

第一个 Spring Boot 程序

通过第1章的学习,已经为 Spring Boot 完成了搭建环境的工作。本章将在搭建好的环境上,先介 绍在 Spring 官网上生成一个 Spring Boot 的初始项目文件,再介绍使用 Eclipse 导入、编写、运行和打 包这个 Spring Boot 程序,接着介绍使用测试工具 Postman 测试这个 Spring Boot 程序,而后分别介绍使 用 Eclipse 创建 Spring Boot 项目、使用 IDEA 编写第一个 Spring Boot 程序等内容。为了让读者操作起 来更加方便,本章给出了具有详尽标注的截图。

本章的知识架构及重难点如下。



2.1 使用 Eclipse 编写第一个 Spring Boot 程序

2.1.1 在 Spring 官网上生成初始项目文件

Spring 官方提供了一个自动创建 Spring Boot 项目的网页,可以为程序开发人员省去大量的配置操作。使用该网页创建 Spring Boot 项目的步骤如下。

(1) 打开浏览器,输入网址 https://start.spring.io,在这个页面中填写项目各种配置和基本信息, 具体内容如图 2.1 所示,矩形框里的文字是笔者给出的标注。

在图 2.1 中,含有很多标签,例如 Project、Language、Spring Boot、Project Metadata 下的 Group 等。那么,这些标签表示的含义是什么呢? 各个标签的说明如下:

Spring Boot 从入门到精通

Project 〇 Gradle Proje	创建 Maven 项目 ct ● Maven Project ● Jav	age O Kotlin O Groovy
Spring Boot O 3.0.0 (SNAP O 2.6.14 (SNAP	使用 Java SHOT) 〇 3.0.0 (RC2) 〇 2.7. PSHOT) 〇 2.6.13	语言 6 (SNAPSHOT)
Project Metada	ta	的稳定版本
Group	com.mr 开发团队的唯一	一标识
Artifact	MyFirstSpringBootProject	项目的唯一ID
Name	MyFirstSpringBootProject	项目名称
Description	Demo project for Spring Boot	项目描述
Package name	com.mr	指定底层包
Packaging	● Jar War	项目的打包方式为 jar 包
Java	● 19 ○ 17 ○ 11 ○ 8	使用的是 JDK 19

图 2.1 填写 Spring Boot 项目的相关内容

- ☑ Project: 表示创建什么类型的项目。书中使用 Maven 作为项目构建工具,所以这里选择 Maven Project,也就是 Maven 项目。
- ☑ Language: 表示使用哪种开发语言。这里选择 Java。
- ☑ Spring Boot: 表示使用哪个版本的 Spring Boot。SNAPSHOT 表示仍在开发过程中的试用版, RELEASE 表示稳定版,表单中未做任何标注的版本则认为是 RELEASE 稳定版,因此选择最新的稳定版本 2.7.5。

说明

Spring 官方一直在不断更新 Spring Boot, 读者打开网站时看到的稳定版本可能会高于 2.7.5, 可以下载最新的稳定版本。

- ☑ Project Metadata 下的 Group:这是开发团队或公司的唯一标志。命名规则通常为团队/公司主页域名的转置,例如域名为 www.mr.com, Group 就应该写成 com.mr,忽略域名前缀。
- ☑ Project Metadata 下的 Artifact: 表示项目的唯一 ID。因为同一个团队下可能有多个项目,这个 ID 就是用来区分不同项目的。图中填写的是 MyFirstSpringBootProject。
- ☑ Project Metadata 下的 Name:项目的名称,也是导入 Eclipse 之后看到的项目名。图中填写的 是 MyFirstSpringBootProject。
- ☑ Project Metadata 下的 Description:项目的描述。对于学习者来说使用默认值即可。
- ☑ Project Metadata 下的 Package name:用于指定 Spring Boot 的底层包,也就是 Spring Boot 启动 类所在的包。图中填写的是 com.mr。
- ☑ Project Metadata 下的 Packaging:表示项目以哪种格式打包。项目如果打包成 jar 包,可以直接在 JRE 环境中启动运行;如果打包成 War 文件,可以直接部署到服务器容器中。这里推荐大家打包成 jar 包,便于学习。

☑ Project Metadata 下的 Java: 用于指定项目使用哪个版本的 JDK, 这里选择 JDK 19。

(2) 因为 Spring Boot 主要用于 Web 项目开发,所以还需要给项目添加 Web 依赖。单击页面右侧的 ADD DEPENDENCIES 按钮,列出可选的依赖项。按钮的位置如图 2.2 所示。

💋 spri	Spring initializr					
Project	Language	⊖ Kotlin ⊖ Groovy	Dependencies	ADD DEPENDENCIES CTRL + B		
Spring Boot		<u> </u>	No dependency selected			
O 3.0.0 (SNAP O 2.6.14 (SNA	SHOT) (3.0.0 (RC2)) (2.7.6 (SM PSHOT) (2.6.13)	NAPSHOT)				
Project Metada	ita					
Group	com.mr					
Artifact	MyFirstSpringBootProject					
Name	MyFirstSpringBootProject					
Description	Demo project for Spring Boot					
Package name	com.mr					
Packaging	• Jar O War					
Java	● 19 O 17 O 11 O 8					

图 2.2 为项目添加依赖

(3) 在列出的如图 2.3 所示的依赖项中,找到并单击 Spring Web 选项。

Web, Security, JPA, Actuator, Devtools	Press Ctrl for multiple adds
DEVELOPER TOOLS	
GraalVM Native Support Support for compiling Spring applications to native executables using the compiler.	e GraalVM native-image
Spring Boot DevTools Provides fast application restarts, LiveReload, and configurations for enl experience.	hanced development
Lombok Java annotation library which helps to reduce boilerplate code.	
Spring Configuration Processor Generate metadata for developers to offer contextual help and "code co with custom configuration keys (ex.application.properties/.yml files).	mpletion" when working
WEB	
Spring Web Build web, including RESTful, applications using Spring MVC. Uses Apa default embedded container.	iche Tomcat as the 🛛 🛶
Spring Reactive Web Build reactive web applications with Spring WebFlux and Netty.	
Spring for GraphQL	

图 2.3 选择 Spring Web 依赖

(4) 完成以上操作,在如图 2.4 所示的页面的右侧可以看见已经添加的 Spring Web 依赖。单击页面下方的 GENERATE 按钮,下载自动生成的项目压缩包。

💋 sprii	Spring initializr					
Project	et 🔿 Mayon Draiget	Language		Dependencies	ADD DEPENDENCIES CTRL + B	
Spring Boot 3.0.0 (SNAP 2.6.14 (SNAI	SHOT) O 3.0.0 (RC2) PSHOT) O 2.6.13	O 2.7.6 (SNAPSHOT) 0 2.7.5		Spring Web WEB Build web, including RESTful, applications u as the default embedded container.	sing Spring MVC. Uses Apache Tomcat	
Project Metada	ta					
Group	com.mr		_			
Artifact	MyFirstSpringBootProject		_			
Name	MyFirstSpringBootProject		_			
Description	Demo project for Spring Bo	ot	_			
Package name	com.mr		_			
Packaging	Jar O War					
Java	● 19 O 17 O 11	0 8				
		GENERATE CTRL + ∅	EXPLORE	E CTRL + SPACE SHARE		

图 2.4 生成并下载初始项目

(5)如图 2.5 所示,在 D 盘根目录下,新建一个名为 SpringBootProject 的文件夹,将下载好的压 缩包解压到该文件夹中,这样就完成了初始项目的准备工作。



图 2.5 解压项目压缩包

2.1.2 Eclipse 导入 Spring Boot 项目

Eclipse 支持导入 Maven 项目,不过导入 Maven 项目的方式与导入普通 Java 项目不太一样,本节 将演示 Eclipse 导入 Maven 项目的步骤。

(1) 依次选择 File/Import 菜单,如图 2.6 所示。在导入的类型中,选择 Maven 菜单下的 Existing Maven Projects 子菜单,单击 Next 按钮,如图 2.7 所示。

Ô	eclipse-workspace - Eclipse IDE	🖾 Import 📃
<u>F</u> ile	<u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un	Select
	New Alt Open File	Import existing Maven projects
1	Open Projects from File System Recent Files	Select an import wizard:
	Close Editor Close All Editors Ctrl	by bitter text by bitter text comparison of the text comparison of the text comparison of text compa
	Save Save As Save All Ctr Revert	Maven Maven Check out Maven Projects from SCM Destrict Material Projects Josef Install or deploy an artifact to a Maven repository
1	Move Rename Refresh Convert Line Delimiters To	Materialize Mayer Binary Project Materialize Mayer Projects from SCM > > > Oomph Diver in Development
Ð	Print	
1 1	Import	? < <u>B</u> ack <u>N</u> ext > <u>F</u> inish Canc

