

## 第1-3课：Spring Boot 依赖环境和项目结构介绍

---

使用 Spring Boot 开发项目需要有两个基础环境和一个开发工具，这两个环境是指 Java 编译环境和构建工具环境，一个开发工具是指 IDE 开发工具。

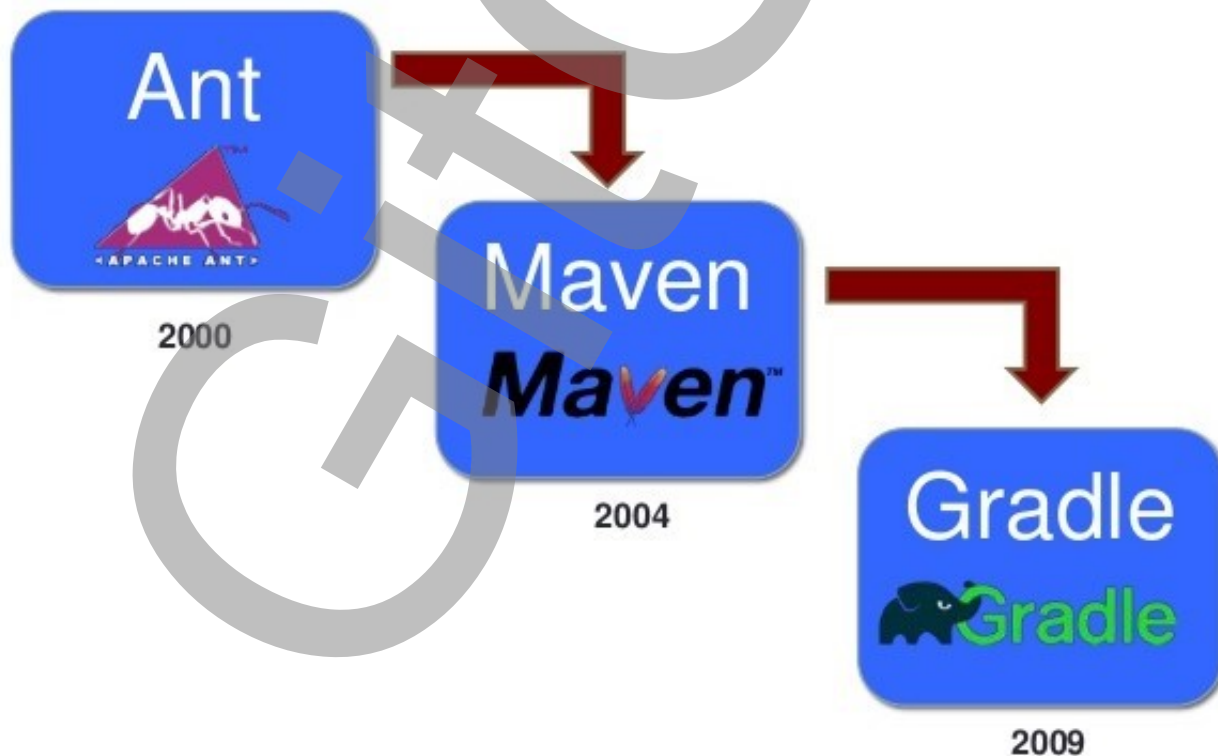
Spring Boot 2.0 要求 Java 8 作为最低版本，需要在本机安装 JDK 1.8 并进行环境变量配置，同时需要安装构建工具编译 Spring Boot 项目，最后准备一个顺手的 IDE 开发工具即可。

构建工具是一个把源代码生成可执行应用程序的自动化工具，Java 领域中主要有三大构建工具：Ant、Maven 和 Gradle。

- Ant (AnotherNeatTool) 的核心是由 Java 编写，采用 XML 作为构建脚本，这样就允许你在任何环境下运行构建。Ant 是 Java 领域最早的构建工具，不过因为操作复杂，慢慢的已经被淘汰了。
- Maven，Maven 发布于 2004 年，目的是解决程序员使用 Ant 所带来的一些问题，它的好处在于可以将项目过程规范化、自动化、高效化以及强大的可扩展性。
- Gradle，Gradle 是一个基于 Apache Ant 和 Apache Maven 概念的项目自动化建构工具。它使用一种基于 Groovy 的特定领域语言来声明项目设置，而不是传统的 XML。结合了前两者的优点，在此基础上做了很多改进，它具有 Ant 的强大和灵活，又有 Maven 的生命周期管理且易于使用。

---

### Build System Evolution



Spring Boot 官方支持 Maven 和 Gradle 作为项目构建工具。Gradle 虽然有更好的理念，但是相比 Maven 来讲其行业使用率偏低，并且 Spring Boot 官方默认使用 Maven，因此本系列课程选择使用 Maven 作为 Spring Boot 项目构建工具。

