

★ 服务热线: 400-615-1233
★ 配套精品教学资料包
★ www.huatengedu.com.cn



Spring Boot 项目开发教程

策划编辑: 李 勇
责任编辑: 许 青
封面设计: 张瑞阳

ISBN 978-7-5635-7117-8



定价: 55.00元

北京邮电大学出版社



X-A

Spring Boot 项目开发教程

主编 马力 张娜

主编 马力 张娜
主审 邵 华



“十四五”职业教育国家规划教材



Spring Boot 项目开发教程

Spring Boot XIANGMU KAIFA JIAOCHENG

校企“**二元**”合作开发, 融合岗位要求
引入“**个人应学系统**”, 提高实战技能



北京邮电大学出版社
www.buptpress.com



“十四五” 职业教育国家规划教材



Spring Boot 项目开发教程

主 编 马 力 张 娜

副主编 肖 鑫 张丹丹 杨 贺 王立华

主 审 邵 华



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书采用“任务驱动”的编写模式,由浅入深、循序渐进、全面系统地介绍了 Spring Boot 开发技术。全书共设计 9 个模块,分别为 Spring Boot 开发入门、开发第一个 Spring Boot 程序、Spring Boot 核心配置和注解使用、Spring Boot 与数据访问、Spring Boot 实现 Web 开发、Spring Boot 缓存利器、Spring Boot 安全管理、Spring Boot 日志管理和接口文档、综合项目——个人应学系统。

本书既适合作为高等职业院校计算机类相关专业的教材,也可以作为广大 Spring Boot 开发爱好者的实用参考书。

图书在版编目(CIP)数据

Spring Boot 项目开发教程/马力,张娜主编. -- 北京:北京邮电大学出版社,2023.12(2026.1重印)
ISBN 978-7-5635-7117-8

I. ①S… II. ①马… ②张… III. ①JAVA 语言—程序设计—教材 IV. ①TP312.8

中国国家版本馆 CIP 数据核字(2023)第 247080 号

策划编辑:李 勇 责任编辑:许 青 封面设计:张瑞阳

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号

邮政编码:100876

发 行 部:电话:010-62282185 传真:010-62283578

E-mail: publish@bupt. edu. cn

经 销:各地新华书店

印 刷:河北龙大印务有限公司

开 本:787 mm×1 092 mm 1/16

印 张:18 插页 1

字 数:372 千字

版 次:2023 年 12 月第 1 版

印 次:2026 年 1 月第 3 次印刷

ISBN 978-7-5635-7117-8

定 价:55.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

服务电话:400-615-1233



前言

P R E F A C E

党的二十大报告指出：“加快建设国家战略人才力量，努力培养造就更多大师、战略科学家、一流科技领军人才和创新团队、青年科技人才、卓越工程师、大国工匠、高技能人才。”

当前互联网后端开发中,Java EE 占据了主导地位。对于 Java EE 开发,首选框架是 Spring 框架。在传统的 Spring 开发中,使用大量的 XML 配置才能使 Spring 框架运行起来。而随着 Spring 4. x 的发布,Spring 脱离 XML,只使用注解就可以运行项目了。

为了进一步简化 Spring 的开发,2014 年 Spring Boot 诞生,它是一个由 Pivotal 团队开发的全新框架,其设计目的是简化 Spring 应用的搭建以及开发过程,并适应微服务(将一个系统拆分为多个子系统,然后通过 REST 风格的请求将多个子系统集成起来,进一步简化分布式系统的开发)思维,越来越多的企业选择 Spring Boot。2017 年 9 月 Spring 5. x 推出,2018 年 Spring Boot 推出了 2. x 版本。

Spring Boot 不是代替 Spring,而是使 Spring 项目可以更加快速地开发、部署和测试。它采用了“约定优于配置”的理念,在内部提供了大量的 starter,而这些 starter 又提供了许多自动配置类,让开发者可以开箱即用。虽然这样能够快速地进行开发、部署和测试,但是也会带来很大的问题,如果不懂 Spring 的原理,一旦出现开发的问题,开发者就很容易陷入困境,难以找到问题的根源。所以要学习 Spring Boot 就必须掌握 Spring 的基础知识。基于这种情况,本书结合 Spring 的原理讨论 Spring Boot 的应用。

本书具有以下特点。

(1)编写体例、形式和内容具有职业教育特点。本书采用目标分解“微课”模式进行教学模块的设计和开发,教学模块贴近实际生活,难易适中,有针对性,可以帮助学生加深对企业岗位的认知,完成校企对接。

(2)明确思政目标,强化价值引领。本书通过“素质目标”“明德育才”等栏目挖掘课程中的思想政治教育元素,使在学习过程中不仅能够掌握专业知识,还能够受到思想观念、职业道德、创新意识的浸润与熏陶。



(3)图文结合,突出直观性。本书任务实施过程分步展开,图文并茂,更加突出操作要点,体现直观性和实用性,有利于学生理解过程,增加学习兴趣。

(4)内容安排有层次性。本书内容安排体现了从基本技能到核心技能再到综合技能的培养过程,项目的设置由浅入深、循序渐进,任务的选取考虑课程内容的实用性、专业岗位工作对象的典型性和教学过程的可操作性。