图 2.6 导入菜单

图 2.7 选择已存在 Maven 项目

(2) 在弹出的对话框中,单击右侧的 Browse 按钮,如图 2.8 所示。找到上一节下载并解压完毕的项目目录,项目目录确认成功后,单击下方的 Finish 按钮完成导入,如图 2.9 所示。

Import Maven Projects		– 🗆 X	Import Maven Projects	– 🗆 X
Maven Projects	畄土 Drama 挖闭		Maven Projects	
Select Maven projects	毕击 Browse 按钮		Select Maven projects	
Root Directory:		✓ <u>B</u> rowse	<u>Root Directory</u> : D:\SpringBootProject\MyFirstSpringBootProject	Browse
Projects:			Projects:	
		Select <u>A</u> ll	/pom.xml com.mr:MyFirstSpringBootProject:0.0.1-SNAPSHOT:jar	Select <u>A</u> ll
		Deselect All		Deselect All
		Select Tree		Select Tree
		Deselect Tree		Deselect Tree
		<u>R</u> efresh		<u>R</u> efresh
Add project(s) to working set			Add project(s) to working set	
		~	MyFirstSpringBootProject	~
Adyanced			,Advanced 单击 Finish 按钮	
? < <u>B</u> a	ck <u>N</u> ext > <u>F</u> inish	Cancel	(?) < Back Next > Einish	Cancel

图 2.8 找到项目所在目录

图 2.9 确认导入

(3) 导入完成后, Eclipse 会自动启动 Maven 下载 jar 包的操作,此时 Eclipse 右下方出现一个滚动条,显示下载进度,程序开发人员可以单击滚动条右侧的图标查看下载明细,如图 2.10 所示。 导入之后的项目结构如图 2.11 所示。

Spring Boot 从入门到精通

💓 eclipse-workspace - Eclipse IDE		- 🗆 X
<u>File E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject	<u>R</u> un <u>W</u> indow <u>H</u> elp	
	2. ©. R = 31 © : ☆ • O • Q • 9	L ▼ = ▼
월 · 웹 · 약 약 수 · 수 · Id		Q. 18 1
Project Explorer X		P 0
ি ঝি ি 8 > ₩ MyFirstSpringBootProject		
	Reads V In U.S.	
	No consoles to display at this time.	
		Mayon 白动下栽佐
		Maven H 40 F 40 K
		赖的 jar 包的进度
0 items selected		Importing Maven projects: (99%) 🚺 🖷 🖓
	图 2.10 Maven 自动	力下载 jar 包
V 🗁 MyFirstSpringBootPr	roject	项目名
✓ /曲 src/main/iava -		
∠ # com.mr		
> D MyEirstSprin	ngBootProjectApplication.java	— 项目的启动举
✓ /曲 src/main/resource	es	资源句代码引用的资源都故这里
i⊜ static		— 静态资源句,存放图片,文件等
🕞 templates —		
application.pro	operties	— Spring Boot默认配置文件
✓ i src/test/iava —		—— 测试包,单元测试的Java文件都放这里
× ⊕ com.mr		Source (Providence of the source of
> D MyEirstSprin	a BootProjectApplicationTests java	自动生成的单元测试举
> =\ JRE System Librar	v	项目使用的Java运行环境
		Marrangel)的低声外部仿晶
> >> Maven Dependen	cies	
> 🖹 Maven Dependen	icies	—— 源码包和资源包的另一种展示形式
> 🖹 Maven Dependen > 🗁 src	icies —	—— 派码包和资源包的另一种展示形式 —— 项目打完包后存放的位置
> 🖹 Maven Dependen > 🗁 src	icies	— 顶码包扣资源包的另一种展示形式 — 项目打完包后存放的位置 — 自动生成的帮助文档,可以忽略
> 🛋 Maven Dependen > 🗁 src — > 🗁 target — W HELP.md —	cies	— 派码包扣资源包的另一种展示形式 — 项目打完包后存放的位置 — 自动生成的帮助文档,可以忽略

图 2.11 初始项目的文件结构

构建项目的核心配置文件

2.1.3 编写简单的跳转功能

m pom.xml

Spring Boot 自带 Tomcat 容器,无须部署项目就可以直接启动 Web 服务。下面将演示如何编写一个简单的跳转功能,当用户访问一个网址后,页面会展示一段程序开发人员自己编写的文字。

(1) 首先在 com.mr 包下创建子包 controller, 然后在该子包中创建名为 HelloController 的类, 如 图 2.12 所示。





(2) 打开该类文件,补充以下代码:

```
package com.mr.controller;
```

import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RestController;

```
@RestController
public class HelloController {
    @RequestMapping("hello")
    public String sayHello() {
        return "你好,这是我的第一个 Spring Boot 项目";
    }
}
```

(3)运行 com.mr 包下的 MyFirstSpringBootProjectApplication.java 文件(即项目的启动类),可以 在控制台中看到如图 2.13 所示的启动日志。其中,日志 "Started MyFirstSpringBootProjectApplication in 3.045 seconds"表示项目启动成功,耗时 3.045 秒,这样就可以在浏览器里访问 Web 服务了。

				-
📓 Markers 📖 Properties 🕫 Servers 📲 Data Source Exp	plorer 🖆 Snippets 🖉 Terminal 🖳 Console 🗙			3
MyFirstSpringBootProjectApplication [Java Application] D:\Jav	va\jdk-19\jdk-19.0.1\bin\javaw.exe (2022年11月12日 上午)	10:20:15) [pid: 10592]		
				~
' _ _ _/ / / / /				
:: Spring Boot :: (v2.7.5)				
2022-11-12 10:20:20.686 INFO 10592 [main] c.m.MyFirstSpringBootProjectApplication	 Starting MyFirstSpringBootProjectApplication using Java 19.0.1 on JIS-20180106EA 	X with	
2022-11-12 10:20:20.690 INFO 10592 [main] c.m.MyFirstSpringBootProjectApplication	: No active profile set, falling back to 1 default profile: "default"		
2022-11-12 10:20:22.310 INFO 10592 [mainj o.s.p.w.embedded.tomcat.lomcatwebServer	: Tomcat Initialized with port(s): 8080 (http)		
2022-11-12 10:20:22.313 INFO 10592 [mainj o.a.catalina.core.AprLitecycleListener	: Loaded Apache Tomcat Native Library [1:2:33] Using APK Version [1:7:0].		
2022-11-12 10:20:22.515 INFO 10592 [mainj o.a.catalina.core.AprilifecycleListener	· APR capabilities: IPV6 [iPue], sendice [iPue], accept filters [faise], random [crue],	
2022-11-12 10:20:22.514 INFO 10592 [2022-11-12 10:20:22.514 INFO 10592 [main] o a catalina.core.AprilifecycleListener	· OpenSSI successfully initialized [OpenSSI 1 16 3 May 2021]		
2022-11-12 10:20:22.321 INFO 10552 [main] o anache catalina core StandardService	Starting carvice [Tomost]		
2022-11-12 10:20:22.330 INFO 10552 [main] org anache catalina core StandardEngine	- Starting Service (nomcac)		
2022-11-12 10:20:22.520 INFO 10592 [main] o.a.c.c.C.[Tomcat].[localhost].[/]	: Initializing Spring embedded WebApplicationContext		
2022-11-12 10:20:22.520 INFO 10592 [main] w.s.c.ServletWebServerApplicationContext	: Root WebApplicationContext: initialization completed in 1567 ms		
2022-11-12 10:20:23.043 INFO 10592 [main] o.s.b.w.embedded.tomcat.TomcatWebServer	. Tomcat started on port(s): 8080 (http) with context path ''		
2022-11-12 10:20:23.059 INFO 10592 [main] c.m.MyFirstSpringBootProjectApplication	Started MyFirstSpringBootProjectApplication in 3.045 seconds (JVM running for 7.	307)	
				~
<			>	

图 2.13 项目的启动日志

打开浏览器,访问 http://127.0.0.1:8080/hello 地址,就可以在页面中看到代码返回的字符串,如 图 2.14 所示。



图 2.14 在浏览器中访问得到的结果

2.1.4 测试工具——Postman

Postman 是一款功能强大的网络接口测试工具,它可以模拟各种网络场景,发送各式各样的请求。 Spring Boot 是一个专门编写服务器接口的框架,一些特殊场景很难用前端页面模拟,因此推荐使用 Postman 来完成复杂场景的测试工作。本节将对 Postman 的下载、启动和使用进行详解。

下载 Postman 的步骤如下。

(1) 打开浏览器,输入 https://www.Postman.com/downloads/地址,网页会自动识别你的操作系统,给出适合操作系统的安装包。如图 2.15 所示,单击左侧的 Windows 64-bit 按钮。