Java 领域最流行的 IDE 开发工具有 Eclipse 和 IDEA。Eclipse 是 Java 的集成开发环境（IDE），也是 Java 领域最流行的 IDE 开发工具之一，只是 Eclipse 这些年发展缓慢，慢慢的有些陈旧。IDEA（IntelliJ IDEA）是用于 Java 语言开发的集成环境，在业界被公认为是最好的 Java 开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE 支持、创新的 GUI 设计等方面的功能可以说是超常的。因此强烈推荐大家使用 IntelliJ IDEA 开发 Spring Boot 项目。

接下来将介绍如何搭建基础环境以及 IntelliJ IDEA 的安装。

## 安装 JDK

### 下载安装

首先打开 [Oracle 官网 JDK 1.8 下载页面](#)，根据下图选择下载各系统对应的版本，这里以 Win10 64 位操作系统为例。

Overview **Downloads** Documentation Community Technologies Training

## Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u181 checksum

### Java SE Development Kit 8u181

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

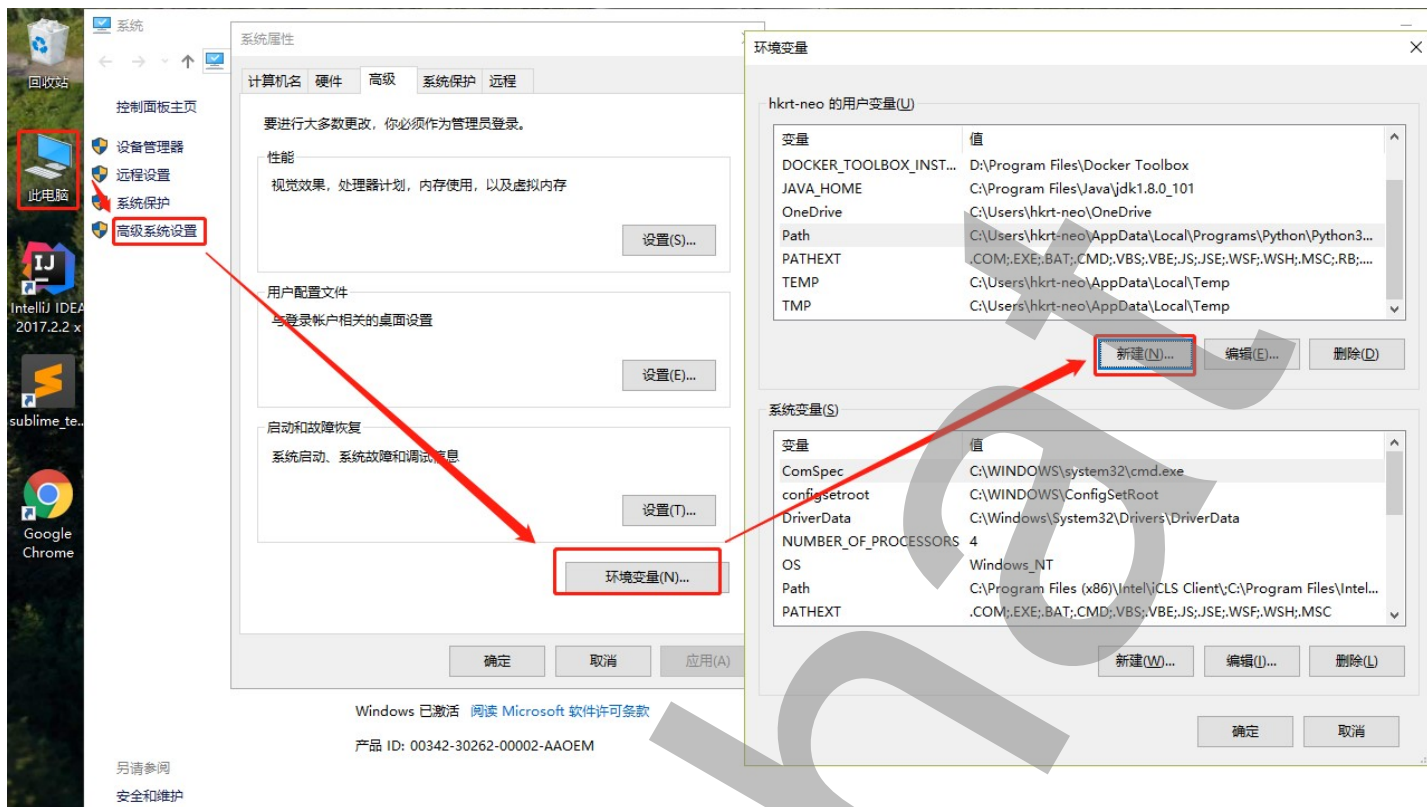
☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.95 MB	<a href="#">jdk-8u181-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.89 MB	<a href="#">jdk-8u181-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	165.06 MB	<a href="#">jdk-8u181-linux-i586.rpm</a>
Linux x86	179.87 MB	<a href="#">jdk-8u181-linux-i586.tar.gz</a>
Linux x64	162.15 MB	<a href="#">jdk-8u181-linux-x64.rpm</a>
Linux x64	177.05 MB	<a href="#">jdk-8u181-linux-x64.tar.gz</a>
Mac OS X x64	242.83 MB	<a href="#">jdk-8u181-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	133.17 MB	<a href="#">jdk-8u181-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	94.34 MB	<a href="#">jdk-8u181-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	93.83 MB	<a href="#">jdk-8u181-solaris-x64.tar.Z</a>
Solaris x64	92.11 MB	<a href="#">jdk-8u181-solaris-x64.tar.gz</a>
Windows x86	194.41 MB	<a href="#">jdk-8u181-windows-i586.exe</a>
Windows x64	202.73 MB	<a href="#">jdk-8u181-windows-x64.exe</a>