(5)打造新形态一体化教材,实现教学资源共建共享。本书发挥“互联网+教材”的优势,配备二维码学习资源,学生用手机扫描书中二维码即可获取在线数字资源。同时,本书还提供了配套教学课件、课程标准、练习题等供任课教师使用。

阅读本书前,读者需要具备 Java 编程语言基础、Java EE (Servlet 和 JSP) 基础、前端 (HTML、JavaScript 和 JQuery) 基础和数据库 (MySQL 和 Redis) 基础。

本书由沈阳职业技术学院马力和张娜担任主编,沈阳职业技术学院肖鑫、张丹丹、杨贺和东软集团股份有限公司王立华担任副主编,中国计算机学会沈阳分会秘书长邵华主审。以上编审人员不仅有丰富的教学经验和教材编写经验,还有在企事业单位工作的经验。

由于编者水平有限,书中难免存在不当之处,恳请广大读者批评指正。

编 者

目 录

C O N T E N T S

模块 1	Spring Boot 开发入门	1
任务 1	认识 Spring Boot	2
任务 2	工具安装与环境搭建	6

模块 2	开发第一个 Spring Boot 程序	16
任务 1	创建 Spring Boot 项目	17
任务 2	实现单元测试与热部署	24
任务 3	Spring Boot 的依赖管理和自动配置	28

模块 3	Spring Boot 核心配置和注解使用	44
任务 1	属性文件和配置项的使用	45
任务 2	配置加载原理	46
任务 3	自定义配置	49
任务 4	外部化配置的方式	53
任务 5	多环境配置	55



模块4 Spring Boot 与数据访问 58

任务 1	了解 Spring Boot 数据访问	59
任务 2	在 Spring Boot 中使用 JDBC	60
任务 3	Spring Boot 整合 MyBatis	67
任务 4	Spring Boot 整合 Spring Data JPA	74

模块5 Spring Boot 实现 Web 开发 88

任务 1	熟悉 Spring Boot 视图技术	89
任务 2	了解 Spring MVC 支持	98
任务 3	使用过滤器、监听器和拦截器	104
任务 4	文件上传与下载	111
任务 5	Spring Boot 应用的部署和运行	117

模块6 Spring Boot 缓存利器 123

任务 1	启用缓存和配置缓存管理器	124
任务 2	Spring Boot 整合 Redis 缓存实现	134

模块7 Spring Boot 安全管理 143

任务 1	认识 Spring Security	144
任务 2	Spring Security 实战入门	147
任务 3	了解 Spring Security 原理	150
任务 4	自定义用户认证	155
任务 5	自定义用户授权管理	164

模块8 Spring Boot 日志管理和接口文档 172

- 任务 1 整合 Spring Boot 日志 173
- 任务 2 生成接口文档 180

模块9 综合项目——个人应学系统 185

- 任务 1 了解系统 186
- 任务 2 了解项目设计 191
- 任务 3 搭建系统环境 194
- 任务 4 前台 App 模块的实现 200
- 任务 5 后台管理模块的实现 259

参考文献 281



模块 1

Spring Boot 开发入门

随着互联网的兴起, Spring 势如破竹, 占据着 Java 领域轻量级开发的王者地位。随着 Java 语言的发展及市场开发的需求, Spring 推陈出新, 推出了全新的 Spring Boot 框架。Spring Boot 是 Spring 家族的一个子项目, 其设计初衷是简化 Spring 配置, 从而可以轻松构建独立运行的程序, 并极大提高开发效率。接下来, 本模块将从 Spring Boot 开发入门入手, 带领大家正式进入 Spring Boot 框架的学习, 并对 Spring Boot 框架的相关原理进行深入分析。



学习目标

知识目标:

- 了解 Spring Boot 的基础知识。
- 熟悉 IDEA 和 JDK 的安装。
- 了解 Maven 和 Gradle 的基础知识。
- 熟悉 Maven 的安装方法。

能力目标:

- 掌握 Spring Boot 框架的优势及学习的必要性。
- 对 Spring Boot 有初步的认识。

素质目标:

- 基于学生认知规律, 引导学生自主探索, 动手实操。
- 培养学生养成严谨的学习习惯, 在进行程序开发时严谨细心。

明德育才

西方哲学家柏拉图曾说过这样一句话, 良好的开端, 等于成功的一半。这句话虽简单, 却道出了成功的真谛。一个好的开始, 往往能够引导事情走向成功。学习一门课程也是一样, 只有在学习课程初期打下基础, 才会对课程有一个更好的认识, 学习才会越来越有趣味。总之, 这句名言告诉我们要重视每一件事情的开端, 只有在一开始就做好充分的准备和规划, 才有可能在接下来的过程中少走弯路。让我们在生活中和工作时刻铭记这个道理, 不断追求卓越和进步!