图 2.15 Postman 下载页面

(2)下载完成之后,双击安装包,软件会自动安装。安装完成后会在桌面生成如图 2.16 所示的快 捷图标。



图 2.16 桌面上的 Postman 快捷图标

启动 Postman 的步骤如下。

32

(1) 双击 Postman 快捷图标打开 Postman 后,如果弹出用于提示用户登录或注册的对话框,就单击 skip and go to the app 超链接,跳过登录。

(2)对话框显示如图 2.17 所示的主界面后,单击 New 按钮,创建新的连接测试。

(3) 在弹出的如图 2.18 所示的对话框中,选择 HTTP Request 连接类型。



图 2.18 选择 HTTP Request 连接

API

Manage all aspects of API

00

(4)回到如图 2.19 所示的主界面后,程序开发人员需要先在功能面板中填写要访问的 URL 地址, 再设置请求类型、请求参数等,单击 Send 按钮即可向服务器发送一条请求。服务器返回的内容会在面 板底部展示。

使用 Postman 的方式如下:要使用 Postman 测试第一个 Spring Boot 程序中的连接,只需先在 URL 的位置填写 http://127.0.0.1:8080/hello,再单击右侧的 Send 按钮,即可在底部可看到服务器返回的结果,如图 2.20 所示。不难发现,图 2.20 所示的结果与图 2.14 所示的用户在浏览器上看到的内容相同。

Learn more on Postman Docs

Spring Boot 从入门到精通

≡ Hom	ne Workspaces v Explore		Q Search Postman	ලා හා sig	n In Create Accoun	t —	
公》Working locally in Scratch Pad. Switch to a Workspace					×		
Scratch Pad	New Import	Overview	GET www.mingrisoft.com • + •••		No Environmen	t	~ 🗄
Collections	+ = 000	www.mingrisoft.com			Save ·	/	=
0 00 APIs		GET ~ www.mit	ngrisoft.com			Send	~
Environments		Params Authorization H Query Params	leaders (6) Body Pre-request Script	Tests Settin	gs	Co	okies
	UU_	KEY	VALUE		DESCRIPTION	••• Bul	k Edit
Mock Servers	You don't have any collections	Key	Value		Description		
An Monitors	Collections let you group related requests, making them easier to access and run.						
45	Create Collection	Body Cookies (1) Headers	(13) Test Results	200 c	0K 435 ms 16.35 KB	Save Respons	se ∨
History		Pretty Raw Previe	w Visualize HTML ~ $\overline{-\varphi}$				Q
		1 html 2 <html> 3 4 <head> 5 <meta charse<br=""/>6 <title>明日料</title></head></html>	:="utf-8"> ¢				

图 2.19 发送请求的功能面板

Overvi	GET http://127.0.0.1:8080/hr •	+ 000		No Environment	~
http	://127.0.0.1:8080/hello			🖺 Save 🗸	
GET	 http://127.0.0.1:8080/hello 				Send ~
Parar	ns Authorization Headers (6) Body Pr	e-request Script Tests Settings			Cookies
Que	ry Params				
	KEY	VALUE	DESCRIPTION	1	••• Bulk Edit
	Кеу	Value	Description		
Body	Cookies Headers (5) Test Results	Status: 200 Status) OK Time: 12 n	ns Size: 211 B Sa	ve Response 🗸
Pre	tty Raw Preview Visualize Text	,			🔳 Q
1	L 你好,这是我的第一个Spring Boot项目				

图 2.20 使用 Postman 发送请求后得到的结果

2.1.5 打包项目

Spring Boot 可以将所有依赖都打包到一个 jar 包中,只需要执行这个 jar 包就可以启动完整 Spring

Boot项目。这为程序开发人员省去了不少配置和部署的工作。本节介绍如何在 Eclipse 环境中为 Spring Boot项目打包。步骤如下。

(1) 在项目上单击鼠标右键, 依次选择 Run As/Maven install 菜单, 位置如图 2.21 所示。选择之 后 Maven 会自动下载并打包所需的 jar 包。

20 M >)) >) >) =) =)	_	New Go Into	>		
		Open in New Window			
> =/		Open Type Hierarchy	F4		
) 		Show In	Alt+Shift+W >		
	D	Сору	Ctrl+C		
W	Þ	Copy Qualified Name			
	Ê	Paste	Ctrl+V		
6	ж	Delete	Delete		
M	S.	Remove from Context	Ctrl+Alt+Shift+Down		
		Build Path	>		
		Source	Alt+Shift+S >		
		Refactor	Alt+Shift+T >		
	è	Import			
	Ś	Export			
	R	Refresh	F5		
		Close Project			
		Assign Working Sets			
	0	Coverage As	>		
	\bigcirc	Run As	>	J	1 Java Application
	夺	Debug As	>	Ju	2 JUnit Test
		Profile As	>	m2	3 Maven build
		Restore from Local History		m2	4 Maven build
		Maven	>	m2	5 Maven clean
		Team	>	m2	6 Maven generate-sources
		Compare With	>	m2	7 Maven install
		Configure	>	m2	8 Mayen test

图 2.21 使用 Maven 的打包功能

(2) 打包时控制台会打印大量日志,当打包程序结束时,日志中出现如图 2.22 所示的 BUILD SUCCESS 字样,表示打包成功。

🖹 Markers 🔲 Properties 🚸 Servers	🛍 Data Source Explorer 🛛 🚡 Snipp	ets 🖉 Terminal 📮 Console	×	• X 💥 🗟 🖬 🖓 🖵 🗗 🖬 - 📬 🗸	8
<terminated> D:\Java\jdk-19\jdk-19.0.1\b</terminated>	n\javaw.exe (2022年11月12日 上午1	0:48:12) [pid: 4644]			
[INFO] Downloading from ali	yunmaven: https://maver	.aliyun.com/reposite	ry/public/com/google/code/findbugs/j	sr305/2.0.1/jsr305-2.0.1.jar	^
[INFO] Downloaded from aliy	unmaven: https://maven.	aliyun.com/repositor	y/public/commons-codec/commons-codec	/1.6/commons-codec-1.6.jar (233 kB at 9	5
[INFO] Downloading from ali	yunmaven: https://maver	.aliyun.com/reposite	ry/public/org/codehaus/plexus/plexus	-utils/3.0.15/plexus-utils-3.0.15.jar	
[INFO] Downloaded from aliy	unmaven: https://maven.	aliyun.com/repositor	y/public/classworlds/classworlds/1.1	-alpha-2/classworlds-1.1-alpha-2.jar (3	8
[INFO] Downloaded from aliy	unmaven: https://maven.	aliyun.com/repositor	y/public/junit/junit/3.8.1/junit-3.8	.1.jar (121 kB at 431 kB/s)	
[INFO] Downloaded from aliy	unmaven: https://maven.	aliyun.com/repositor	y/public/com/google/code/findbugs/js	r305/2.0.1/jsr305-2.0.1.jar (32 kB at 1	1
[INFO] Downloaded from aliy	unmaven: https://maven.	aliyun.com/repositor	y/public/org/apache/maven/shared/mav	en-shared-utils/0.4/maven-shared-utils-	0
[INFO] Downloaded from aliy	unmaven: https://maven.	aliyun.com/repositor	y/public/org/codehaus/plexus/plexus-	utils/3.0.15/plexus-utils-3.0.15.jar (2	3
[INFO] Installing D:\Spring	BootProject\MyFirstSpri	ngBootProject\target	\MyFirstSpringBootProject-0.0.1-SNAP	SHOT.jar to D:\Maven_3.8.6\apache-maven	-
[INFO] Installing D:\Spring	BootProject\MyFirstSpri	ngBootProject\pom.xr	l to D:\Maven_3.8.6\apache-maven-3.8	.6\Maven-lib\com\mr\MyFirstSpringBootPr	o
[INFO]					
[INFO] BUILD SUCCESS					
[INFO]					
[INFO] Total time: 45.874	s				
[INFO] Finished at: 2022-11	-12T10:49:01+08:00				
[INFO]					
					~
1					

图 2.22 打包成功日志

(3) 在项目上单击鼠标右键,在弹出的快捷菜单中选择 ReFresh (或按 F5 键)刷新项目,就可以 在 target 文件夹下看到很多文件,其中 MyFirstSpringBootProject-0.0.1-SNAPSHOT.jar 包就是本项目打

包生成的执行文件,位置如图 2.23 所示。



图 2.23 项目打包生成的 jar 包

(4) 将这个 jar 包保存到 D 盘根目录,再打开"命令提示符"对话框,输入以下命令:

d:

java -jar MyFirstSpringBootProject-0.0.1-SNAPSHOT.jar

命令执行结果如图 2.24 所示,可以看到 Spring Boot 项目成功启动,启动日志与 Eclipse 控制台中 打印的日志相同。此时就能打开浏览器访问项目资源了。



图 2.24 在"命令提示符"对话框中启动的效果

2.2 使用 Eclipse 创建 Spring Boot 项目

如果每次创建 Spring Boot 项目都要到官方网页下载,则不仅操作非常麻烦,而且学习成本太高。 那么,如何使用 Eclipse 创建 Spring Boot 项目呢?