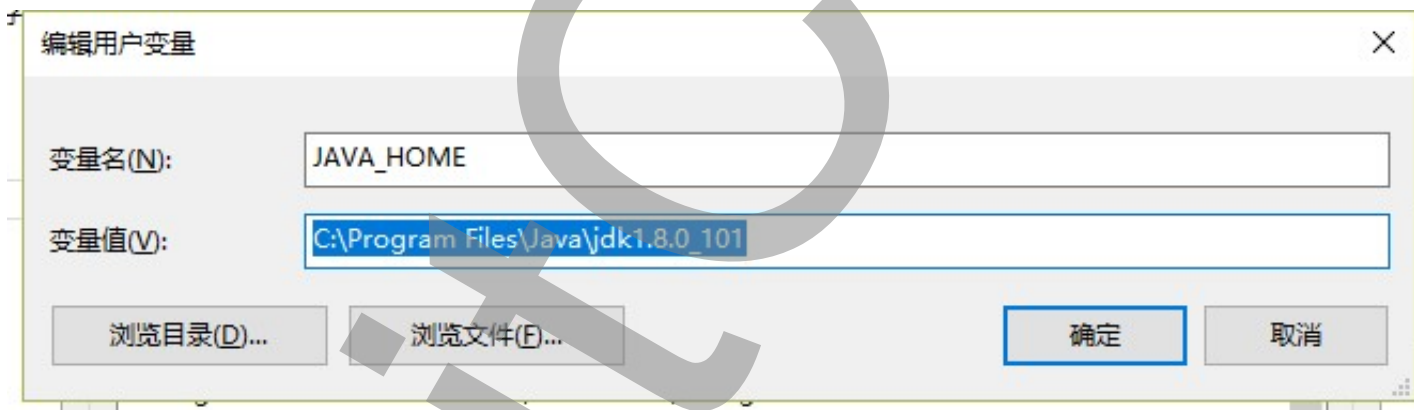
下载完成之后，双击鼠标进行安装，一直单击“下一步”按钮直至安装完毕。

## 环境变量配置

JDK 安装完毕后，接下来配置 JDK 环境变量，选择“我的电脑”|“属性”|“高级系统设置”|“环境变量”|“新建”命令：

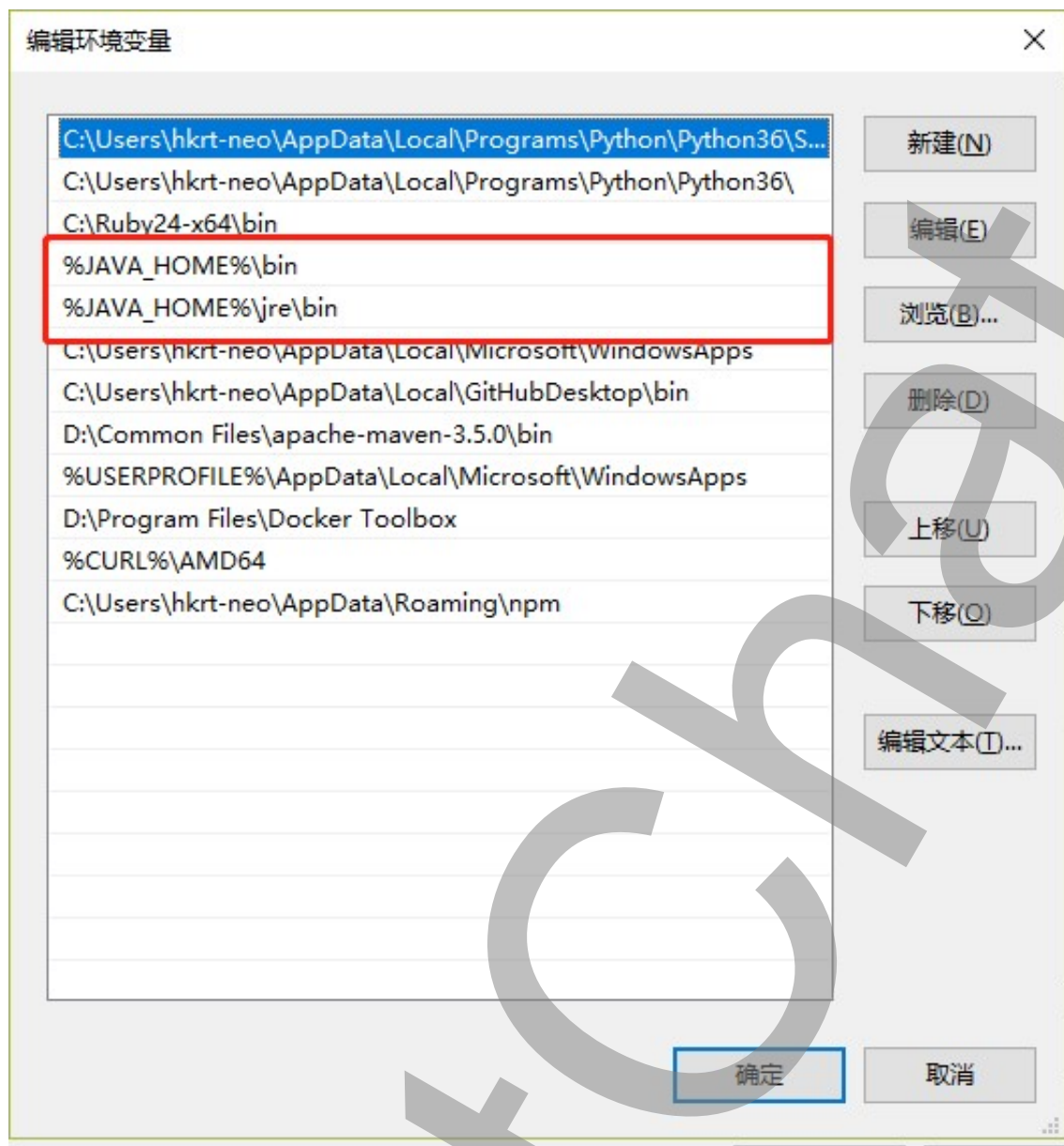


在弹出的对话框中新建 `JAVA_HOME` 变量以及 Java 安装地址，如下图：



单击“确定”按钮后，回到环境变量界面，双击 `Path` 变量，添加两条 JDK 路径，如下：





以上，Java 环境变量配置完毕！

## 测试

配置完成之后，测试一下 JDK 是否配置正确，Win10 下使用快捷 window+r 输入 cmd 命令，进入运行窗口，执行命令 `java -version`，若出现如下结果，则表示安装成功！

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.165]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\hkrt-neo>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)

