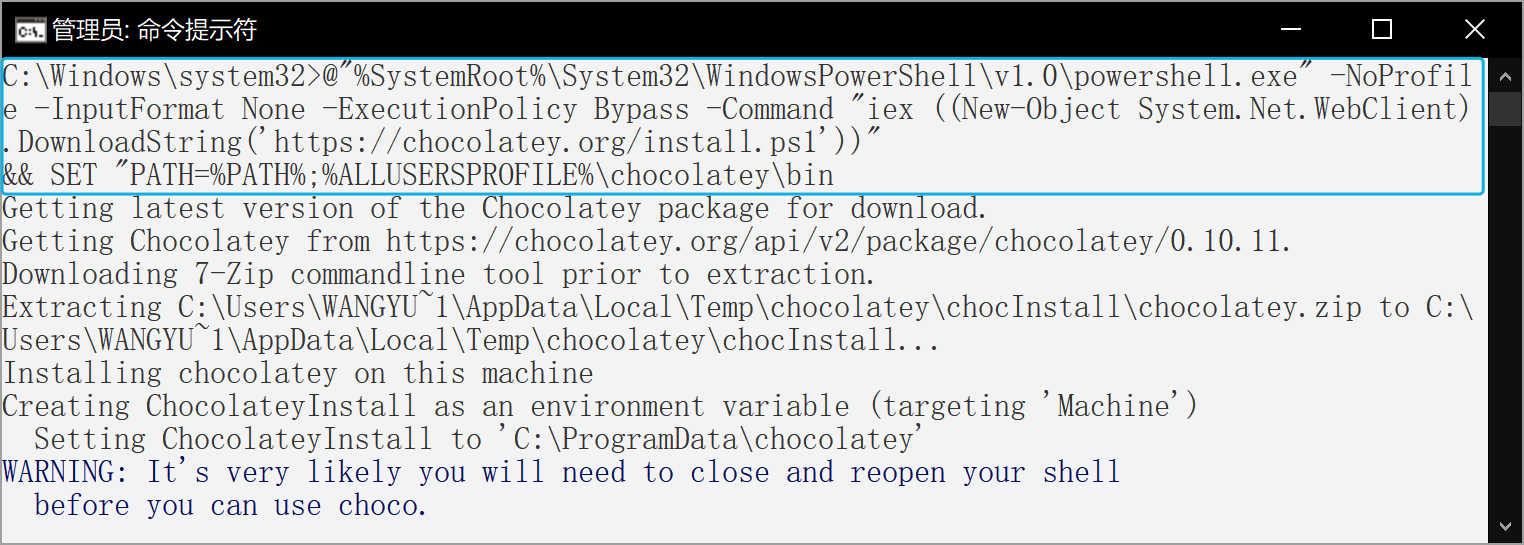


图2-10

（4）安装Gradle，安装命令为choco install gradle，安装完成后重新打开cmd，如图2-11所示。

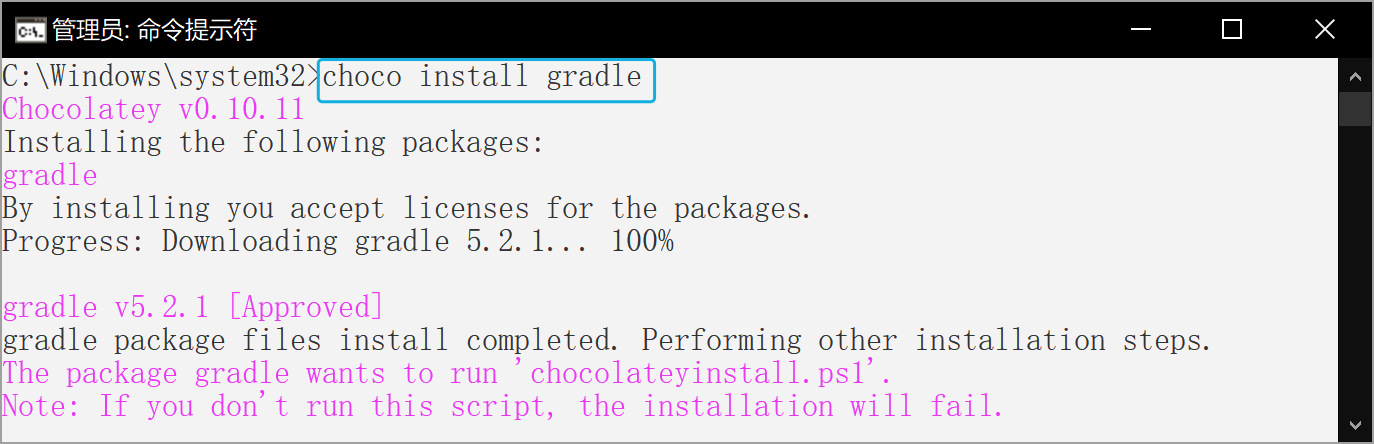


图2-11

（5）确认安装：gradle –v，如图2-12所示。

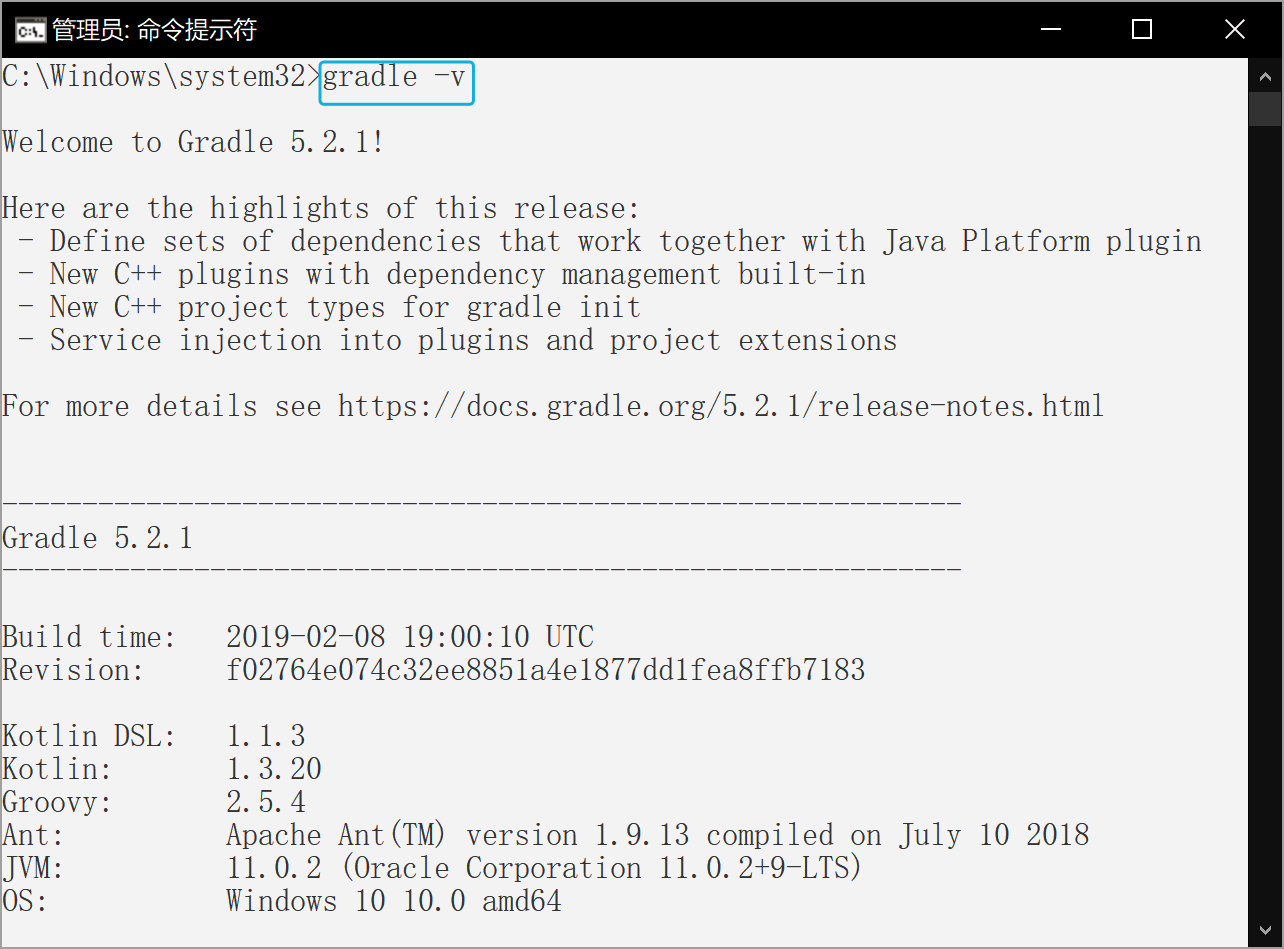


图2-12

2．macOS

（1）确认Java安装：java –version，如图2-13所示。



图2-13

（2）安装macOS下的软件包管理工具Homebrew，地址为<https://brew.sh/>。安装命令如下。

/usr/bin/ruby -e "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install>)"