为了能够在 Eclipse 中创建 Spring Boot 项目,需要为 Eclipse 安装 Spring 插件。Spring 插件的英文 全称是 Spring Tool Suite,简称 STS。

下面将分别介绍在 Eclipse 中安装 STS 插件和使用 STS 插件创建 Spring Boot 项目。

2.2.1 安装 STS 插件

在 Eclipse 自带的应用市场中安装 STS 插件的步骤如下。

(1) 依次选择 Help/Eclipse Marketplace 菜单,打开 Eclipse 自带的应用市场,如图 2.25 所示。

(2) 在应用市场中搜索 "sts", 在搜索结果中找到包含 "Sping Tool Suite" 字样的结果,并且确认 在这个结果中包含 "STS" 标签的插件,单击此插件的 Install 按钮,如图 2.26 所示。

图 2.25 打开 Eclipse 自带的应用市场

图 2.26 找到并安装 STS 插件

(3) 在确认安装内容的对话框中,选择所有内容后单击 Confirm 按钮,如图 2.27 所示。

🗑 Eclipse Marketplace			×
Confirm Selected Features			L.
Press Confirm to continue with the installation. Or go back to cho more solutions to install.	ose	Ľ	5
 Indie solutions to instali. Image: Solutions to instali. Image: Spring Tools 4 (aka Spring Tool Suite 4) 4.16.1.RELEASE Image: Image: Image: Spring Tool Suite 4 Main Feature (required) Image: Image: Im	https;	//downl	oad.sp
(?) < Install More Confirm > Einish		Cance	el

图 2.27 确认安装的内容

(4) 正式安装前需要同意该插件的许可声明,选择 I accept the terms of the license agreements 单选 按钮后,单击 Finish 按钮,如图 2.28 所示。

Eclipse Marketplace	– 🗆 X
Review Licenses Licenses must be reviewed and accepted before the software ca	in be installed.
Licenses:	License <u>t</u> ext:
 > Eclipse Public License - v 1.0 > SPRING IDE PROJECT SOFTWARE USER AGREEMENT > The content of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is provided under the EPL v1.0 licent of this package is package is provided under the EPL v1.0 licent of this package is package is provided under the EPL v1.0 licent of this package is package is package is package is package. 	Eclipse Public License - v 1.0 THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT. 1. DEFINITIONS "Contribution" means: a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and Image: Control of the license agreements O I go not accept the terms of the license agreement
? < <u>B</u> ack	<u>N</u> ext > <u>Finish</u> Cancel

图 2.28 同意许可声明

(5)同意许可之后,应用市场的对话框会自动关闭, Eclipse 右下方会显示插件安装的进度条,如 图 2.29 所示。因为 Eclipse 会从服务器上下载插件并安装,所以下载、安装的时间会很长。

Installing Software: (5%)	

图 2.29 开始自动安装

(6) 安装之后会弹出如图 2.30 所示的重启 Eclipse 对话框,单击 Restart Now 按钮即可立即重启。 重启之后就可以使用插件功能了。

Software Updates	Х
Restart Eclipse IDE to apply the software update?	
<u>R</u> estart Now <u>N</u> o	

图 2.30 重启提示

2.2.2 创建 Spring Boot 项目

安装完 STS 插件并重启 Eclipse 之后,就可以在 Eclipse 中直接创建 Spring Boot 项目了。使用 Eclipse 创建 Spring Boot 项目的步骤如下。

(1) 依次选择 File/New/Other 菜单,选择创建其他类型(Other)的项目。操作步骤如图 2.31 所示。

File	Edit Navigate Search Project	Run Window H	Help	
	New	Alt+Shift+N >	M	Maven Project
	Open File		Ê	Enterprise Application Project
Ē,	Open Projects from File System		63	Dynamic Web Project
	Recent Files	>	5	EJB Project
	Close Editor	Ctrl+W	Ê	Connector Project
	Close All Editors	Ctrl+Shift+W	₩¢ ₽	Application Client Project
n=n	Sava	C+-l + S	Î	Static Web Project
	Save As	Cuita	÷	JPA Project
	Save All	Ctrl+Shift+S	Ľ	Project
٩ā	Revert	carromero	ŝ	CSS File
	NOVEL 1		<u> </u>	JavaScript File
	Move		6	Servlet
	Rename	F2	Eŝ	Session Bean (EJB 3.x)
68	Refresh	F5	⊡\$	Message-Driven Bean (EJB 3.x)
	Convert Line Delimiters To	>	Â	Web Service
۵	Print	Ctrl+P	Ċ	Folder
2	Import		Ľ	File
4	Export		Ŷ	JSP File
	Properties	Alt+Enter	Ľ	Example
	Switch Workspace	>	Eì	Other Ctrl+N

图 2.31 选择创建其他类型的项目

(2)在所有项目类型中找到 Spring Boot 类型,展开之后选择 Spring Starter Project 选项,单击 Next 按钮,如图 2.32 所示。Spring Boot 类型是 STS 插件添加的新项目类型。

Select a wizard	_		×
Select a wizard Create new Spring Starter Project			\$
<u>W</u> izards:			
type filter text			
 > © Oomph > Plug-in Development > Server > Spring Boot > Import Spring Getting Started Content > SQL Development > Ouser Assistance > Web > Web Services 			~
? < Back Next > Einish		Cancel	

图 2.32 通过 Spring Starter 创建项目

(3) 进入创建项目的界面后, Eclipse 会先连接 Spring 官网, 再读取 Spring Boot 的版本以及创建项目所要填写的项,而后将网页中需要填写的内容展示在界面中。这一过程可能会因网速原因存在一定的延迟。当 Eclipse 从官网读取到所有信息后,会显示如图 2.33 所示的界面,程序开发人员可以在此界面填写之前在如图 2.1 所示的网页表单中填写过的内容。填写后单击 Next 按钮。

		×
	New Spring Starter Project	C
	Service URL https://start.spring.io Spring 官方地址 Name MySpringBootDemo 项目名称 ☑ Use default location Location Dtheclipse iee 2022-09\eclipse\eclipse-workspace\MySprin Bi	owse
创建 Maven 项目	Type: Maven Project V Packaging: Jar	项目的打包方式为 jar 包
使用 JDK 19	Java Version: 19 V Language: Java	使用 Java 语言
项目的唯一ID	Group com.mr于开及但队的唯一你识 Artifact MySpringBootDemo	
项目版本	Version 0.0.1-SNAPSHOT Description Demo project for Spring Boot - 项目描述	
指定底层包	Package com.mr	
	Working sets New Working sets: Vorking sets:	
	? < <u>B</u> ack <u>N</u> ext > Einish	Cancel

图 2.33 填写 Spring Boot 项目的相关内容

如果 Eclipse 无法连接到 Spring 官网,则会显示如图 2.34 所示的错误提示。遇到这种情况需要 先关闭创建项目的对话框,然后再重复创建步骤,直到 Eclipse 可以正常显示如图 2.33 所示的内容 为止。

🗕 SocketTimeo	utException: Connect timed out	
Service LIRI	https://start.spring.jo	
Service OKL	https://start.spring.io	`

图 2.34 无法连接 Spring 官网的情况

(4) 进入图 2.35 所示的界面中后,须选择 Spring Boot 的版本和依赖。版本使用默认的稳定版本 即可,添加 Web 依赖只需先在搜索框里输入 web,再选择 Spring Web 依赖。如果选错依赖,可以在右 侧的 Selected 分页中单击选错的依赖前面的×。添加完 Web 依赖之后,单击 Finish 按钮即可完成 Spring Boot 项目的创建。

(5) 如图 2.36 所示,使用 STS 插件创建的项目与 Spring 官网创建的项目并无差别。但是,安装 STS 插件之后, Spring Boot 项目中的 application.properties 配置文件将不再是文本图标,而是变成了 Spring 的树叶图标。