C:\Users\hkrt-neo>
```

## 安装 Maven

安装 Maven 的前提是完成 Java 环境安装，Maven 依赖于 Java 环境。

## 下载安装

访问 [Maven 官网](#) 下载 Maven 对应的压缩包，如下：

### Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	<a href="#">apache-maven-3.5.4-bin.tar.gz</a>	<a href="#">apache-maven-3.5.4-bin.tar.gz.sha1</a>	<a href="#">apache-maven-3.5.4-bin.tar.gz.asc</a>
Binary zip archive	<a href="#">apache-maven-3.5.4-bin.zip</a>	<a href="#">apache-maven-3.5.4-bin.zip.sha1</a>	<a href="#">apache-maven-3.5.4-bin.zip.asc</a>
Source tar.gz archive	<a href="#">apache-maven-3.5.4-src.tar.gz</a>	<a href="#">apache-maven-3.5.4-src.tar.gz.sha1</a>	<a href="#">apache-maven-3.5.4-src.tar.gz.asc</a>
Source zip archive	<a href="#">apache-maven-3.5.4-src.zip</a>	<a href="#">apache-maven-3.5.4-src.zip.sha1</a>	<a href="#">apache-maven-3.5.4-src.zip.asc</a>

- [Release Notes](#)
- [Reference Documentation](#)
- [Apache Maven Website As Documentation Archive](#)
- All current release sources (plugins, shared libraries,...) available at <https://www.apache.org/dist/maven/>
- [latest source code from source repository](#)
- Distributed under the [Apache License](#), version 2.0

### Previous Releases

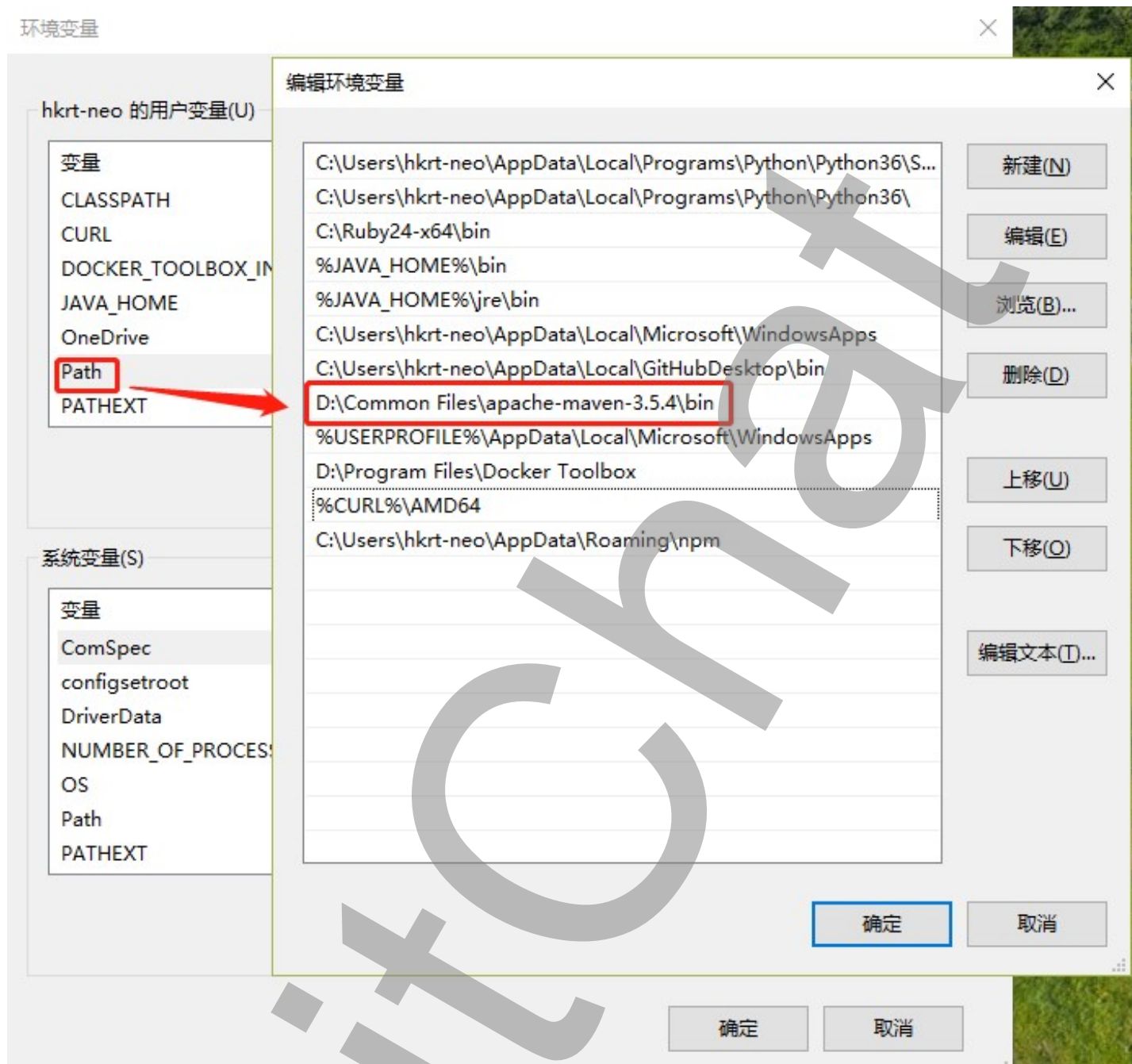