如图2-14所示。

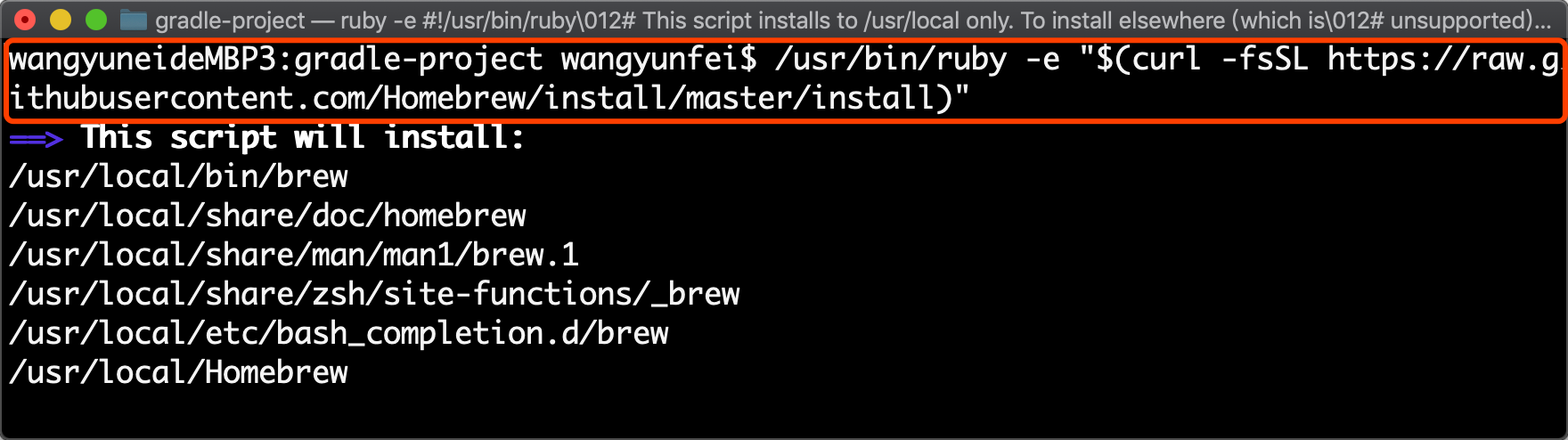


图2-14

（3）安装Gradle，安装命令为brew install gradle，如图2-15所示。

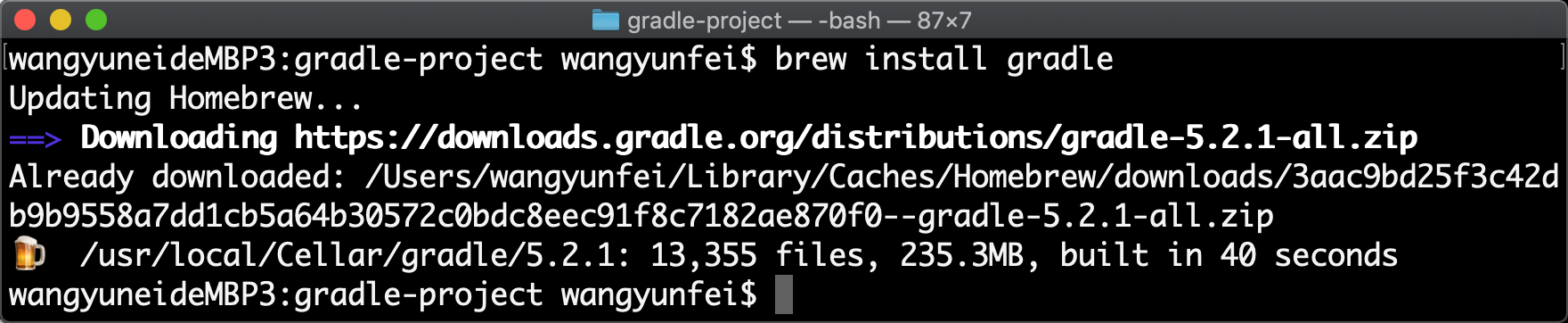


图2-15

（4）确认安装：gradle –v，如图2-16所示。

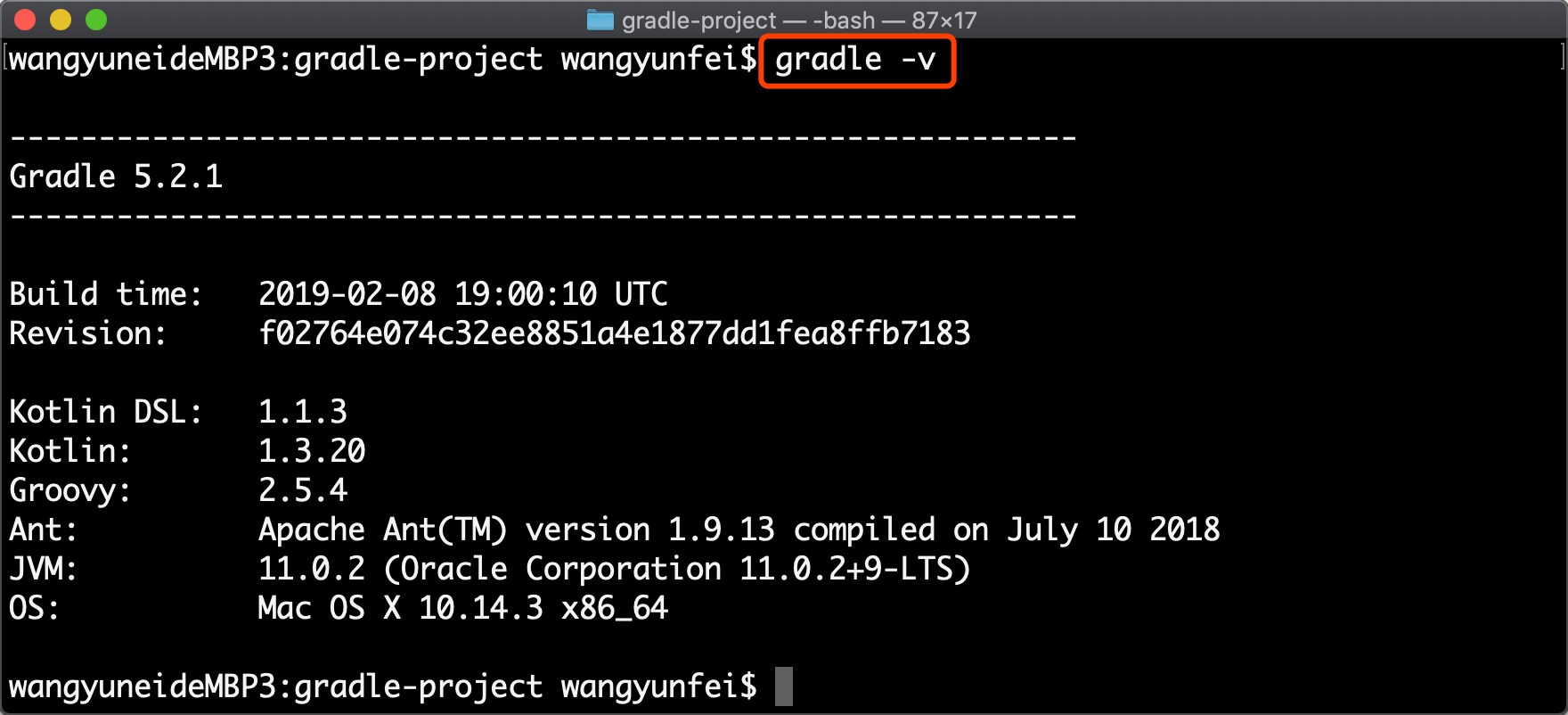


图2-16

更多安装方式请参考官网。

2.2.2 任务

像Maven的pom.xml一样，Gradle有一个构建文件build.gradle，这个文件主要由任务（task）、插件（plugin）和依赖（dependecy）组成。下面新建一个目录gradle-project，在目录中新建一个文件build.gradle，并在文件中写入下面的内容。

task hello {  
 doLast {  
 println 'Hello World!'  
 }  
}  
task hello2 {  
 doLast {  
 println 'Hello Gradle!'  
 }  
}