©.	— 🗆 X		
New Spring Starter Project Dependencies			
Spring Boot Version: 2.7.5	•		
✓ Messaging		Project Explorer X	4-
WebSocket		> B MyFirstSpringBootProject [boot]	
		V 🔂 MySpringBootDemo [boot]	
Thymeleaf		✓ (
Apache Freemarker		✓	
		> 🗾 MySpringBootDemoAppli	cation.jav
Testcontainers		✓	
= Web		🗁 static	
Cashing Web		🗁 templates	
Spring Reactive Web		application.properties	
Spring Web Services		> 🌁 src/test/java	
Jersey		> 🛋 JRE System Library	
Vaadin		> 🛋 Maven Dependencies	
		> 🗁 src	
	× 1	> 🗁 target	
		W HELP.md	
		mvnw	
		mvnw.cmd	
(f) < <u>B</u> ack <u>N</u> ext >	Einish Cancel	m pom.xml	

图 2.35 选择 Spring Boot 版本并添加 Web 依赖

图 2.36 通过 STS 插件在 Eclipse 中创建的项目

2.3 使用 IDEA 编写第一个 Spring Boot 程序

通过第1章的学习,已经完成了 IDEA 的下载、安装和配置。下面将介绍使用 IDEA 创建 Spring Boot 项目的步骤。

2.3.1 添加 Spring Initializr 选项

通过 IDEA 中的 Spring Initializr 选项,程序开发人员即可创建 Spring Boot 项目。只不过,在使用 IDEA 创建 Spring Boot 项目之前,需要先向 IDEA 添加 Spring Initializr 选项。向 IDEA 添加 Spring Initializr 选项的步骤如下。

(1) 双击 IntelliJ IDEA 的图标,打开 IntelliJ IDEA 后,将看到如图 2.37 所示的对话框。

图 2.37 打开 IntelliJ IDEA 后弹出的对话框

(2) 先单击图 2.37 中的 Plugins,再在搜索框内输入 Spring Boot,而后将看到如图 2.38 所示的用于显示 Spring Boot Helper 相关信息的对话框。

(3) 当单击 Spring Boot Helper 的 Install 按钮时,会弹出如图 2.39 所示的对话框。

(4) 单击 Accept 按钮后, IDEA 就会开始下载 Spring Boot Helper。待 Spring Boot Helper 下载完成后, Spring Boot Helper 的 Install 按钮会变为如图 2.40 所示的 Restart IDE 按钮。

第2章 第一个 Spring Boot 程序

图 2.38 显示 Spring Boot Helper 相关信息的对话框

图 2.39 Third-Party Plugins Privacy Note 对话框

(5) 单击图 2.40 中的 Restart IDE 按钮后, IDEA 会弹出如图 2.41 所示的对话框。单击对话框中的 Restart 按钮,重启 IDEA。

图 2.40 Spring Boot Helper 的 Install 按钮会变为 Restart IDE 按钮

图 2.41 单击 Restart 按钮

(6) 待 IDEA 重启后, 会弹出如图 2.42 所示的 Licenses 对话框。单击 Close 按钮,关闭 Licenses 对话框。

(7) Licenses 对话框被关闭后,单击如图 2.43 所示的 New Peoject 按钮,先创建一个 Java 项目。

44

(8) 如图 2.44 所示,把这个 Java 项目的名称(Name)设置为 JavaProject,把这个 Java 项目的存储路径(Location)设置为 D:\IDEA\IdeaProjects,单击 Create 按钮。

🖳 New Project		
Q		
New Project		JavaProject
Empty Project		D:\IDEA\IdeaProjects
Generators		
M Maven Archetype		
📭 JavaFX		Java Kotlin Groovy HTML +
Kotlin Multiplatform		
Compose Multiplatform	Build system:	IntelliJ Maven Gradle
🕛 IDE Plugin		■ 19 Oracle OpenJDK version 19.0.2
🛎 Android		
	🗹 Add sample	
	> Advanced Se	ttings
?		Cancel

图 2.44 创建一个 Java 项目

(9) 使用 IDEA 创建 Java 项目需要消耗一段时间,待 Java 项目创建完毕, IDEA 的工作区如图 2.45 所示。

图 2.45 IDEA 的工作区

说明 通过观察图 2.45 右下角是否有进度条,即可判断 Java 项目是否创建完毕。如果图的右下方没 有进度条,那么说明 Java 项目已创建完毕。 (10) 如图 2.46 所示,选择 File/Settings,打开 Settings 对话框。

Ŀ	<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>N</u> avigate	<u>C</u> ode <u>R</u> efactor <u>B</u> u	uild R <u>u</u> n <u>T</u> o	ols VC <u>S W</u> indo	ow <u>H</u> elp	JavaProject - Mai	in.java					×		>	<
Ja	va <u>N</u> ew							<u>n</u> - 1		Current File 🔻		G	Q	•	
📕 Project	 <u>Copen</u> <u>Recent Projects</u> Close Project Settings 	> ¢ .jec Ctrl+Alt+S	- C Mair	i java × no usages public class											Notificat
	Project Structure File Properties Local <u>H</u> istory	Ctrl+Alt+Shift+S > >				main(String[] args)		tem.	out.println(tions
	Save All Save All Reload All from Disk Repair IDE Invalidate Caches														
	Manage IDE Settings New Projects Setup Save File as Temp <u>l</u> ate														
Structure	Export <u>Print</u> Power Save Mode														
arks 🚛 :	E <u>x</u> it														
🔳 Bookm															
	♥ Version Control	🛛 Problems 🛛 T	Terminal 🛛 🖸												
	Edit application settings											UTF-8	4 spa	ces	1

图 2.46 打开 Settings 对话框

(11) 如图 2.47 所示, 先找到并单击 Plugins, 再在搜索输入框中输入 Spring Boot, 接着选择 Spring Boot Helper, 再接着单击 Apply 按钮, 而后单击 OK 按钮。

图 2.47 选择 Spring Boot Helper

46

(12) 这时 IDEA 会弹出如图 2.48 所示的对话框,其作用是提示用户是否重启 IDEA 以在插件中应用更改,单击 Restart 按钮。

图 2.48 重启 IDEA

(13) 待 IDEA 重启后, 会弹出如图 2.49 所示的 Licenses 对话框。单击 Close 按钮, 关闭 Licenses 对话框。

- (14) 如图 2.50 所示, 单击 Cancel 按钮, 关闭 Confirm Exit 对话框。
- (15) 如图 2.51 所示,单击 Close 按钮,关闭 Tip of the Day 对话框。

(16) 如图 2.52 所示, 在返回至 IDEA 的工作区后, 选择 File/New/Project, 打开 New Project 对话

框。

48

Ľ	E	ile <u>E</u> dit <u>V</u> iew <u>N</u> avigate	e <u>C</u> ode <u>R</u> efactor	r <u>B</u> uild R <u>u</u> n	<u>T</u> ools	VC <u>S W</u> indow <u>H</u> e	p JavaProject ·	Main.java					_		>	K
	Ja	New		Project					<u>n</u> -		Current File 🔻	ĕ G		Q	•	
t	<u>ء</u>	₽ <u>O</u> pen		Project fro	m Existin	g Sources										
- <u>oj</u> e		<u>R</u> ecent Projects		Project fro	m Versio	n Control										N
ĥ		Close Project		Module												otific
100	1	Settings	Ctrl+Alt+S	Module fr	om Existi	ng Sources										atio
		Project Structure	Ctrl+Alt+Shift+S	🗳 Scratch File	e Ct	rl+Alt+Shift+Insert	oid main(Stri	(anas)	{ SV	tem.	out println(ns
		File Properties			esigner			ingri argo,	,.		oo er prizine en ç					
		Local <u>H</u> istory														
	E	<u>S</u> ave All														
	ť	G Reload All from Disk	Ctrl+Alt+Y													
		Repair IDE														
		Invalidate Caches														
		Manage IDE Settings														
		New Projects Setup														
s		Export														
mark		Power Save Meda														
l age		Fower save would														
		EXIT														
tar																
Stru																
.:																
		Version Control 🛛 🖼 TODO	Problems	E Terminal	Service											
	Crea	ate a new project from scr	ratch										TF-8 4	4 spac		

图 2.52 打开 New Project 对话框

(17) 如图 2.53 所示,经过上述操作后,即可在 New Peoject 对话框中的 Generators 版块下找到 Spring Initializr 选项。

U New Project			×
	Name:	untitled	
Empty Project		D:\IDEA\IdeaProjects	
Maven Archetype		Create Git repository	
📲 JavaFX	Language:	Java Kotlin Groovy HTML +	
📕 Kotlin Multiplatform			
Compose Multiplatform	Build system:	IntelliJ Maven Gradle	
ि IDE Plugin	JDK:	■ 19 Oracle OpenJDK version 19.0.2 ▼	
🛎 Android			
🥏 Spring Initializr	🗹 Add sample	code	
	> Advanced Set		
		Create	

