

## 掌握Spring Boot全栈开发流程，独立实现大型SPA应用

- 讲述Spring、Spring MVC、MyBatis、Spring Boot和Vue全栈开发技术
- 所有的知识点都配有实例，让读者理解理论的同时也掌握开发技能
- 通过微人事项目实战，提高你的全栈开发水平



# Spring Boot+Vue 全栈开发实战

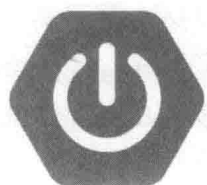
王松 著



本书示例源代码



清华大学出版社



# Spring Boot+Vue 全栈开发实战

王松 著

清华大学出版社  
北京

## 内 容 简 介

Spring Boot 致力于简化开发配置并为企业级开发提供一系列非业务性功能，而 Vue 则采用数据驱动视图的方式将程序员从烦琐的 DOM 操作中解救出来。利用 Spring Boot+Vue，我们可以快速开发出大型 SPA 应用。

本书分为 16 章，重点讲解 Spring Boot 2 + Vue 2 全栈开发所涉及的各种技术点。所有技术点都配有操作实例，循序渐进，直到引导读者开发出一个完整的微人事 SPA 应用。

本书适合有一定基础的 Java 开发者及 Spring Boot 初学者学习，也适合高等院校和培训学校相关专业的师生作为教学参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目 (CIP) 数据

Spring Boot+Vue 全栈开发实战/王松著. —北京：清华大学出版社，2019  
ISBN 978-7-302-51797-9

I. ①S… II. ①王… III. ①JAVA 语言—程序设计IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 269524 号

责任编辑：夏毓彦

封面设计：王 翔

责任校对：闫秀华

责任印制：丛怀宇

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>；<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：三河市龙大印装有限公司

经 销：全国新华书店

开 本：190mm×260mm 印 张：21.75

版 次：2019 年 1 月第 1 版

定 价：69.00 元

字 数：557千字  
印 次：2019 年 1 月第 1 次印刷



产品编号：081014-01

# 前 言

接触 Spring Boot 有好几年了，也曾断断续续出过一些教程，但是都比较零散，所使用的 Spring Boot 版本比较老，一直希望能够系统地写一本 Spring Boot 相关的图书，后来终于下定决心，在工作之余加班加点，于是有了读者现在所看到的这本书。

传统的 Spring 项目环境配置复杂臃肿，开发者早已不堪其苦，Spring Boot 带来的全新自动化配置解决方案一出现就受到了极大的关注，使得 Spring Boot 这两年成为 Java 领域的焦点之一。本书基于 Spring Boot 2.0.4（该版本是作者写作本书时 Spring Boot 的最新版本）完成。相对于 Spring Boot 1.5.X，Spring Boot 2 带来了许多新变化，这些在本书的相关章节都有体现。

本书分为 16 章，从以下方面向读者介绍 Spring Boot：

- 第 1 章 Spring Boot 入门
- 第 2 章 Spring Boot 基础配置
- 第 3 章 Spring Boot 整合视图层技术
- 第 4 章 Spring Boot 整合 Web 开发
- 第 5 章 Spring Boot 整合持久层技术
- 第 6 章 Spring Boot 整合 NoSQL
- 第 7 章 构建 RESTful 服务
- 第 8 章 开发者工具与单元测试
- 第 9 章 Spring Boot 缓存
- 第 10 章 Spring Boot 安全管理
- 第 11 章 Spring Boot 整合 WebSocket
- 第 12 章 消息服务
- 第 13 章 企业开发
- 第 14 章 应用监控
- 第 15 章 项目构建与部署
- 第 16 章 微人事项目实战

其中，第 1~15 章从视图层技术、持久化技术、NoSQL、RESTful、缓存、安全、WebSocket、消息服务以及企业开发等各个技术点对 Spring Boot 进行介绍；第 16 章通过一个 Spring Boot+Vue 搭建的前后端分离项目带领读者将前面 15 章所学的技术点应用到项目中，使读者深入体会前后端分离带来的好处，并学会搭建前后端分离的项目架构。

## 读者定位

本书适合有一定 Java Web 基础的开发者阅读，零基础的读者可以先学习 Java SE 和 Java Web 基础，再来阅读本书。

## 代码下载

本书示例源代码请扫描右边的二维码下载。如果下载有问题，请联系 [booksaga@163.com](mailto:booksaga@163.com)，邮件主题为“Spring Boot+Vue 全栈开发实战”。



## 技术支持

由于水平有限，疏漏之处在所难免，若读者发现疏漏之处，可以通过以下方式联系作者：

- 邮箱：[wangsong0210@gmail.com](mailto:wangsong0210@gmail.com)
- 微信：[ws584991843](https://www.wechat.com/p/584991843)