当前构建文件定义了两个任务（task），名称分别为hello和hello2，内容分别是打印“Hello World!”和打印“Hello Gradle!”。

（10）项目task列表：gradle tasks –all，如图2-17所示。

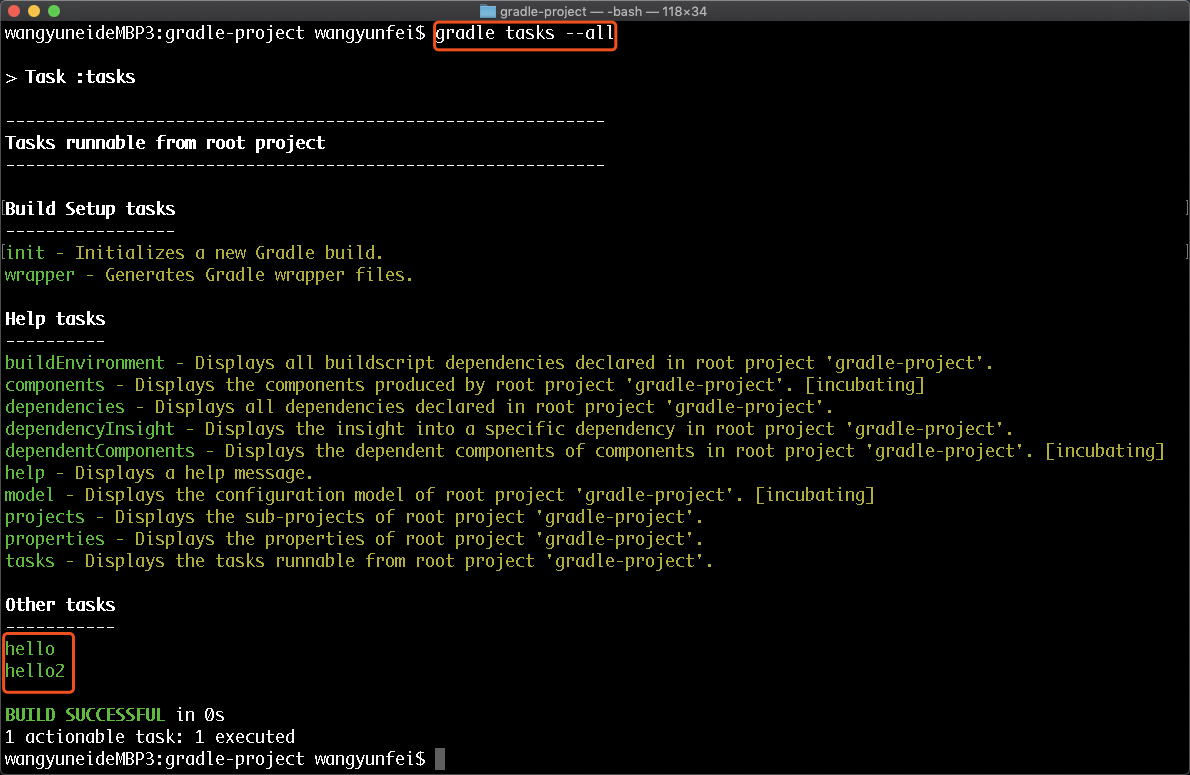


图2-17

（11）执行hello task：gradle hello、gradle hello2，如图2-18所示。

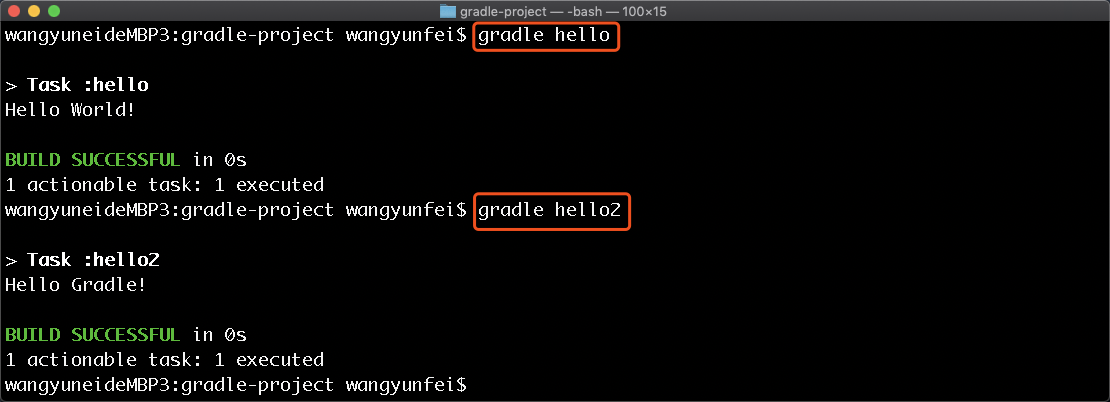


图2-18

2.2.3 插件

插件是由一系列的任务组成的，在用Java进行开发测试时，需要使用Java插件。Java插件包含assemble、build、buildDependents、buildNeeded、classes、clean、jar和testClasses等任务。使用Java插件可以在build.gradle文件中做如下定义。

apply plugin: 'java' 或

plugins {  