任务 1 认识 Spring Boot

任务描述

Spring Boot 是一个综合性的实战型应用技术框架, 是在开发者已经完全掌握 Spring、SSM 开发框架整合基础之上实现的进一步的简化开发模型, 不仅简单易学, 而且属于当前企业应用开发中特别实用的技术之一。本任务将详细讲解 Spring Boot 是什么, Spring、Spring MVC 和 Spring Boot 的基本概念、原理和结构, 以及 Spring Boot 的优势和特点。

任务实施

步骤 1 认识 Spring Boot 是什么

在 Spring Boot 框架出现之前,Java EE 开发最常用的框架是 Spring,该框架兴起于 2003 年,它是由 Rod Johnson 创建的一个轻量级开源框架。Spring 框架是为了解决企业应用开发的复杂性而创建的,它的出现使得开发者无须开发重量级的 Enterprise JavaBean (EJB),而是通过控制反转 (IOC) 和面向切面编程 (AOP) 的思想进行更轻松的企业应用开发,取代了 EJB 臃肿、低效的开发模式。虽然 Spring 框架是轻量级的,但它的配置是重量级的。早期版本的 Spring 专注于 XML 配置,开发一个程序需要配置各种 XML 文件。为了简化开发,在 Spring 2. x 版本开始引入少量的注解,如 @Component、@Service 等,由于支持的注解不是很多且功能尚不完善,所以只能辅助使用。

随着实际生产中敏捷开发的需要,及 Spring 注解的大量出现和功能改进,到了 Spring 4. x 版本基本可以脱离 XML 配置文件进行项目开发,多数开发者也逐渐感受到了基于注解开发的便利性,因此,在 Spring 中使用注解开发逐渐占据了主流地位。与此同时,Pivotal 团队在原有 Spring 框架的基础上通过注解的方式进一步简化了 Spring 框架的使用,并基于 Spring 框架开发了全新的 Spring Boot 框架,于 2014 年 4 月正式推出了 Spring Boot 1.0 版本,在 2018 年 3 月又推出了 Spring Boot 2.0 版本。Spring Boot 2. x 版本在 Spring Boot 1. x 版本的基础上进行了诸多功能的改进和扩展,同时进行了大量的代码重构,所以在学习开发过程中务必注意使用版本,同时也推荐使用优化后的 Spring Boot 2. x 版本。

Spring Boot 框架本身并不提供 Spring 框架的核心特性及扩展功能,只是用于快速、敏捷地开发新一代基于 Spring 框架的应用,并且在开发过程中大量使用“约定优先配置”(convention over configuration)的思想来摆脱 Spring 框架中各种复杂的手动配置,同时衍生出了 Java Config(取代传统 XML 配置文件的 Java 配置类)这种优秀的配置方式。也就是说,Spring Boot 并不是替代 Spring 框架的解决方案,而是和 Spring 框架紧密结合用于提升 Spring 开发者体验的工具,同时 Spring Boot 还集成了大量常用的第三方库配置(如 Jackson、JDBC、Redis、Mail 等)。使用 Spring Boot 开发程序时,几乎是开箱即用(out-of-the-box),大部分 Spring Boot 应用都只需少量的配置,这一特性更能促使开发者专注于业务逻辑的实现。

另外,随着近几年微服务开发的需求和火爆,怎样快速、简便地构建一个准生产环境的 Spring 应用也是摆在开发者面前的一个难题,而 Spring Boot 框架的出现恰好完美地解决了这些问题,同时内部简化了许多常用的第三方库配置,使得微服务开发更加便利,这也间接体现了 Spring Boot 框架的优势和学习 Spring Boot 的必要性。

步骤 2 掌握 Spring、Spring MVC 和 Spring Boot 的基本概念、原理和结构

1. 基本概念

1) Spring 的基本概念

Spring 是一个开源容器框架,可以接管 Web 层、业务层、Dao 层、持久层的组件,并且可以配置各种 Bean 和维护 Bean 与 Bean 之间的关系。其核心就是 IOC 和 AOP,简单地说就是一个分层的轻量级开源框架。

2) Spring MVC 的基本概念

Spring MVC 属于 Spring Framework 的后续产品,已经融合在 Spring Web Flow 里面。Spring MVC 是一种 Web 层 MVC 框架,用于替代 Servlet 处理响应请求、获取表单参数、表单校验等。Spring MVC 是一个 MVC 的开源框架, $\text{Spring MVC} = \text{Struts2} + \text{Spring}$, 即 Spring MVC 相当于 Struts2 加上 Spring 的整合。

3) Spring Boot 的基本概念

Spring Boot 是一个微服务框架,延续了 Spring 框架的核心思想 IOC 和 AOP,简化了应用的开发和部署。Spring Boot 是为了简化 Spring 应用的创建、运行、调试、部署等而出现的,使用它可以做到专注于 Spring 应用的开发,而无需过多关注 XML 的配置。Spring Boot 提供了一堆依赖打包,并已经按照使用习惯解决了依赖问题。

2. 原理和结构

1) Spring 的原理和结构

Spring 为简化开发工作,封装了一系列的开箱即用的组件功能模块,包括 Spring JDBC、Spring MVC、Spring Security、Spring AOP、Spring ORM、Spring Test 等。Spring 的原理结构如图 1-1-1 所示。