最后，祝每位读者阅读本书后都会有所收获，有所成长！

作者  
2018 年 10 月

# 目 录

第 1 章 Spring Boot 入门 .....	1
1.1 Spring Boot 简介 .....	1
1.2 开发第一个 Spring Boot 程序 .....	2
1.2.1 创建 Maven 工程 .....	2
1.2.2 项目构建 .....	5
1.2.3 项目启动 .....	7
1.3 Spring Boot 的简便创建方式 .....	8
1.3.1 在线创建 .....	8
1.3.2 使用 IntelliJ IDEA 创建 .....	9
1.3.3 使用 STS 创建 .....	11
1.4 小结 .....	12
第 2 章 Spring Boot 基础配置 .....	13
2.1 不使用 spring-boot-starter-parent .....	13
2.2 @SpringBootApplication .....	14
2.3 定制 banner .....	16
2.4 Web 容器配置 .....	17
2.4.1 Tomcat 配置 .....	17
2.4.2 Jetty 配置 .....	20
2.4.3 Undertow 配置 .....	21
2.5 Properties 配置 .....	21
2.6 类型安全配置属性 .....	22
2.7 YAML 配置 .....	24
2.7.1 常规配置 .....	24
2.7.2 复杂配置 .....	25
2.8 Profile .....	26
2.9 小结 .....	27
第 3 章 Spring Boot 整合视图层技术 .....	28
3.1 整合 Thymeleaf .....	28
3.2 整合 FreeMarker .....	31

3.3 小结.....	33
<b>第4章 Spring Boot 整合 Web 开发.....</b>	<b>34</b>
4.1 返回 JSON 数据.....	34
4.1.1 默认实现.....	34
4.1.2 自定义转换器.....	36
4.2 静态资源访问.....	40
4.2.1 默认策略.....	40
4.2.2 自定义策略.....	42
4.3 文件上传.....	42
4.3.1 单文件上传.....	43
4.3.2 多文件上传.....	45
4.4 @ControllerAdvice.....	46
4.4.1 全局异常处理.....	46
4.4.2 添加全局数据.....	48
4.4.3 请求参数预处理.....	48
4.5 自定义错误页.....	50
4.5.1 简单配置.....	52
4.5.2 复杂配置.....	55
4.6 CORS 支持.....	62
4.7 配置类与 XML 配置.....	67
4.8 注册拦截器.....	68
4.9 启动系统任务.....	70
4.9.1 CommandLineRunner.....	70
4.9.2 ApplicationRunner.....	72
4.10 整合 Servlet、Filter 和 Listener.....	73
4.11 路径映射.....	75
4.12 配置 AOP.....	75
4.12.1 AOP 简介.....	75
4.12.2 Spring Boot 支持.....	76
4.13 其他.....	78
4.13.1 自定义欢迎页.....	78
4.13.2 自定义 favicon.....	79
4.13.3 除去某个自动配置.....	79
4.14 小结.....	80
<b>第5章 Spring Boot 整合持久层技术.....</b>	<b>81</b>
5.1 整合 JdbcTemplate.....	81
5.2 整合 MyBatis.....	86

5.3	整合 Spring Data JPA.....	89
5.4	多数据源.....	95
5.4.1	JdbcTemplate 多数据源.....	96
5.4.2	MyBatis 多数据源.....	99
5.4.3	JPA 多数据源.....	102
5.5	小结.....	106
第 6 章	Spring Boot 整合 NoSQL.....	107
6.1	整合 Redis.....	108
6.1.1	Redis 简介.....	108
6.1.2	Redis 安装.....	108
6.1.3	整合 Spring Boot.....	110
6.1.4	Redis 集群整合 Spring Boot.....	113
6.2	整合 MongoDB.....	121
6.2.1	MongoDB 简介.....	121
6.2.2	MongoDB 安装.....	122
6.2.3	整合 Spring Boot.....	124
6.3	Session 共享.....	127
6.3.1	Session 共享配置.....	128
6.3.2	Nginx 负载均衡.....	129
6.3.3	请求分发.....	130
6.4	小结.....	131
第 7 章	构建 RESTful 服务.....	132
7.1	REST 简介.....	132
7.2	JPA 实现 REST.....	133
7.2.1	基本实现.....	133
7.2.2	自定义请求路径.....	138
7.2.3	自定义查询方法.....	138
7.2.4	隐藏方法.....	139
7.2.5	配置 CORS.....	140
7.2.6	其他配置.....	140
7.3	MongoDB 实现 REST.....	141
7.4	小结.....	142
第 8 章	开发者工具与单元测试.....	143
8.1	devtools 简介.....	143
8.2	devtools 实战.....	143
8.2.1	基本用法.....	143