 id 'java'  
}

新建目录gradle-java，目录下的build.gradle中的内容为apply plugin: java，在“gradle-java/src/main/java/top/wisely”目录下新建一个普通的“Hello.java”类。

package top.wisely;  
public class Hello {  
 public static void main(String[] args) {  
 System.out.println("Hello World!!");  
 }  
}

查看当前任务列表：gradle tasks，可以发现多了很多任务，如图2-19所示。

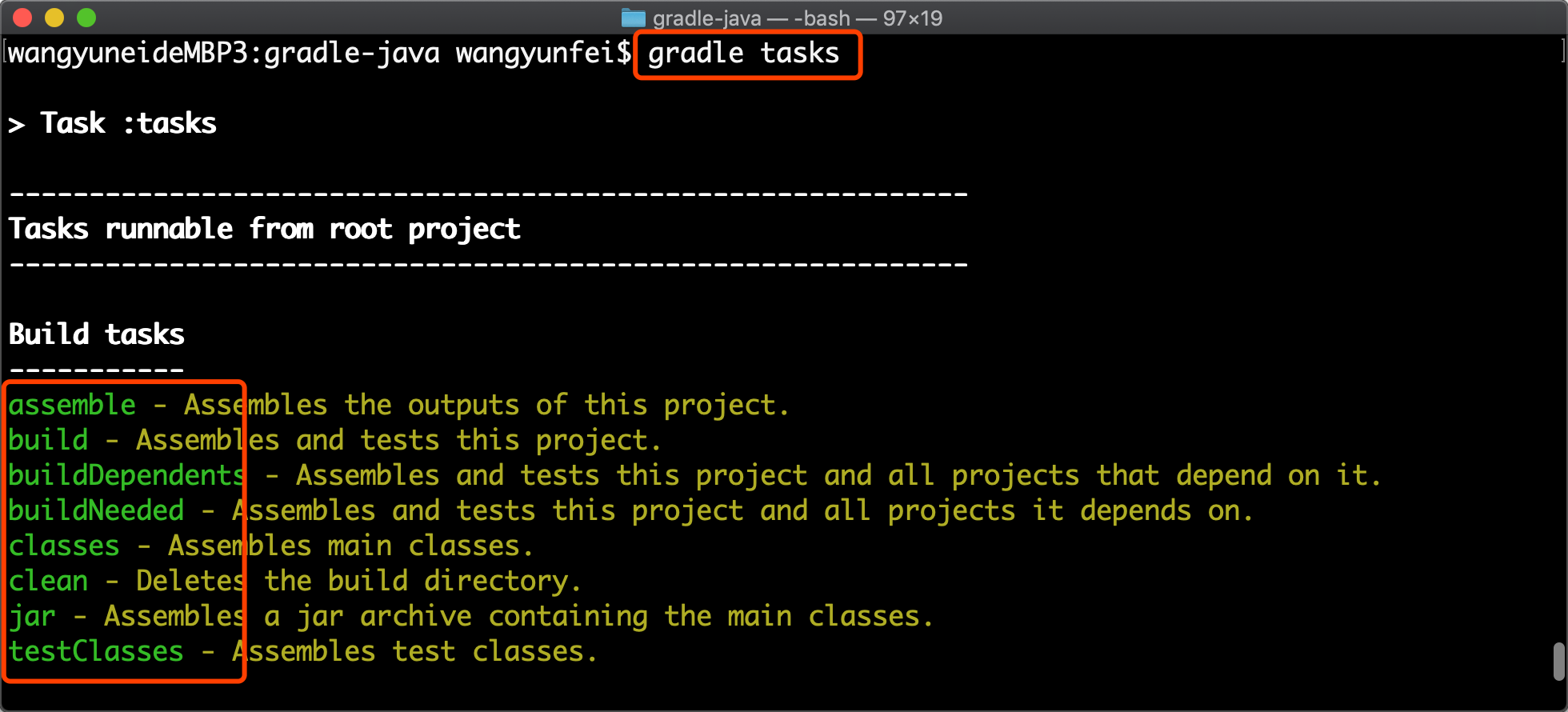


图2-19

执行编译任务：gradle build，在gradle-java下多了一个build目录。“Hello.java”被编译成了“Hello.class”，并构建了一个“gradle-java.jar”，如图2-20所示。