图 2.53 在 New Peoject 对话框中的 Generators 版块下找到 Spring Initializr 选项

2.3.2 使用 IDEA 创建 Spring Boot 项目

在成功地向 IDEA 添加 Spring Initializr 选项以后,即可使用 IDEA 创建 Spring Boot 项目。使用 IDEA 创建 Spring Boot 项目的步骤如下。

(1) 如图 2.54 所示,在单击 Spring Initializr 选项以后,将显示用于创建 Spring Boot 项目的相关 信息的界面。

🕑 New Project			×
	Project <u>S</u> DK:		
New Project			
Empty Project	Server URL:http	ps://start.spring.io 🌣	
	Project Metadat	ta	
M Maven Archetype	Project:	Gradle - Groovy 🔻	
Kotlin Multiplatform	Language:	Java 🔻	
Compose Multiplatform	<u>G</u> roup:	com.example	
IDE Plugin	Artifact:	demo	
🛎 Android			
🥏 Spring Initializr	<u>V</u> ersion:	0.0.1-SNAPSHOT	
	Na <u>m</u> e:	demo	
	<u>D</u> escription:	Demo project for Spring Boot	
	Pac <u>k</u> age name:	com.example.demo	
		17 •	
	Packaging:		
?		<u>N</u> ext Car	ncel

图 2.54 单击 Spring Initializr 选项

(2) 根据如图 2.55 所示的内容,修改用于创建 Spring Boot 项目的相关信息,单击 Next 按钮。

()注意	
在填写"项目的唯一 ID" (即 A	tifact)时,务必注意以下两点:
(1) 英文字母须小写。	
(2) 不得包含特殊字符。	
否则, IDEA 会弹出错误提示框。	

(3) 如图 2.56 所示,单击并打开 Web 下拉列表后,选择 Spring Web 选项,单击 Next 按钮。

🖳 New Project		\times
٩	Project <u>S</u> DK: ■ 19 Oracle OpenJDK version 19.0.2 使用 JDK 19	•
New Project		
Empty Project	Server URL:https://start.spring.io 🌣	
Generators	Project Metadata	
M Maven Archetype	Project: Maven — Maven 项目	
JavaFX	Language: Lava — 体田 Lava 语言	
Kotlin Multiplatform		
Compose Multiplatform	Group:	
	Artifact: sprbtdemo 项目的唯一 ID	
Spring Initializr	Version: 0.0.1-SNAPSHOT	
	News. Lagebtdown 顶日夕称	
	Name: sproudemo 次日石称	
	Description: Demo project for Spring Boot	
	Package name: com.mr.sprbtdemo 包名	
	Packaging: Jar 项目的打包方式为 jar 包	
?	<u>N</u> ext Cancel	

Spring Boot 从入门到精通

图 2.55 修改用于创建 Spring Boot 项目的相关信息

图 2.56 选择 Web 下拉列表中的 Spring Web 选项

(4) 如图 2.57 所示,在确认项目的名称和项目的存储路径后,单击 Create 按钮。

New Project	×
Project n <u>a</u> me:项目名称	
Project location: D:\IDEA\IdeaProjects\sprbtdemo项目	的存储路径
	▶
► More Settings	
	<u>P</u> revious <u>C</u> reate Cancel

图 2.57 确认项目的名称和项目的存储路径

(5) 如图 2.58 所示,单击 New Window 按钮,让名为 sprbtdemo 的 Spring Boot 项目在一个新的 窗口中显示。

图 2.58 让名为 sprbtdemo 的 Spring Boot 项目在一个新的窗口中显示

(6) 在名为 sprbtdemo 的 Spring Boot 项目马上要创建完毕时, IDEA 可能会出现闪退的情况。一旦 IDEA 出现闪退的情况,读者朋友只需重启 IDEA 即可。

(7)如图 2.59 所示,在重启 IDEA 后,把鼠标移至任务栏中的 IDEA 图标,会发现 IDEA 打开了两个窗口,左边的窗口用于显示 Java 项目,右边的窗口用于显示 Spring Boot 项目。

(8) 如图 2.60 所示,在打开用于显示 Spring Boot 项目的窗口后,会发现名为 sprbtdemo 的 Spring Boot 项目已经创建完毕。

Spring Boot 从入门到精通

📱 JavaProject – Main.java	📱 sprbtdemo
A such der den seiner und Berner Ahl der sein im Jahren im der eine im eine stehlten. An der der September im der Berner Ahl der seine im Anlahmen im An	En and e de maner une mener allé de serie en annes En allé de l'apprendie en anne et allé de serie en annes En allé de l'apprendie en anne et allé de serie en anne et allé de l'apprendie en anne et allé de serie et alle de serie et allé de serie et alle de serie et alle de serie et
Foreign Date from Bard Bart	Provide the second seco
]

图 2.59 IDEA 打开的两个窗口

图 2.60 Spring Boot 项目创建完毕并显示在窗口里

2.3.3 使用 IDEA 编写第一个 Spring Boot 程序

如图 2.61 所示,在 IDEA 中,先打开 sprbtdemo 文件夹,再依次打开 src 文件夹及其子文件夹,即 可看到 Spring Boot 项目 sprbtdemo 的项目结构。

对比图 2.61 和图 2.11,能够发现 IDEA 中 Spring Boot 项目的项目结构和 Eclipse 中 Spring Boot 项目的项目结构有些许不同。最为重要的不同在于 IDEA 的项目底层包是 com.mr.sprbtdemo, Eclipse 的项目底层包是 com.mr。明确这个不同点后,下面将在 IDEA 中实现 2.1.3 节(即"编写简单的跳转功能")

的内容,步骤如下。

图 2.61 Spring Boot 项目 sprbtdemo 的项目结构

(1) 如图 2.62 所示,在 com.mr.sprbtdemo 上单击鼠标右键,选择 New/Package。

图 2.62 新建 Package

(2)在弹出 New Package 对话框后,会发现 IDEA 已自动填写了"com.mr.sprbtdemo."。根据 2.1.3 节的要求,需要在 com.mr.sprbtdemo 包下创建子包 controller。如图 2.63 所示,因为 IDEA 已自动填写

了"com.mr.sprbtdemo.",所以读者只需要手动输入"controller"并按下回车键。

图 2.63 命名 Package

(3) 如图 2.64 所示,在已新建的包 controller 上单击鼠标右键,选择 New/Java Class。

图 2.64 新建 Java Class

(4) 如图 2.65 所示, 在弹出 New Java Class 对话框后, 输入新建 Java 类的类名 (即"HelloController") 并按下回车键。

New Java Class
C HelloController
C Class
Interface
Record
6 Enum
Annotation

图 2.65 命名 Java Class

(5) 如图 2.66 所示,在包 controller 下创建 Java 类 HelloController 后,就可以在 IDEA 的工作区 中编写 2.1.3 节中用于实现跳转功能的代码了。

(6) 如图 2.67 所示,在编码完毕后,会发现@RestController 和@RequestMapping 呈现红色,这 是 IDEA 的错误提示。

图 2.67 IDEA 出现错误提示

(7)如图 2.68 所示,把鼠标光标移至@RestController 处,IDEA 会弹出提示框。在提示框中找到 并单击 Import class 后,IDEA 就会自动向当前.java 文件导入与@RestController 相对应的包,即添加 "import org.springframework.web.bind.annotation.RestController;"。这时,@RestController 将呈现黄色, 说明已经消除了@RestController 的错误提示。再次通过上述的操作步骤,即可消除@RequestMapping 的错误提示。

Spring Boot 从入门到精通

Eile Edit View Navigate Code Refactor	<u>Build Run Tools VCS Window H</u> elp sprbtdemo - HelloController.java	-	o x
<code>sprbtdemo</code> $ angle$ src $ angle$ main $ angle$ java $ angle$ com $ angle$ mr $ angle$ spr	otdemo 🖯 controller) 🌀 HelloController 🥼 🕹 Current File		Q 🕦 🖻
번 🔲 Project 👻 🕀 프 🍝 🌩 —	G HelloController.java ×		: m
♀ ■ sprbtdemo D:\IDEA\IdeaProjects\sprbtd ▶ ■ .idea	1 package com.mr.sprbtdemo.controller; 2		2 ^ Y Maven
<pre>> ■ .mvn > ■ src > ■ main</pre>	no usages 3 @RestController 4 public c		∛• Notifi
 	5 0Reg Import class Alt+Shift+Enter More actions Alt+Enter		cations
HelloController SprbtdemoApplication HelloController	 o public String saynetlo() (ア return "你好,这是我的第一个Spring Boot项目"; 8 白 } 		-
static templates			-1
in application.properties in application.properties in diagram of the second			
P Version Control	🙋 Terminal 🔮 Services 📚 Dependencies	10:1 CRLF UTF-8 4	spaces 🎦