选择 Maven 的 zip 压缩包（`apache-maven-3.5.4.zip`），下载完成后解压到本机目录下。例如，路径：`D:\Common Files\apache-maven-3.5.4`：

此电脑 > Program (D:) > Common Files > apache-maven-3.5.4 >			▼ ↻
名称	修改日期	类型	
bin	2018/9/18 19:06	文件夹	
boot	2018/9/18 19:06	文件夹	
conf	2018/9/18 19:06	文件夹	
lib	2018/9/18 19:06	文件夹	
LICENSE	2017/4/3 20:41	文件	
NOTICE	2017/4/3 20:40	文件	
README.txt	2017/4/3 20:36	EditPlus	

Maven 为绿色软件解压后即可使用。

## 环境变量配置

按照上面步骤打开环境变量设置页面，双击 Path 变量，将上一步解压的目录添加到 Path 中。



以上，Maven 环境配置完毕！

## 测试

Win10 下使用快捷 window+r 输入 cmd 命令，弹出“运行”对话框，执行命令 mvn -v，若出现如下结果，则表示安装成功！



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.285]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\hkrt-neo>mvn -v
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-18T02:33:14+08:00)
Maven home: D:\Common Files\apache-maven-3.5.4\bin\..
Java version: 1.8.0_101, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_101\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

## settings.xml 设置

Maven 解压后目录下会有一个 settings.xml 文件，位置：`${M2_HOME}/conf/settings.xml`，用来配置 Maven 的仓库和本地 Jar 包存储地址。Maven 仓库地址代表从哪里去下载项目中的依赖包 Jar 包；Maven 会将所有的 Jar 包统一存储到一个地址下，方便各个项目复用。

localRepository 设置本地存放 Jar 包地址，可以根据自己的情况改动：