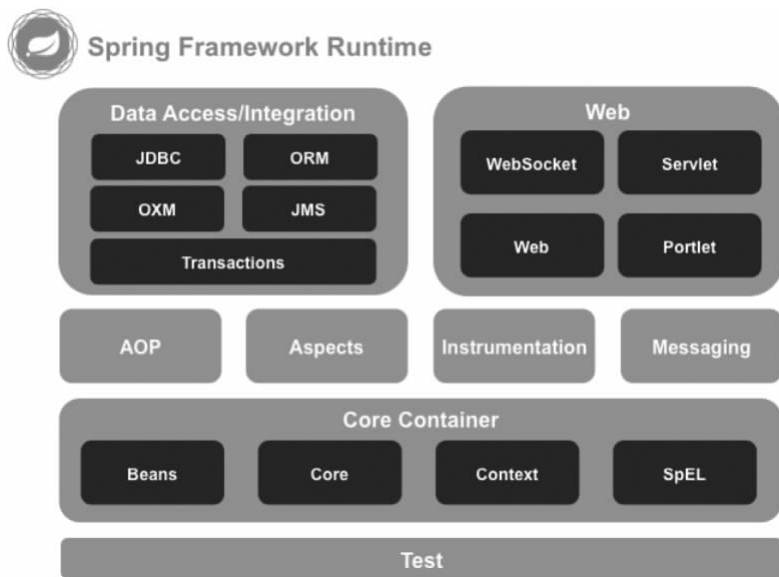


图 1-1-1 Spring 的原理结构

2) Spring MVC 的原理和结构

从图 1-1-1 中可以看出, Spring MVC 属于 Spring Web 中的一个功能模块(Spring Web MVC),是专门用来开发 Spring Web 项目的一种 MVC 模式的技术框架实现。其原理结构如图 1-1-2 所示。

3) Spring Boot 的原理和结构

Spring Boot 基本上是 Spring 框架的扩展,它消除了设置 Spring 应用程序所需的 XML 配置,为更快、更高效地开发生态系统铺平了道路。Spring Boot 的原理结构如图 1-1-3 所示。