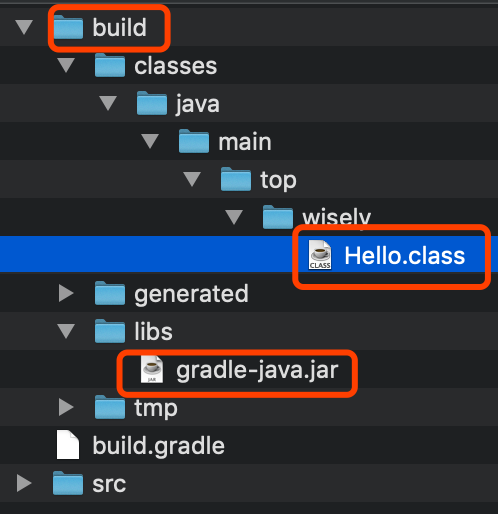


图2-20

2.2.4 依赖

Gradle最重要的功能便是对第三方包的依赖管理。

1．定义下载库

首先，在build.gradle里声明依赖包从何处下载。

repositories {  
 mavenCentral()   
}

（1）添加Maven中心库作为依赖查找库，地址为<https://repo.maven.apache.org/maven2/>。

（2）也可以以url形式进行添加。

repositories {  
 mavenCentral()  
 maven { url 'https://repo.spring.io/snapshot' }  
 maven { url 'https://repo.spring.io/milestone' }  
}

2．添加依赖

定义依赖，如添加spring-web依赖。

implementation group: 'org.springframework', name: 'spring-web', version: '5.1.5.RELEASE'

或简写成下面这种形式。

implementation 'org.springframework:spring-web:5.1.5.RELEASE'

依赖的定义分成三个部分。

（1）group：组织域名。

（2）name：组织域名内唯一名称。

（3）version：版本号。

3．依赖类型

（1）implementation：编译时依赖。如implementation 'org.springframework:spring-web:5.1.5. RELEASE'。

（2）compileOnly：只在编译时依赖，而不在运行时依赖。org.projectlombok:lombok:1.18.6；

（3）runtimeOnly：只在运行时依赖，而不在编译时依赖。org.postgresql:postgresql:42.2.5；

（4）testImplementation：用来编译测试时依赖。junit:junit:4.12。

4．版本号说明

以Spring为例：5.1.5.RELEASE。

（1）第一个数字5为主要版本Major，版本更新后API不兼容。

（2）第二个数字1为次要版本Minor，版本更新后API兼容。

（3）第三个数字5为补丁版本Patch，版本更新后API兼容的bug修复。

后缀英文有以下几种。

（1）BUILD-SNAPSHOT：快照版本，处于密集开发阶段的一次版本发布，不建议使用。

（2）Milestone：里程碑版本，完成了一组重要功能特性，可以尝试新特性。如M1、M2、M3。

（3）Release Candidate：候选发布版本，属于beta版本，意味着功能已完备，经过几轮bug修复后即将发布为正式版本，可以全面学习和测试。如RC1、RC2、RC3。

（4）RELEASE：正式发布版。

5．完成gradle-java

（1）像第三方依赖一样，我们自己的应用也需要定义自己的group和version，name默认为应用名称。

group = 'top.wisely'   
version = '0.0.1-SNAPSHOT'

（2）完整的build.gradle如下。

apply plugin: 'java'  
  
group = 'top.wisely'  
version = '0.0.1-SNAPSHOT'  
  
repositories {  
 mavenCentral()  
}  
  
dependencies {  
 implementation 'org.springframework:spring-web:5.1.5.RELEASE'  
}