```
<localRepository>D:\Common Files\maven\repository</localRepository>
```

mirrors 为仓库列表配置的下载镜像列表：

```
<mirrors>
  <mirror>
    <id>repo2</id>
    <mirrorOf>central</mirrorOf>
    <name>spring2.0 for this Mirror.</name>
    <url>https://repo.spring.io/libs-milestone</url>
  </mirror>
  ...
</mirrors>
```

为了方便大家使用，我已经配好了一份 settings.xml 模板，可下载后直接覆盖默认的 settings.xml 文件，覆盖完成后需要修改 localRepository 路径。

[模板 settings.xml 文件地址，详见这里。](#)

## IntelliJ IDEA 安装

### 下载

打开 IntelliJ IDEA 2018 [官方下载地址](#)，IDEA 分为两种版本，即社区版和商业版，商业版是付费的，其功能多，社区版是免费的，功能相对较少。



中文服务  
支付宝

JetBrains 针对个人开发者及企业组织提供不同的授权方式。我们提供多种付款方式，包含银联卡、各种国际信用卡、PayPal、支付宝和微信支付。企业用户也可通过银行汇款的方式完成付款。若您有任何购买或授权上的疑问，欢迎联系我们的中文销售代表为您服务。  
立即联系 [中文销售代表](#)



Version: 2018.2.3  
Build: 182.4323.46  
Released: September 4, 2018  
[Release notes](#)

[System requirements](#)

## Download IntelliJ IDEA

Windows

macOS

Linux

### Ultimate

Web, mobile and enterprise development

DOWNLOAD

.EXE

Free trial

### Community

Java, Groovy, Scala and Android development

DOWNLOAD

.EXE

Free, open-source

## 安装

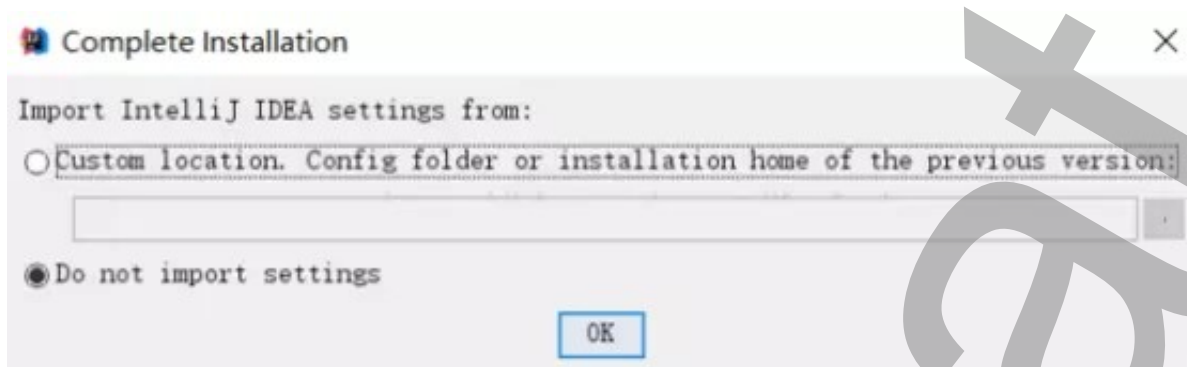
下载完成后，双击安装包开始安装，一直单击 Next 按钮：



安装过程中会选择安装路径，展示 32 或者 64 位的启动快捷键，最后单击 Finish 按钮，安装完成。

## 配置

安装完成后，第一次会提示选择配置设置，选择下面一个，单击 OK 按钮。



后面选择同意协议，输入“License server”激活软件，根据自己的习惯选择白色或者黑色的默认主题，然后选择 IntelliJ IDEA 支持的扩展功能，也可以使用默认选项，然后直接单击“下一步”按钮，最后单击 Start using IntelliJ IDEA 按钮完成配置。

以上，开发工具安装完毕！

## 构建项目

我们有两种方式来构建 Spring Boot 项目基础框架，第一种是使用 Spring 官方提供的构建页面；第二种是使用 IntelliJ IDEA 中的 Spring 插件来创建。

### 使用 Spring 官方提供页面构建

- 访问 <http://start.spring.io/> 网址。
- 选择构建工具 Maven Project，编程语言选择 Java、Spring Boot 版本 2.0.5 以及一些工程基本信息，具体可参考下图。