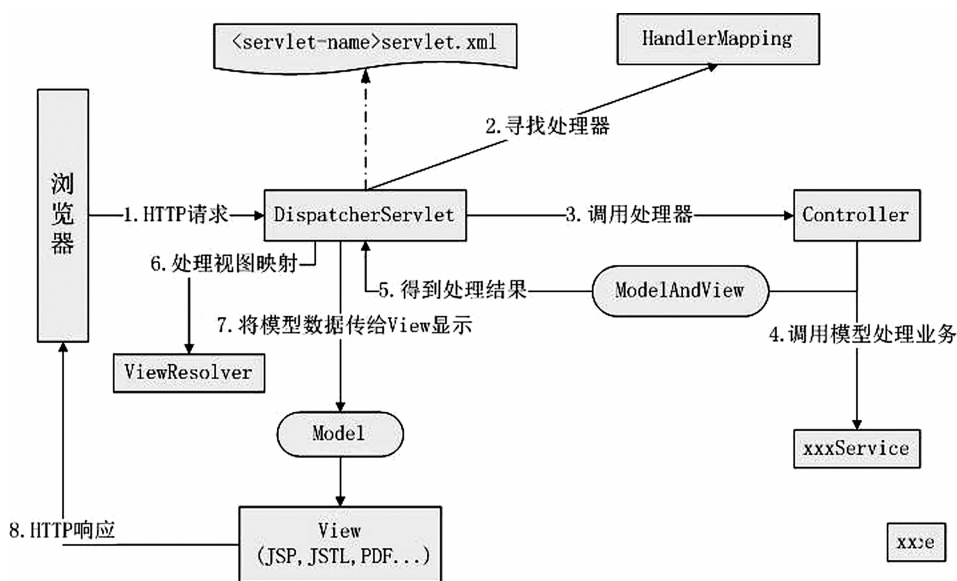


图 1-1-2 Spring MVC 的原理结构

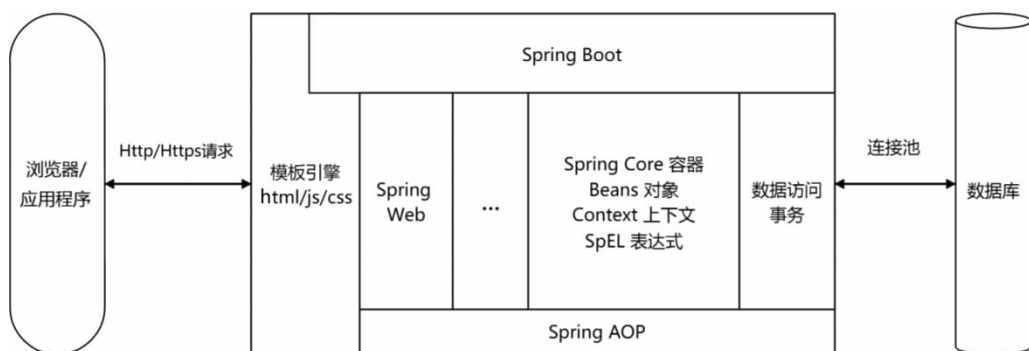


图 1-1-3 Spring Boot 的原理结构

步骤3 了解 Spring Boot 的优势和特点

相较于传统的 Spring 框架, Spring Boot 框架具有以下优势和特点。

1. 可快速构建独立的 Spring 应用

Spring Boot 主要是通过注解和自动化配置的方式推出的全新框架,旨在快速、敏捷地开发新一代基于 Spring 框架的应用程序。在构建 Spring Boot 项目时,只要根据需求选择对应的场景依赖, Spring Boot 就会自动添加该场景所需要的全部依赖并提供自动化配置,在无须额外手动添加配置的情况下可以快速构建出一个独立的 Spring 应用程序。

2. 直接嵌入 Tomcat、Jetty 和 Undertow 服务器(无须部署 WAR 文件)

Spring Boot 项目不需要像传统的 Spring 应用一样打成 WAR 包的形式部署到 Tomcat、Jetty 或 Undertow 服务器中,运行一个 Spring Boot 项目,可以直接将项目打成 JAR 包的形式,并通过命令“`java -jar xx.jar`”运行。这是因为, Spring Boot 项目内嵌了 Tomcat、Jetty 和 Undertow 服务器,因此在部署过程中减少了对第三方插件的依赖和管理。

3. 提供依赖启动器简化构建配置

在 Spring Boot 项目构建过程中,无须准备各种独立的 JAR 文件,只需在构建项目时根据开发场景需求选择对应的依赖启动器“starter”即可,在引入的依赖启动器“starter”内部已经包含了对应开发场景所需的依赖,并会自动下载和拉取相关 JAR 包。例如,在进行 Web 开发时,只需在构建项目时选择对应的 Web 场景依赖启动器 spring-boot-starter-web, Spring Boot 项目便会自动导入 spring-webmvc、spring-web、spring-boot-starter-tomcat 等子依赖,并自动下载和拉取 Web 开发需要的相关 JAR 包。

4. 极大程度地自动化配置 Spring 和第三方库

Spring Boot 充分考虑到了与传统 Spring 框架及其他第三方库融合的场景,在提供了各种场景依赖启动器的基础上,内部还默认提供了各种自动化配置类(如 RedisAutoConfiguration)。使用 Spring Boot 开发项目时,一旦引入了某个场景的依赖启动器, Spring Boot 内部提供的默认自动化配置类就会生效,开发者无须再手动进行配置文件的配置(除非开发者修改了相关默认配置,如 Redis 地址、密码等),从而极大减少了开发人员的工作量,提高了程序的开发效率。

5. 提供生产就绪功能

Spring Boot 提供了一些用于生产环境运行时的特性,如指标、监控检查和外部化配置。其中,指标和监控检查可以很方便地帮助运维人员在运维期间监控项目运行情况;外部化配置可以很方便地让运维人员快速进行外部化配置和部署工作。

6. 极少的代码生成和 XML 配置

Spring Boot 框架内部已经实现了与 Spring 及其他常用第三方库的整合连接,并提供了默认最优化的整合配置,使用时基本上不需要额外生成配置代码和 XML 配置文件。在需要自定义配置的情况下, Spring Boot 更加提倡使用 Java Config 替换传统的 XML 配置方式,这样更加方便查看和管理。

Spring Boot 虽然有诸多优点,但也有一些缺点。例如, Spring Boot 入门较简单,但是深入理解和学习却有一定难度,这是因为 Spring Boot 是在 Spring 框架的基础上推出的,所以想要弄明白 Spring Boot 的底层运行机制,有必要先对 Spring 框架有一定的了解。

► 总结反思

总体来说, Spring Boot 简化了 Spring 应用程序的开发和部署,使开发者能够更专注于业务逻辑而不是配置。它适用于各种类型的应用程序,从传统的 Web 应用到微服务架构。

任务2 工具安装与环境搭建

► 任务描述

在进行 Spring Boot 程序开发之前,需要进行工具安装与环境搭建,本任务主要进行了安装 IDEA 与 JDK、了解 Maven 和 Gradle、安装 Maven 等准备工作。

任务实施

步骤 1 安装 IDEA 与 JDK

1. 下载 IntelliJ IDEA

官方网站提供了两个版本:Community 版和 Ultimate 版。Community 版是免费的,适用于个人和非商业用途。而 Ultimate 版则需要付费购买,提供更多高级功能。初学者建议下载 Community 版,下载地址为 <https://www.jetbrains.com/idea/download/>。

2. 安装 IDEA

下载完成后,会有一个 IntelliJ IDEA 安装包,双击打开下载的安装包,单击“Next”按钮,正式开始安装,如图 1-2-1 所示。



图 1-2-1 开始安装界面

设置 IDEA 的安装目录,建议不要安装在系统盘(通常 C 盘是系统盘),这里选择安装到 D 盘。自行选择需要的功能,若无特殊需求,按图 1-2-2 所示选择即可。