（3）显示依赖：gradle dependencies，如图2-21所示。

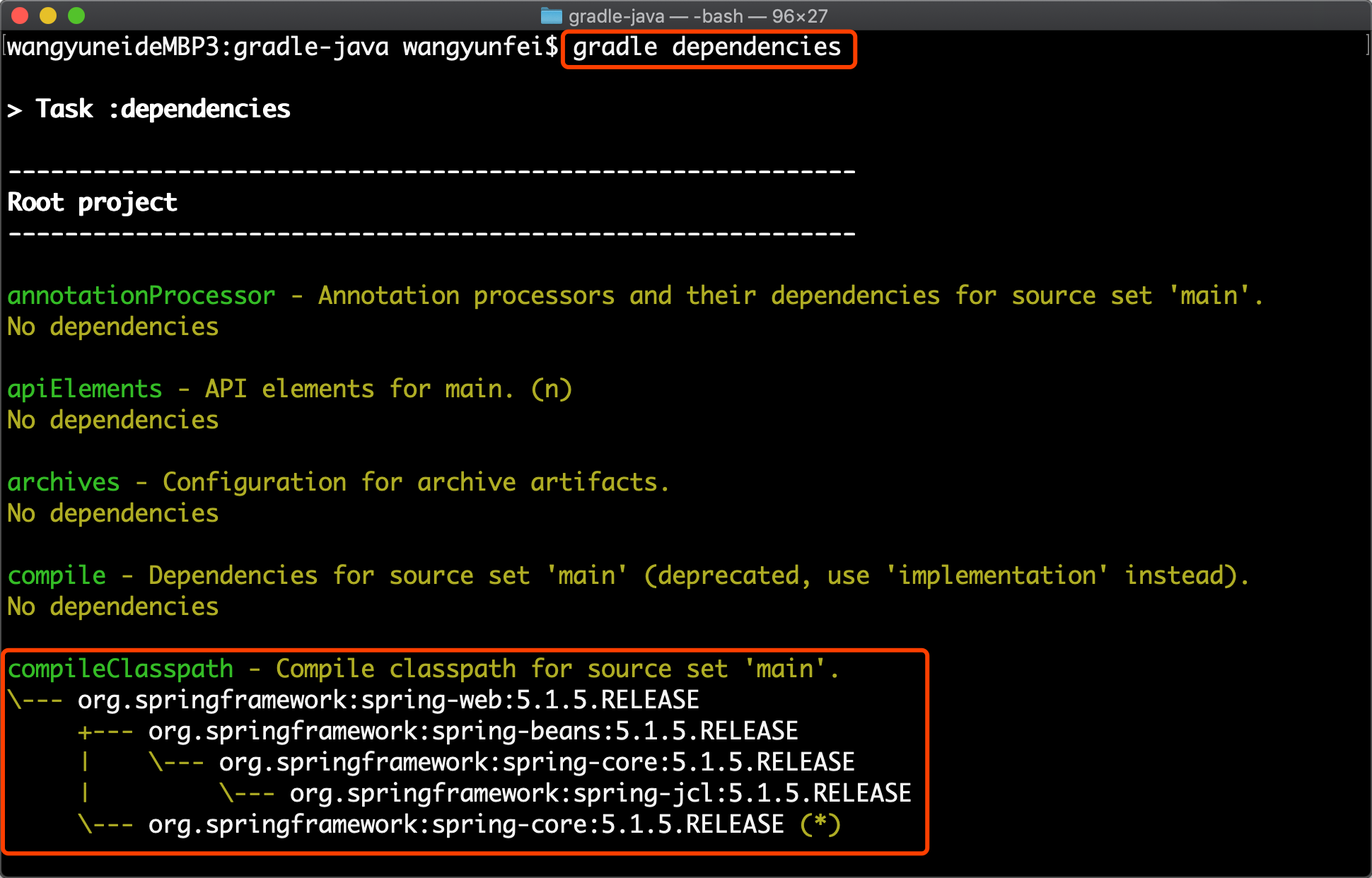


图2-21

6．查找依赖

可以访问<https://mvnrepository.com/>，检索所需要的第三方依赖，如图2-22所示。

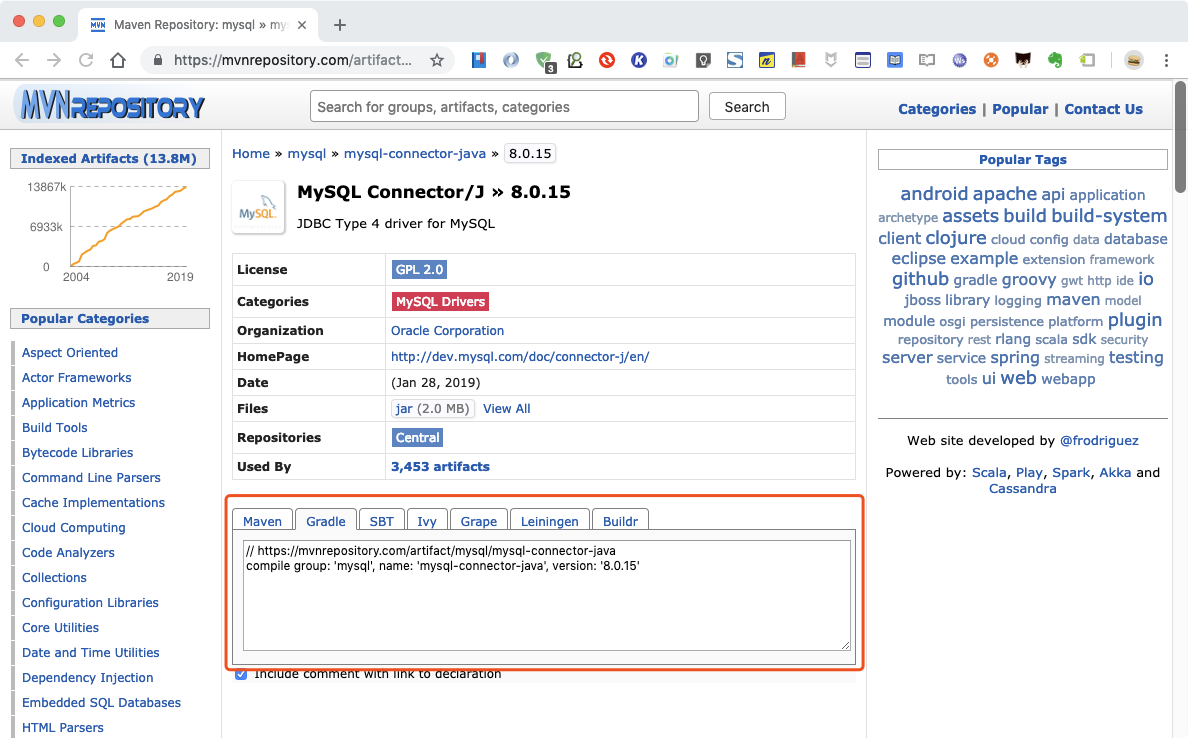


图2-22

2.2.5 Wrapper

Wrapper是包含了指定Gradle版本的脚本，可以在使用的时候下载。这样在开发程序时，无须下载安装配置Gradle即可直接使用。

Spring Boot使用的是Gradle Wrapper，因而可以使用与Spring Boot最匹配的Gradle版本，且无须专门安装配置Gradle。

Spring Boot的应用下有gradlew和gradlew.bat两个文件，其中，gradlew可在macOS或Linux系统下使用，gradlew.bat可在Windows系统下使用。