# SPRING INITIALIZR bootstrap your application now

Generate a Maven Project with Java and Spring Boot 2.1.0 (SNAPSHOT)

## Project Metadata

Artifact coordinates

Group

com.neo

Artifact

hello

## Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, DevTools...

Selected Dependencies

Web

DevTools

Generate Project alt + ⌘

Don't know what to look for? Want more options? [Switch to the full version.](#)

start.spring.io is powered by [Spring Initializr](#) and [Pivotal Web Services](#)

- 单击 Generate Project 下载项目压缩包。
- 解压后，使用 IDEA 导入项目，选择 File | New | Model from Existing Source.. | 选择解压后的文件夹 | OK 命令，选择 Maven，一路单击 Next 按钮，OK done!
- 如果使用的是 Eclipse，选择 Import | Existing Maven Projects | Next | 选择解压后的文件夹 | Finish 按钮，OK done!

## 使用 IDEA 构建

- 选择 File | New | Project... 命令，弹出新建项目的对话框。
- 选择 Spring Initializr，Next 也会出现上述类似的配置界面，IDEA 帮我们做了集成。

**Project Metadata**

Group:

Artifact:

Type:

Language:

Packaging:

Java Version:

Version:

Name:

Description:

Package:

- 填写相关内容后，单击 Next 按钮，选择依赖的包再单击 Next 按钮，最后确定信息无误单击 Finish 按钮。

对上面的配置做如下解释。

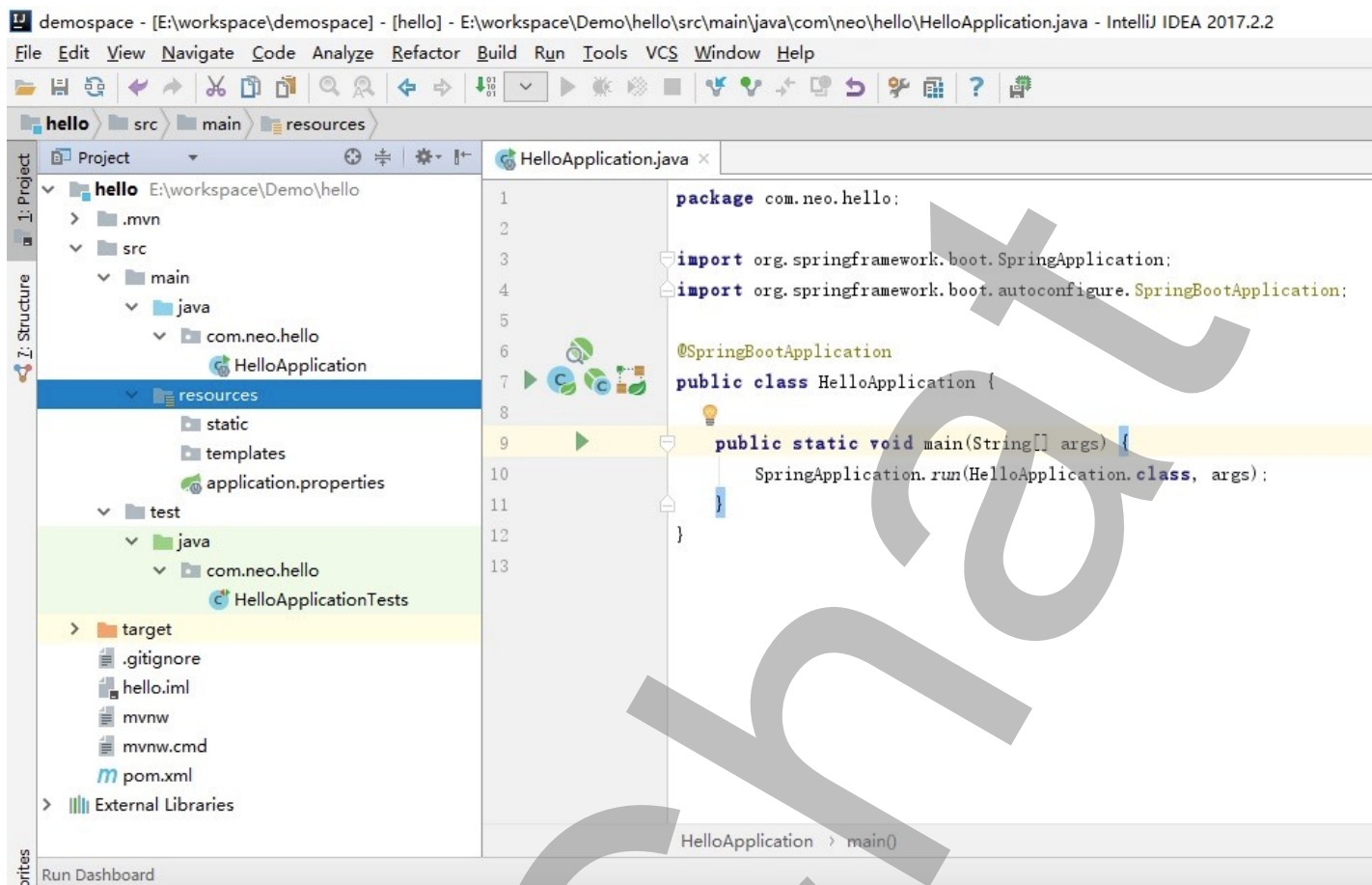
- 第一个选择框选择创建以 Maven 构建项目，还是以 Gradle 构建项目，这是两种不同的构建方式，其中 Gradle 配置内容更简洁一些，并且包含了 maven 的使用，但我们日常使用 maven 居多。
- 第二个选择框选择编程语言，现在支持 Java、Kotlin 和 Groovy。
- 第三个选择框选择 Spring Boot 版本，可以看出 Spring Boot 2.0 的最新版本是 2.0.5。

下面就是项目的配置信息了。

- Group，一般填写公司域名，比如百度公司就会填：com.baidu，演示使用 com.neo。
- Artifact，可以理解为项目的名称了，可以根据实际情况来填，本次演示填 hello。
- Dependencies，在这块添加我们项目所依赖的 Spring Boot 组件，可以多选，本次选择 Web、Devtools 两个模块。

## 项目结构介绍





如上图所示，Spring Boot 的基础结构共三个文件，具体如下：

- `src/main/java`：程序开发以及主程序入口；
- `src/main/resources`：配置文件；
- `src/test/java`：测试程序。

另外，Spring Boot 建议的目录结构如下。

`com.example.myproject` 目录下：