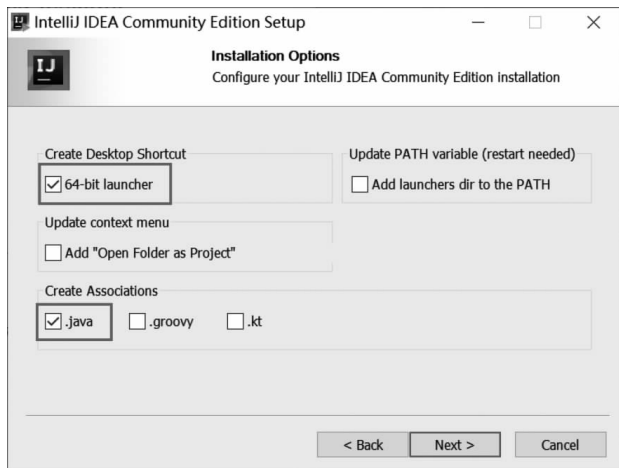


图 1-2-2 自行选择需要的功能

图 1-2-2 中的安装属性配置选项说明如下。

Create Desktop Shortcut: 创建桌面快捷方式图标, 建议选择 64-bit launcher。

Update context menu: 是否将从文件夹打开项目添加至鼠标右键, 根据需要选择。

Create Associations: 关联文件格式, 不推荐选择, 使用如 Sublime Text、EditPlus 等轻量级文本编辑器打开。

Download and install 32-bit JetBrains Runtime: 下载并安装 JetBrains 的 JRE。如果已经安装了 JRE, 则无须选择此项。

Update PATH variable(restart needed): 是否将 IDEA 启动目录添加到环境变量中, 即可以从命令行中启动 IDEA, 根据需要选择。

选择开始菜单文件夹后, 单击“Install”按钮等待安装, 如图 1-2-3 所示。

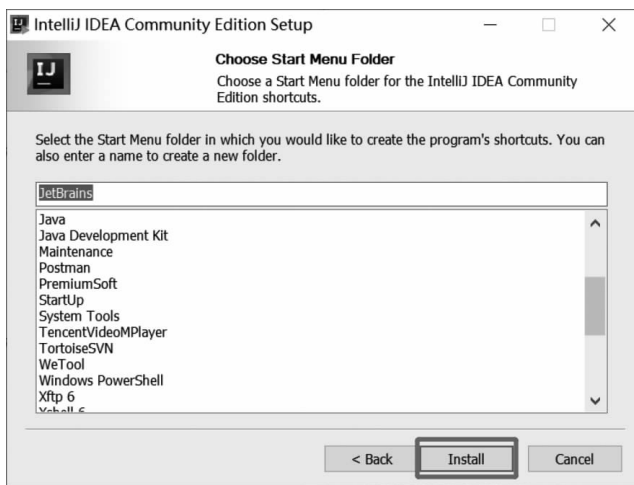


图 1-2-3 开始安装

等待安装进度条达到 100% 后, 单击“Finish”按钮, IDEA 就安装完成了, 如图 1-2-4 所示。可以选择“Run IntelliJ IDEA Community Edition”选项, 则关闭此窗口后将运行 IDEA。



图 1-2-4 安装完成

3. 下载 JDK

打开下载地址 <https://www.oracle.com/java/technologies/downloads/#java8>, 找到自己的计算机系统对应的 JDK, 单击下载链接, 如图 1-2-5 所示。

4. 安装 JDK

(1) 双击下载的 exe 文件, 开始安装, 单击“下一步”按钮, 如图 1-2-6 所示。

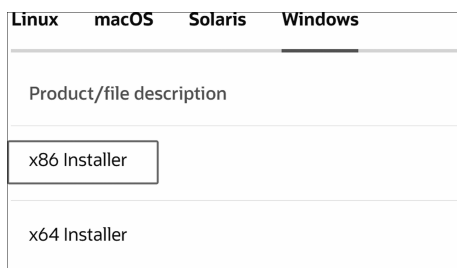


图 1-2-5 JDK 版本选择



图 1-2-6 开始安装 JDK

(2) 更改 JDK 的默认安装目录, 然后单击“下一步”按钮, 如图 1-2-7 所示。



图 1-2-7 修改 JDK 默认安装目录

(3) 安装结束前, 会弹出新窗口, 提示安装 JRE, 如图 1-2-8 所示。



图 1-2-8 安装 JRE

(4)最后弹窗告知已安装成功,如图 1-2-9 所示。



图 1-2-9 JDK 安装完成

小提示:

- 不建议安装到系统盘。一般软件默认安装到 C 盘, C 盘是系统盘, 安装太多软件系统会卡顿(不仅 JDK 的安装如此, 一些其他软件也不建议安装在 C 盘)。
- 不建议安装到中文目录或带空格的目录中。很多国外开发的软件不识别中文路径, 安装在中文目录或带空格的目录中会报错, 以后安装的软件最好也是如此。