2.3 Lombok

Lombok可以通过简单的注解代替大量的模板代码，从而极大地提高编码效率。

2.3.1 开启Lombok支持

想要在IntelliJ IDEA下使用Lombok，需要进行如下几步。

1．在IntelliJ IDEA 下安装Lombok插件

（1）进入插件安装界面。

* Windows：File→Settings→plugins
* macOS：IntelliJ IDEA→Preferences→plugins

（2）搜索Lombok插件安装，如图2-23所示。

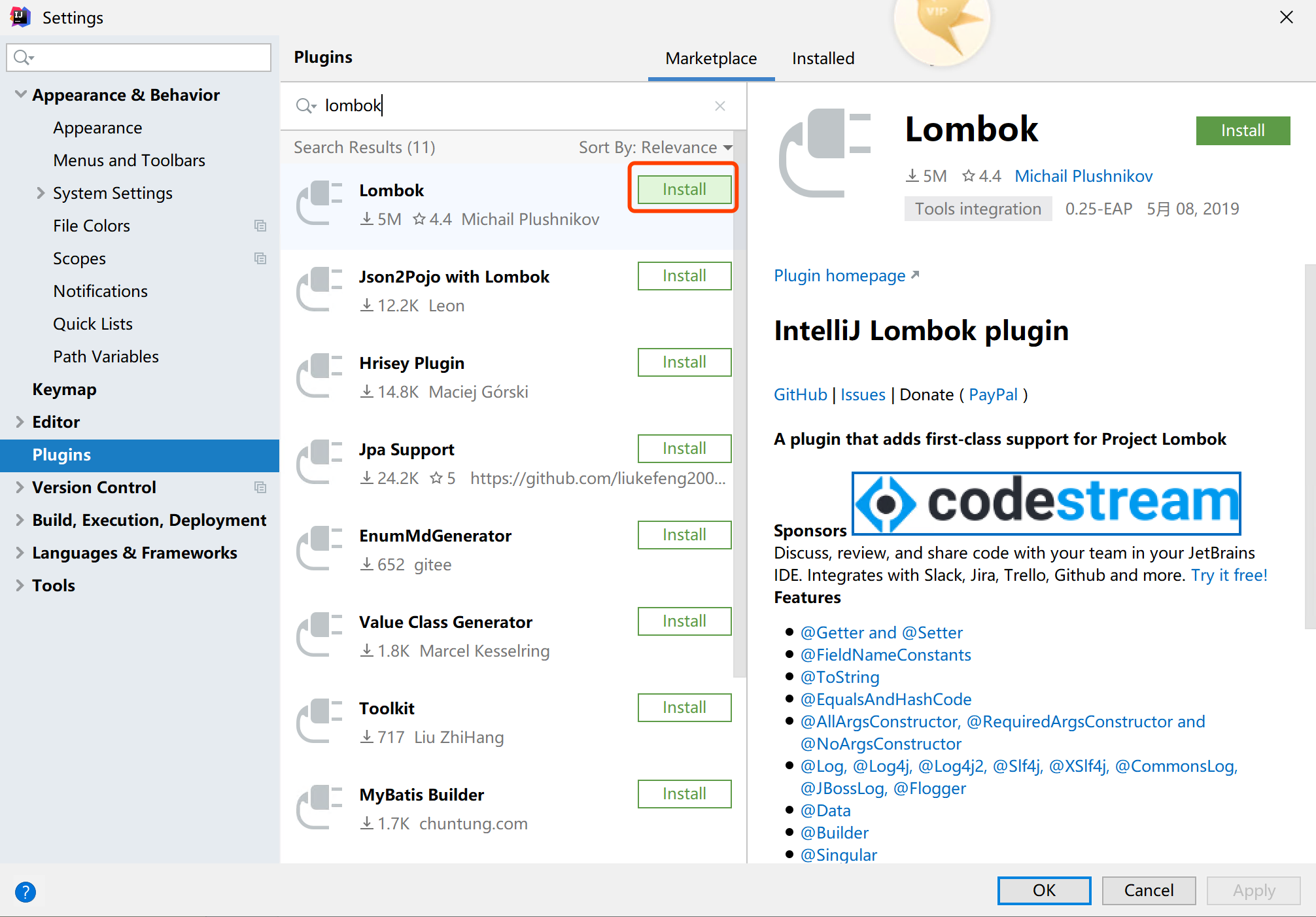


图2-23

（3）安装完成后重启IntelliJ IDEA。

2．应用集成Lombok依赖

上面的intellij-sprint-boot应用已添加了Lombok的依赖，它在build.gradle中的声明如下。

dependencies {  
 //……  
 compileOnly 'org.projectlombok:lombok'  
 annotationProcessor 'org.projectlombok:lombok'  
 //……  
}

3．开启注解处理

至此，在IntelliJ IDEA上开启注解处理即可使用Lombok，如图2-24所示。

Windows：Files→Settings→Build,Execution,Deployment→Compiler→Annotation Processors

macOS：IntelliJ IDEA→Preferences→Build,Execution,Deployment→Compiler→Annotation Processors

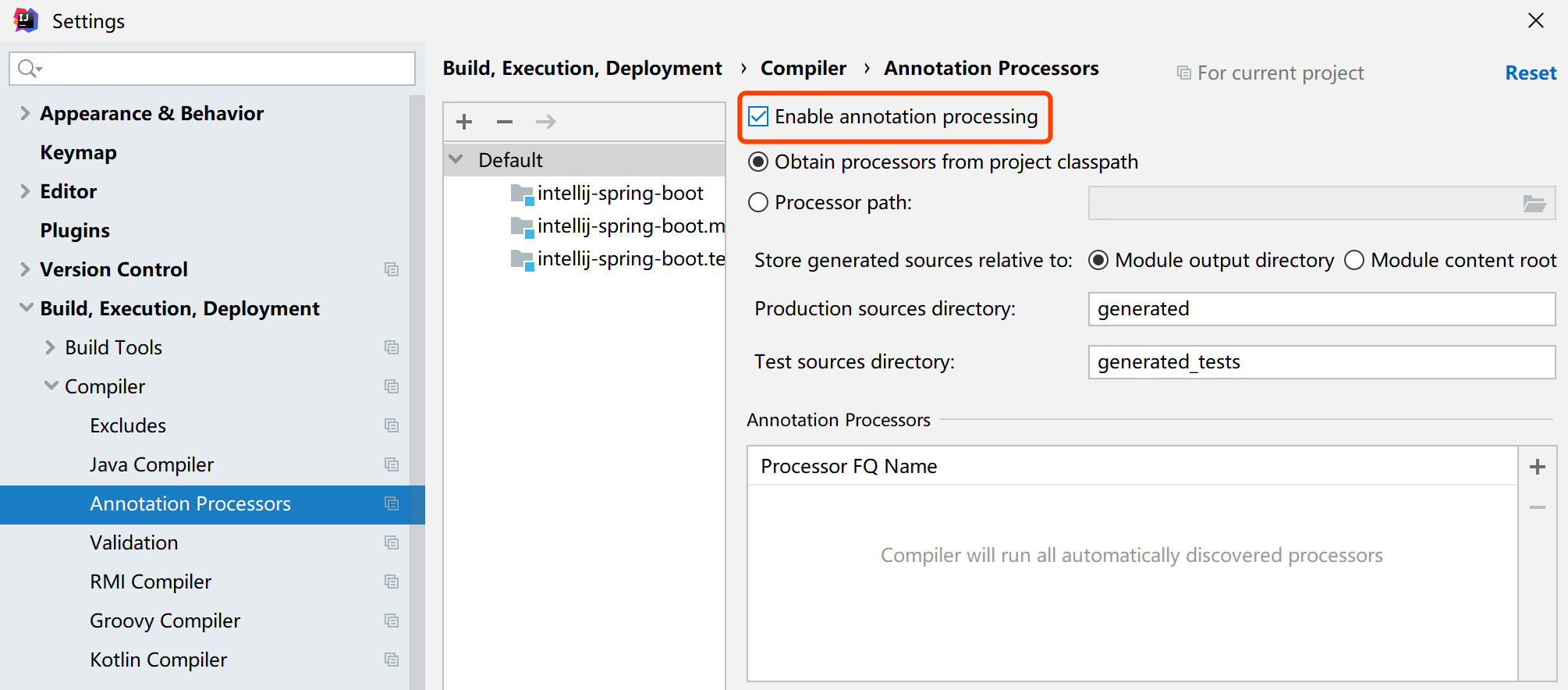


图2-24

2.3.2 Lombok常用注解

（1）@Getter：为属性自动生成getter方法。

（2）@Setter：为属性自动生成setter方法。

（3）@ToString：为类自动生成toString()方法.

（4）@EqualsAndHashCode：为对象相等性比较生成hashCode()和equals()方法.

（5）@NoArgsConstructor：为类自动生成无参构造器.

（6）@AllArgsConstructor：为类自动生成全参构造器.

（7）@Builder：可自动生成建造者模式所需构造对象的方法.

（8）@Log：可自动生成日志对象的声明.