```
myproject
+-src
  +- main
    +- java
      +- com.example.myproject
        +- comm
        +- model
        +- repository
        +- service
        +- web
        +- Application.java
    +- resources
      +- static
      +- templates
      +- application.properties
  +- test
+-pom.xml
```

com.example.myproject 目录下：

- Application.java，建议放到根目录下面，是项目的启动类，Spring Boot 项目只能有一个 main() 方法；
- comm 目录建议放置公共的类，如全局的配置文件、工具类等；
- model 目录主要用于实体（Entity）与数据访问层（Repository）；
- repository 层主要是数据库访问层代码；
- service 层主要是业务类代码；
- web 层负责页面访问控制。

resources 目录下：

- static 目录存放 web 访问的静态资源，如 js、css、图片等；
- templates 目录存放页面模板；
- application.properties 存放项目的配置信息。

test 目录存放单元测试的代码；pom.xml 用于配置项目依赖包，以及其他配置。

采用默认配置可以省去很多设置，也可以根据公司的规范进行修改，至此一个 Java 项目搭建好了！

## Pom 包介绍

pom.xml 文件主要描述了项目包的依赖和项目构建时的配置，在默认的 pom.xml 包中分为五大块。

第一部分为项目的描述信息：

```
<groupId>com.neo</groupId>
<artifactId>hello</artifactId>
<version>2.0.5.RELEASE</version>
<packaging>jar</packaging>

<name>hello</name>
<description>Demo project for Spring Boot</description>
```

- groupId, 项目的包路径;
- artifactId, 项目名称;
- version, 项目版本号;
- packaging, 一般有两个值: jar、war, 表示使用 Maven 打包时构建成 Jar 包还是 War 包;
- name, 项目名称;
- description, 项目描述。

第二部分为项目的依赖配置信息:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.5.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

- parent, 标签内配置 Spring Boot 父级版本 spring-boot-starter-parent, Maven 支持项目的父子结构, 引入父级后会默认继承父级的配置;
- dependencies, 标签内配置项目所需要的依赖包, Spring Boot 体系内的依赖组件不需要填写具体版本号, spring-boot-starter-parent 维护了体系内所有依赖包的版本信息。

第三部分为构建时需要的公共变量：

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
  <java.version>1.8</java.version>
</properties>
```

上面配置了项目构建时所使用的编码，输出所使用的编码，最后指定了项目使用的 JDK 版本。

第四部分为构建配置：

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

使用 Maven 构建 Spring Boot 项目必须依赖于 spring-boot-maven-plugin 组件，spring-boot-maven-plugin 能够以 Maven 的方式为应用提供 Spring Boot 的支持，即为 Spring Boot 应用提供了执行 Maven 操作的可能。spring-boot-maven-plugin 能够将 Spring Boot 应用打包为可执行的 jar 或 war 文件，然后以简单的方式运行 Spring Boot 应用。

以上即为 pom.xml 文件基础内容，几乎所有的 Spring Boot 项目都会用到以上配置信息。

## 总结

这一课我们介绍了 Spring Boot 所依赖的基础环境，如何去搭建 JDK、Maven 环境，安装开发工具 IDEA；对 Spring Boot 项目结构进行了解读，介绍了 pom.xml 文件内容的含义。通过本课的学习，我们发现构建 Spring Boot 项目更简单方便，相比传统项目，Spring Boot 项目配置更加灵活。

[点击这里下载源码。](#)