图 2.68 消除错误提示

(8) 在消除@RestController 和@RequestMapping 的错误提示后, IDEA 的工作区如图 2.69 所示。

	<u> </u>	or <u>B</u> uild R <u>u</u>	i <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp sprbtdemo - HelloController.java				
s	orbtdemo $ angle$ src $ angle$ main $ angle$ java $ angle$ com $ angle$ mr $ angle$ sp	rbtdemo > co	ntroller) 🜀 HelloController 🛛 🕹 🗸 🛛 Current File 💌 🗦		■ Q	. 🕤	۲
ţ	🔲 Project 👻 😯 호 🛨 🗢 —	G HelloCo	troller.java $ imes$				m
Proj	 Image: Sprbtdemo D:\IDEA\IdeaProjects\sprbt Image: Idea Image: Sprbt 		ckage com.mr.sprbtdemo.controller;		A 2		Maven
	✓ Linvn ✓ Lisrc ✓ Linva		port org.springframework.web.bind.annotation.RequestMapping; port org.springframework.web.bind.annotation.RestController;				Votifica
	✓ In com.mr.sprbtdemo ✓ In controller	6 (0 7 p	usages estController blic class HelloController ≮				ations
- 23	SprbtdemoApplication		no usages @RequestMapping("hello")				
	■ static ■ templates ╣₁application.properties		public String sayHello () { return "你好,这是我的第一个Spring Boot项目"; }				
cmarks	> 🖿 test 器.gitignore 器 HFI P.md						
 Structure I Book 	agarice into Da mvnw ∰ pom.xml ∰ sprbtdemo.iml > IIII External Libraries ™ Scratches and Consoles						
	Version Control III TODO O Problems	🗵 Terminal	Services Sependencies	CRLF UT	-8 4 sp		a

图 2.69 消除错误提示后的 IDEA 工作区

2.3.4 使用 IDEA 运行 Spring Boot 项目

编写完实现跳转功能的代码后,即可使用 IDEA 运行这个 Spring Boot 项目。使用 IDEA 运行这个

Spring Boot 项目的步骤如下。

(1) 如图 2.70 所示,双击 SprbtdemoApplication (即 Spring Boot 项目 sprbtdemo 的启动类)。

图 2.70 双击 SprbtdemoApplication

(2) 如图 2.71 所示,在 IDEA 显示 SprbtdemoApplication.java 文件的空白处,单击鼠标右键,单击 Run 'SprbtdemoAppli....main()',即可启动 Spring Boot 项目 sprbtdemo。

<u> <u> </u> </u>	r <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S W</u> indow <u>H</u> elp sprbtdemo - SprbtdemoAp		
<code>sprbtdemo</code> $ angle$ src $ angle$ main $ angle$ java $ angle$ com $ angle$ mr $ angle$ sp	btdemo 🕽 🕑 SprbtdemoApplication	🚨 🚽 🔨 🛛 Current File 💌 🕨 🗯	🗄 🕼 🔲 🔍 💿 🕑
빛	HelloController.java × SprbtdemoApplication.java ×		
Project Project Proj	<pre> HelloControllerjava ×</pre>	Image: Show Context Actions Alt+Enter Image: Paste Special Column Selection Mode Alt+Shift+Insert Image: Column Selection Mode Alt+Shift+Insert Image: Column Selection Mode Alt+Shift+Insert Image: Find Lisages Alt+Shift+Insert Image: Folding > Image: Folding ><	Lass, args); }
> IIII External Libraries Scratches and Consoles		Local <u>H</u> istory >	
		Create Gist	
↓ Version Control III TODO ● Problems	🗷 Terminal 🛛 Services 📚 Dependencies		1 LF UTF-8 Tab* 🎦

(3) 在第一次启动 Spring Boot 项目 sprbtdemo 时,可能会弹出如图 2.72 所示的 "Windows 安全 警报"对话框,确认已经选择"公用网络"后,单击"允许访问"按钮。

Windows 安全警	泯		×			
Windo	ws 防火墙已	经阻止此应用的部分功能				
Windows 防火墙已	阻止所有公用网络	各和专用网络上的 OpenJDK Platform binary 的某些功能。				
C V	名称(<u>N</u>):	OpenJDK Platform binary				
	发布者(P):	Oracle Corporation				
	路径(出):	D:\java\jdk-19\jdk-19.0.2\bin\java.exe				
允许 OpenJDK Plat □专用网络,例	form binary 在 如家庭或工作网	这些网络上通信: 络(<u>R</u>)				
公用网络,例如机场和咖啡店中的网络(不推荐,由于公用网络通常安全性很小或者根本不安全) ① ①						
<u>允许应用通过防火</u> 墙	有何风险?					
		●允许访问(A) 取消				

图 2.72 "Windows 安全警报"对话框

(4) 如图 2.73 所示,成功启动 Spring Boot 项目 sprbtdemo 后,IDEA 会陆续地在控制台上打印日志。需要注意的是,日志的第三行出现错误日志(即 ERROR)。错误日志的意思是:安装了不兼容的 Apache Tomcat 原生库版本[1.2.33],需要安装兼容的 Tomcat 版本[1.2.34]。

图 2.73 出现错误日志(即 ERROR)

(5)如图 2.74 所示,打开浏览器,访问 http://archive.apache.org/dist/tomcat/tomcat-connectors/native/ 地址(即 Apache 官网)。向下滚动鼠标滚轮,找到并单击"1.2.34/"超链接。

	234	5加速浏览器	10.29										〈菜	单	反馈	Ϋ́	_		×
登录	2 <	Cí	2 2	@ a	rchive.apa	che.org	/dist/to	r 4 -	* ~	两老人高铁	霸	Q	6 购	译•	Á		3 »	Ж.	$\underline{\downarrow}$
<1	🕒 Inde	ex of /dist/t	omcat/ton	nca X	+													÷	÷۲
	<u>1.2</u>	2.27/				2021-03	-29 12:4	7 -											-
	<u>1.2</u>	2.28/				2021-04	-06 10:5	4 -											
	<u>1.2</u>	2.3/				2015-12	-15 15:5	4 -											
	<u>1.2</u>	2.30/				2021-06	-04 10:4	7 -											
	<u>1.2</u>	2.31/				2021-09	-01 10:3	6 -											
	<u>1.2</u>	2.32/				2022-03	-22 09:0	2 -											- 11
	<u> </u>	2.33/				2022-05	-09 13:0	0 -											
M	<u>1.2</u>	2.34/				2022-06	-17 11:1	з –											
体	<u>1.2</u>	2.35/				2022-07	-12 19:5	0 -											
	<u>1.2</u>	2.36/				2023-02	-13 20:2	2 -											
汽	<u>1.2</u>	2.37/				2023-06	-02 11:5	4 -											
	<u>1.2</u>	2.4/				2016-01	-11 11:0	I3 -											
~	<u>1.2</u>	2.5/				2016-03	-07 20:5	1 -											
		2.6/				2016-04	-26 10:0	3 -											-
										@ 优化加	速	🖯 浏览	器医生	02	9 C))		Q 100	0% -

图 2.74 找到并单击"1.2.34/"超链接

(6) 如图 2.75 所示,在跳转至 1.2.34 版本的页面后,找到并单击 "binaries/" 超链接。

() () () () ()	2345加速浏览器 10.29 く C 合 合 @ archive.a	pache.org/dist/tor	, ☆ 、	/ 😨 整	条街的胶带	. Q .	〈 菜単 6 肉 译	反馈	ଟି ::::	 %•	$\times \rightarrow$
<1	B Index of /dist/tomcat/tomcat × +									÷	5.
	Index of /dist/to	omcat/to	mc	at-							
Ŀ	connectors/nativ	/e/1.2.34	ŀ								
	Name	Last modified	<u>Size</u> I	Description	2						
	Parent Directory		-								
	binaries/	2022-06-17 11:16	-								
体	<u>source/</u>	2022-06-17 11:16	-								
汽											
~											
					♤ 优化加速	🖯 浏览器	医生	per (]») [Q 100)% -

图 2.75 找到并单击 "binaries/" 超链接

(7) 如图 2.76 所示,在跳转至 binaries 目录的页面后,找到并单击 tomcat-native-1.2.34-openssl-1.1.1o-ocsp-win32-bin.zip 超链接。