5. 配置 JDK 环境变量

(1)在桌面右击“计算机”图标, 并选择“属性”选项, 在打开的窗口中找到“高级系统设置”链接, 如图 1-2-10 所示。

(2)单击“高级系统设置”链接, 打开“系统属性”对话框, 在“高级”选项卡下单击“环境变量”按钮, 如图 1-2-11 所示。



图 1-2-10 高级系统设置



图 1-2-11 单击“环境变量”按钮

(3) 选择系统变量, 单击“新建”按钮, 输入变量名 Java_HOME, 变量值为 JDK 的 bin 目录所在路径, 如图 1-2-12 所示。



图 1-2-12 增加 Java_HOME 系统变量

(4) 选择系统变量, 找到 Path 变量, 双击, 单击“新建”按钮, 输入 %JAVA_HOME%\bin, 然后单击“确定”按钮, 如图 1-2-13 所示。

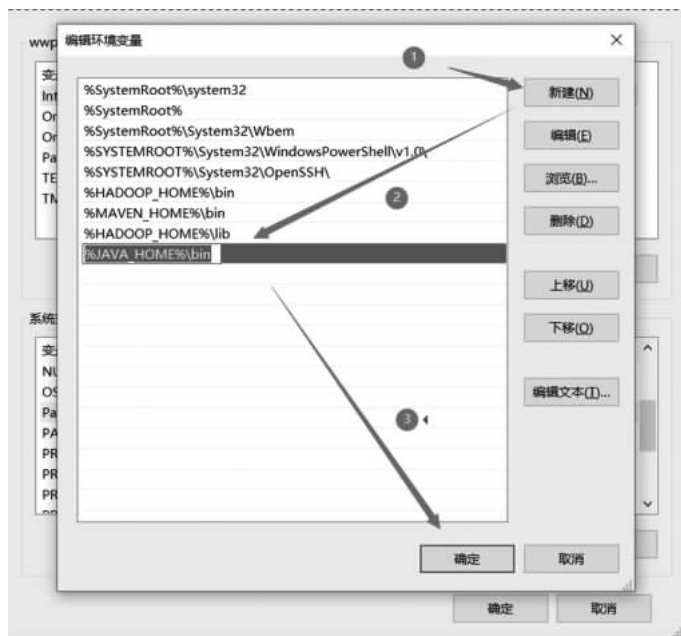


图 1-2-13 Path 变量

(5) 打开控制台窗口, 执行 `java -version` 命令, 若显示当前 Java 版本号, 则代表环境变量配置完成, 如图 1-2-14 所示。



```
C:\Users\14503>java -version
java version "1.8.0_281"
Java(TM) SE Runtime Environment (build 1.8.0_281-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)
```

图 1-2-14 在控制台查看 Java 版本号

步骤 2 了解 Maven 和 Gradle

1. 概念

Maven 是一种基于 XML 的构建工具, 用于管理 Java 项目的构建、依赖关系和发布。Maven 使用传递性依赖关系解决方案, 可使开发人员更轻松地管理项目依赖关系和版本。Maven 使用常见的“坐标”表示法来唯一标识依赖项。

Gradle 是一种基于 DSL(领域特定语言)的构建工具, 它将构建脚本(Groovy 或 Kotlin)编写为构建配置文件。Gradle 的设计目标是可配置性和灵活性, 它通过使用插件提供灵活的构建定制选项。

2. 构建流程和生命周期

Maven 拥有三个标准的生命周期(lifecycle), 最小的运行单元是目标(goal), 并且插件可以把自己的目标绑定在生命周期的某个阶段(phase)上。

Gradle 没有显示的生命周期, 最小的运行单元是任务(task), 任务之间可以相互依赖并且可以动态地创建任务。

3. 包管理和传递性依赖

Maven 的包来源于中央仓库, 一个包由 groupId/artifactId/version 确定唯一坐标, 并且 Maven 的包拥有传递性依赖(当某个包的使用依赖于其他包时, 会自动导入所有的依赖包)。

Gradle 使用 Ivy 的构件系统, 是 Maven 的构件系统的超集。本书采用 Apache Maven 3.8.x 版本进行项目构建管理。

步骤 3 安装 Maven

1. 下载 Maven

打开 Maven 仓库官网: <https://maven.apache.org/download.cgi>, 下载 apache-maven-3.8.6-bin.zip 安装包, 如图 1-2-15 所示。