（9）@Data：组合了@ToString、@EqualsAndHashCode、@Getter、@Setter和@RequiredArgsConstructor注解。

后面的章节会有大量的Lombok功能的演示。

2.4 Docker

Docker是微服务部署实践的核心技术，在本书中，数据库软件的部署都依赖于Docker。

2.4.1 安装Docker Desktop

Docker Desktop的下载地址如下。

Windows：<https://hub.docker.com/editions/community/docker-ce-desktop-windows>。

macOS：<https://hub.docker.com/editions/community/docker-ce-desktop-mac>。

安装前请注意开启主板BIOS上对虚拟化技术的支持，Windows下会自动使用Hyper-V作为虚拟化技术。

2.4.2 阿里云镜像加速

直接下载Docker官方镜像速度很慢，阿里云提供了镜像加速的功能。访问阿里云的“容器镜像服务”（<https://cr.console.aliyun.com/>），在“镜像中心”→“镜像加速器”处获取加速地址，如图2-25所示。

获得加速地址后，在Docker Desktop中添加如下设置。

Windows：Settings->Daemon->Registry mirrors

macOS：Preferences->Daemon->Registry mirrors

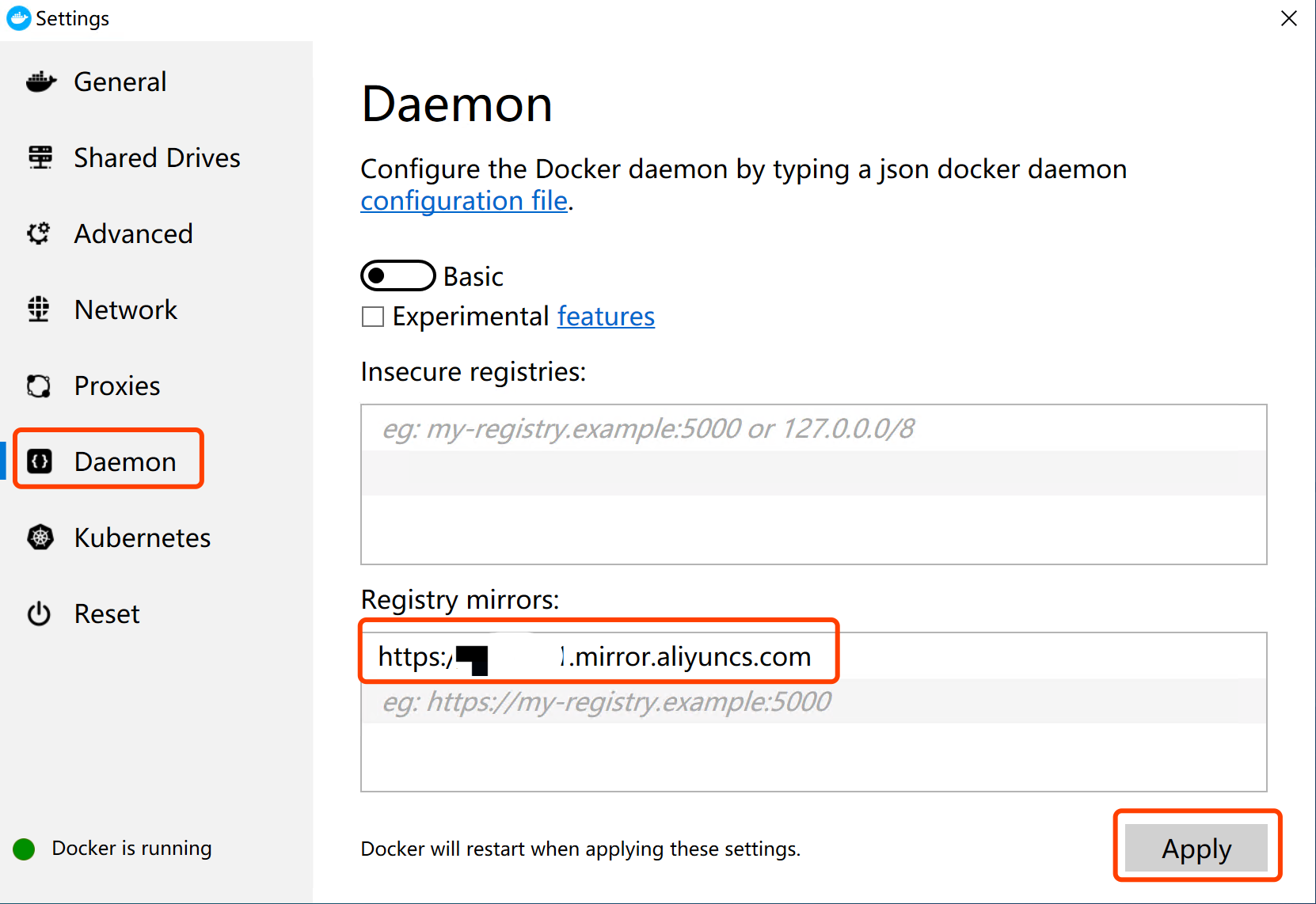


图2-25

2.4.3 Docker的基本使用

（1）查找镜像。

访问https://hub.docker.com/，检索镜像，如Tomcat。

（2）下载镜像：

$ docker pull 镜像名称

下载Tomcat镜像命令：

$ docker pull tomcat

2.4.4 Docker Compose

Docker Compose是用来运行多容器应用的工具，它依赖于一个声明式的YAML文件来运行容器。Docker Desktop自带了Docker Compose工具。

如安装MySQL数据库的YAML文件。

version: '3.1'  
  
services:  
  
 db:  
 image: mysql  
 command: --default-authentication-plugin=mysql\_native\_password  
 restart: always  
 ports:  
 - 3306:3306  
 environment:  
 MYSQL\_DATABASE: first\_db  
 MYSQL\_ROOT\_PASSWORD: zzzzzz  
  
 adminer:  
 image: adminer # 全功能数据库管理工具  
 restart: always  
 ports:  
 - 8081:8080

用Docker Compose命名安装MySQL和adminer：

$ docker-compose -f stack.yml up -d

停止当前容器命令：

$ docker-compose -f stack.yml stop

删除当前容器命令：

$ docker-compose -f stack.yml rm

2.5 其他工具

2.5.1 Postman

本书使用Postman验证来书中出现的各种系统RESTful API。Postman本身是可视化的，因而非常易掌握，后面章节会有大量的Postman的演示。

Postman的下载地址为<https://www.getpostman.com/downloads/>。

2.5.2 Git

Git是分布式的版本控制工具，Git的安装方式如下。

Windows：

choco install git

macOS：

$ brew install git

2.6 总结

了解了本书主要涉及的工具后，预先安装好它们，就可以开始Spring Boot开发之旅啦！