	2345加速浏览器 10.29			<	単 反	溃 🕈	_		×
登录	C C A C @ archive.apache.org/dist/tor 4	☆ ∨ 📓 微信已期	巴帐号	.Q. Ma	译 •	Á	38 »	Ж	$\overline{}$
<	B Index of /dist/tomcat/tomc∈ × +							÷	÷C
	Index of /dist/tomcat/ton	ncat-							
C	connectors/native/1.2.34/	binaries	5						
ROE									
<u> </u>	Name	Last modified	<u>Size</u>	<u>Description</u>					
	Parent Directory		_						
M	tomcat-native-1.2.34-openssl-1.1.10-ocsp-win32-bin.zip	2022-06-07 09:58	3.5M						
体	tomcat-native-1.2.34-openss1-1.1.10-ocsp-win32-bin.zip.asc	2022-06-07 09:58	873						
	tomcat-native-1.2.34-openssl-1.1.10-ocsp-win32-bin.zip.sha51	<u>.2</u> 2022-06-07 09:58	184						
汽	tomcat-native-1.2.34-openssl-1.1.1o-win32-bin.zip	2022-06-07 09:58	3.5M						
	tomcat-native-1.2.34-openssl-1.1.1o-win32-bin.zip.asc	2022-06-07 09:58	873						
	📋 tomcat-native-1.2.34-openssl-1.1.10-win32-bin.zip.sha512	2022-06-07 09:58	179						
		企 优	化加速	🖯 浏览器医生	60	U»		Q 100)% -

图 2.76 找到并单击 tomcat-native-1.2.34-openssl-1.1.1o-ocsp-win32-bin.zip 超链接

(8)如图 2.77 所示,在弹出"新建下载任务"对话框后,先单击"浏览"按钮,选择 Apache Tomcat 版本[1.2.34]的存储位置,再单击"下载"按钮。

新建下载任	务	\times
网 址:	http://archive. apache.org /dist/tomcat/tomcat-connectors/native/1.2.]
文件名:	tomcat-native-1.2.34-openssl-1.1.10-ocsp-win32-bin.zip 3.47 MB]
下裁到:	D:\ 剩余227.18 GB ▼ 浏览]
迅雷下载	直接打开下载取消]

图 2.77 下载 Apache Tomcat 版本[1.2.34]

(9) 如图 2.78 所示,在下载之后,不解压,双击打开 tomcat-native-1.2.34-openssl-1.1.1o-ocsp-win32-bin.zip,双击打开 bin 文件夹。

📱 > 此电脑 > 软件 (D:) > tomcat-native-1.2.34-openssl-1.1.1o-ocsp-win32-bin.zip >							
名称 ^	类型	压缩大小	密码保护	大小	比率	修改日期	
bin	文件夹					2022/6/7 10:18	
LICENSE	文件	6 KB	否	18 K	3 68%	2022/6/6 17:29	
NOTICE	文件	1 KB	否	1 K	3 54%	2022/6/6 17:29	
README.txt	TXT 文件	2 KB	否	4 K	3 55%	2022/6/6 17:29	
VERSIONS	文件	1 KB	否	2 K	3 66%	2022/6/6 17:29	

图 2.78 直接打开 tomcat-native-1.2.34-openssl-1.1.1o-ocsp-win32-bin.zip

(10) 如图 2.79 所示,打开 bin 文件夹后,需要明确的是,x64 文件夹里的文件适用于 64 位的 Windows 系统,openssl.exe、tcnative-1.dll 和 tcnative-1-src.pdb 这 3 个文件适用于 32 位的 Windows 系统。

→ 此电脑 → 软件 (D:) → tomcat-native-1.2.34-openssl-1.1.1o-ocsp-win32-bin.zip → bin							
へ 名称	类型	压缩大小	密码保护	大小	比率	修改日期	
x64	文件夹					2022/6/7 10:18	
📧 openssl.exe	应用程序	1,199 KB	否	2,247 KB	47%	2022/5/3 21:12	
🚳 tcnative-1.dll	应用程序扩展	1,037 KB	否	1,942 KB	47%	2022/6/7 10:16	
🗿 tcnative-1-src.pdb	Program Debug Datab	46 KB	否	180 KB	75%	2022/6/7 10:16	

图 2.79 明确 bin 文件夹中子文件夹和各个文件的作用

(11) 如图 2.80 所示,因为笔者的操作系统是 64 位的 Windows 系统,所以双击打开 x64 文件夹。

→ 此电脑 → 软件 (D:) → tomcat-native-1.2.34-openssl-1.1.10-ocsp-win32-bin.zip → bin → x64							
~ 名称	类型	压缩大小	密码保护	大小	比率	修改日期	
🗟 tcnative-1.dll	应用程序扩展	1,222 KB	否	2,557 KB	53%	2022/6/7 10:17	
🗿 tcnative-1-src.pdb	Program Debug Datab	46 KB	否	180 KB	75%	2022/6/7 10:17	

图 2.80 打开 x64 文件夹

(12) 如图 2.81 所示,把 x64 文件夹中的 tenative-1.dll 和 tenative-1-src.pdb 这两个文件复制并粘 贴到 JDK 的 bin 目录下,即 D:/Java/jdk-19/jdk-19.0.2/bin 下。

🔜 > 此电脑 > 软件(D:) > Java > jdk-	19 > jdk-19.0.2 > bi	n	
	修改日期	类型	大小
🗟 management_ext.dll	2022/12/1 2:19	应用程序扩展	36 KB
🚳 mlib_image.dll	2022/12/1 2:19	应用程序扩展	499 KB
🚳 msvcp140.dll	2022/12/1 2:19	应用程序扩展	559 KB
🚳 net.dll	2022/12/1 2:19	应用程序扩展	67 KB
🚳 nio.dll	2022/12/1 2:19	应用程序扩展	79 KB
🚳 prefs.dll	2022/12/1 2:19	应用程序扩展	26 KB
🚳 rmi.dll	2022/12/1 2:19	应用程序扩展	21 KB
📧 rmiregistry.exe	2022/12/1 2:19	应用程序	24 KB
🚳 saproc.dll	2022/12/1 2:19	应用程序扩展	38 KB
📧 serialver.exe	2022/12/1 2:19	应用程序	24 KB
🚳 splashscreen.dll	2022/12/1 2:19	应用程序扩展	214 KB
🚳 sspi_bridge.dll	2022/12/1 2:19	应用程序扩展	44 KB
🚳 sunmscapi.dll	2022/12/1 2:19	应用程序扩展	48 KB
🗟 syslookup.dll	2022/12/1 2:19	应用程序扩展	28 KB
🗟 tcnative-1.dll	2022/6/7 10:17	应用程序扩展	2,557 KB
🔓 tcnative-1-src.pdb	2022/6/7 10:17	Program Debug	180 KB
🗟 ucrtbase.dll	2022/12/1 2:19	应用程序扩展	1,012 KB

图 2.81 把 x64 文件夹中的文件复制并粘贴到 JDK 的 bin 目录下

(13)关闭正在运行的 Spring Boot 项目 sprbtdemo,按照图 2.71 所示的操作步骤,再次启动 Spring Boot 项目 sprbtdemo。如图 2.82 所示, IDEA 会重新陆续地在控制台上打印日志。通过与图 2.73 进行 对比,会发现在图 2.73 中出现的错误日志(即 ERROR)已经消失了。

(14) 如图 2.83 所示,打开浏览器,访问 http://127.0.0.1:8080/hello 地址,就可以在页面中看到代码返回的字符串。

Spring Boot 从入门到精通

Eile Edit View Navigate <u>C</u> ode <u>R</u> efactor <u>B</u> uild R <u>un</u> <u>T</u> ools VC <u>S Wi</u> ndow <u>H</u> elp sprbtdemo-SprbtdemoApplicati	-	٥	×	
rbtdemo) src) main) java) com) mr) sprbtdemo) 🚳 SprbtdemoApplication 🔹 🖉 🧯		Q	•	Þ
= Project 👻 😌 王 夫 💠 ─ ⑧ HelloController.java × 🞯 SprbtdemoApplication.java ×				m
➤ mrsphtdemo DAUDEAN/deaProjects/sprbz 7 ▶ public class SprbtdemoApplication { > In idea 8 no usages no usages 9 ▶ a public static void main(String[] args) { SpringApplication.run(SprbtdemoApplication.class, args); }				Maven 📽 Nor
· → main 12				
Dia commespristemo 10 F Dia controller 14				
Run: @ScrbtdemoApplication ×		¢		
G ↑))) / / / / / / / / / / / / / / / / / / / / / / / / / / // / / / / / // / / / / / // / / / / / / / // / / / / / / / / // / / / / / / / / / / / / // / / / / / / / / / / / // / //	<u>:\IDEA</u> t" instal .7.0]. 10.18		<u>⊧Pro</u> whi	
presidencemon v kun Herbolo Ortobiens La terminal Orservices "Studio Sclependencies		F-8 Ta	ab* 1	

图 2.83 在浏览器中看到代码返回的字符串

说明 本书第 2~10 章的实例程序是不需要添加依赖的。读者朋友如果喜欢使用 IDEA 对第 2~10 章 的实例程序进行编码,可以参考本书附录的 A.1 节的内容。

62