2. 解压 Maven 并修改配置文件

(1) 将下载的安装包解压到指定路径, 如图 1-2-16 所示。

注意: 一定不能是中文路径。

(2) 在当前目录下创建 MavenRepository 文件夹, 用作 Maven 仓库。需要注意的是, 自定义的 Maven 仓库路径不要设置到 C 盘。

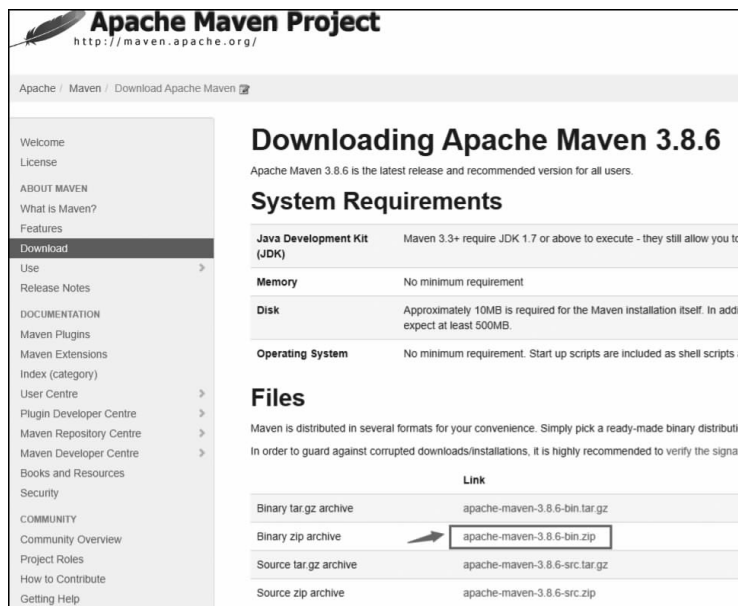


图 1-2-15 下载 Maven

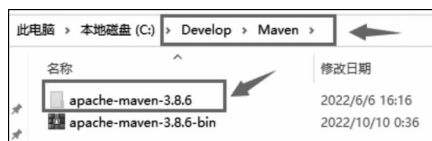


图 1-2-16 解压 Maven

(3)修改 settings.xml 配置文件,配置 Maven 仓库(MavenRepository 文件夹)的绝对路径和阿里云 Maven 公共仓库,如图 1-2-17~图 1-2-19 所示。

①找到并打开 settings.xml 文件。



图 1-2-17 打开 settings.xml 文件

②复制 MavenRepository 文件夹绝对路径,配置 Maven 仓库(注意复制的路径是“\”,要改为“/”),如图 1-2-18 所示。



图 1-2-18 配置仓库目录

③配置阿里云 Maven 公共仓库,如图 1-2-19 所示。

```
<mirrors>
<!-- mirror
| Specifies a repository mirror site to use instead of a g
| this mirror serves has an ID that matches the mirrorOf el
| for inheritance and direct lookup purposes, and must be u
|
-->
<mirror>
<id>mirrorId</id>
<mirrorOf>repositoryId</mirrorOf>
<name>Human Readable Name for this Mirror.</name>
<url>http://my.repository.com/repo/path</url>
</mirror>
-->
<mirror>
<id>aliyunmaven</id>
<mirrorOf>*</mirrorOf>
<name>阿里云公共仓库</name>
<url>https://maven.aliyun.com/repository/public</url>
</mirror>
</mirrors>
```

图 1-2-19 配置阿里云公共仓库地址

3. 配置环境变量

(1)新增系统变量 MAVEN_HOME,如图 1-2-20 所示。

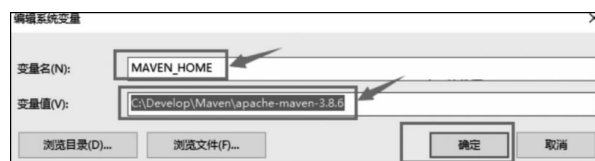


图 1-2-20 新增系统变量 MAVEN_HOME

(2)双击 path 系统变量,单击“新建”按钮,新增环境变量%MAVEN_HOME%\bin,如图 1-2-21 所示。



图 1-2-21 新增 path 变量

总结反思

安装 IDEA 与 JDK 是 Java 开发的第一步。了解 Maven 和 Gradle 有助于更好地管理和构建 Java 项目。安装 Maven 是在 Java 项目中使用依赖管理和构建工具的关键步骤。这些工具将成为日常 Java 开发不可或缺的一部分。



模块小结

本模块主要了解 Spring Boot 开发入门的一些基础知识。首先对 Spring Boot 的概念和特点进行了讲解,让读者快速了解 Spring Boot 框架的优势及学习的必要性;然后进行了 Spring Boot 程序开发前的工具安装与环境搭建工作。通过本模块的学习,相信读者可以对 Spring Boot 有一个初步认识,为后续学习 Spring Boot 做好铺垫。



课后习题

一、选择题

1. (多选)以下选项中,()属于 Spring Boot 的优点。
 - A. 可快速构建独立的 Spring 应用
 - B. 无须打包即可快速部署
 - C. 提供依赖启动器简化构建配置
 - D. 极大程度地自动化配置 Spring 和第三方库
2. 在 IDEA 界面中,Configure→Project Defaults 中的 Project Structure 的主要作用是()。
 - A. 用于全局 JDK 初始化设置
 - B. 用于全局 Maven 初始化设置
 - C. 用于全局运行环境设置
 - D. 以上都不对
3. (多选)下列关于 Spring Boot 项目各个包作用的说法,正确的是()。
 - A. resources 下的 static 中用于存放静态资源文件
 - B. resources 下的 templates 中用于存放模板文件
 - C. application.properties 是项目的全局配置文件
 - D. 以上都正确

二、操作题

根据任务 2 的任务实施步骤进行工具安装与环境搭建。