8.2.2	基本原理 .....	145
8.2.3	自定义监控资源 .....	145
8.2.4	使用 LiveReload .....	146
8.2.5	禁用自动重启 .....	147
8.2.6	全局配置 .....	147
8.3	单元测试 .....	148
8.3.1	基本用法 .....	148
8.3.2	Service 测试 .....	149
8.3.3	Controller 测试 .....	149
8.3.4	JSON 测试 .....	152
8.4	小结 .....	153
<b>第 9 章</b>	<b>Spring Boot 缓存 .....</b>	<b>154</b>
9.1	Ehcache 2.x 缓存 .....	155
9.2	Redis 单机缓存 .....	159
9.3	Redis 集群缓存 .....	160
9.3.1	搭建 Redis 集群 .....	161
9.3.2	配置缓存 .....	161
9.3.3	使用缓存 .....	162
9.4	小结 .....	164
<b>第 10 章</b>	<b>Spring Boot 安全管理 .....</b>	<b>165</b>
10.1	Spring Security 的基本配置 .....	165
10.1.1	基本用法 .....	166
10.1.2	配置用户名和密码 .....	167
10.1.3	基于内存的认证 .....	167
10.1.4	HttpSecurity .....	168
10.1.5	登录表单详细配置 .....	170
10.1.6	注销登录配置 .....	172
10.1.7	多个 HttpSecurity .....	173
10.1.8	密码加密 .....	174
10.1.9	方法安全 .....	176
10.2	基于数据库的认证 .....	177
10.3	高级配置 .....	182
10.3.1	角色继承 .....	182
10.3.2	动态配置权限 .....	183
10.4	OAuth 2 .....	187
10.4.1	OAuth 2 简介 .....	187
10.4.2	OAuth 2 角色 .....	187

10.4.3 OAuth 2 授权流程 .....	188
10.4.4 授权模式 .....	188
10.4.5 实践 .....	189
10.5 Spring Boot 整合 Shiro.....	195
10.5.1 Shiro 简介.....	195
10.5.2 整合 Shiro.....	195
10.6 小结.....	200
<b>第 11 章 Spring Boot 整合 WebSocket.....</b>	<b>201</b>
11.1 为什么需要 WebSocket .....	201
11.2 WebSocket 简介 .....	202
11.3 Spring Boot 整合 WebSocket.....	203
11.3.1 消息群发 .....	204
11.3.2 消息点对点发送 .....	208
11.4 小结.....	213
<b>第 12 章 消息服务.....</b>	<b>214</b>
12.1 JMS .....	214
12.1.1 JMS 简介.....	214
12.1.2 Spring Boot 整合 JMS.....	215
12.2 AMQP.....	218
12.2.1 AMQP 简介.....	218
12.2.2 Spring Boot 整合 AMQP .....	218
12.3 小结.....	228
<b>第 13 章 企业开发.....</b>	<b>229</b>
13.1 邮件发送.....	229
13.1.1 发送前的准备 .....	229
13.1.2 发送 .....	231
13.2 定时任务.....	239
13.2.1 @Scheduled .....	239
13.2.2 Quartz .....	240
13.3 批处理.....	243
13.3.1 Spring Batch 简介 .....	243
13.3.2 整合 Spring Boot.....	243
13.4 Swagger 2.....	248
13.4.1 Swagger 2 简介 .....	248
13.4.2 整合 Spring Boot.....	248
13.5 数据校验.....	252

13.5.1	普通校验 .....	252
13.5.2	分组校验 .....	254
13.5.3	校验注解 .....	255
13.6	小结 .....	256
<b>第 14 章</b>	<b>应用监控 .....</b>	<b>257</b>
14.1	端点配置 .....	257
14.1.1	开启端点 .....	257
14.1.2	暴露端点 .....	259
14.1.3	端点保护 .....	261
14.1.4	端点响应缓存 .....	261
14.1.5	路径映射 .....	262
14.1.6	CORS 支持 .....	262
14.1.7	健康信息 .....	263
14.1.8	应用信息 .....	267
14.2	监控信息可视化 .....	272
14.3	邮件报警 .....	275
14.4	小结 .....	276
<b>第 15 章</b>	<b>项目构建与部署 .....</b>	<b>277</b>
15.1	JAR .....	277
15.1.1	项目打包 .....	277
15.1.2	项目运行 .....	279
15.1.3	创建可依赖的 JAR .....	280
15.1.4	文件排除 .....	281
15.2	WAR .....	283
15.3	小结 .....	283
<b>第 16 章</b>	<b>微人事项目实战 .....</b>	<b>284</b>
16.1	项目简介 .....	284
16.2	技术架构 .....	285
16.2.1	Vue 简介 .....	285
16.2.2	Element 简介 .....	285
16.2.3	其他 .....	286
16.3	项目构建 .....	286
16.3.1	前端项目构建 .....	286
16.3.2	后端项目构建 .....	287
16.3.3	数据模型设计 .....	287
16.4	登录模块 .....	293

---

16.4.1	后端接口实现 .....	293
16.4.2	前端实现 .....	302
16.5	动态加载用户菜单 .....	308
16.5.1	后端接口实现 .....	308
16.5.2	前端实现 .....	310
16.6	员工资料模块 .....	315
16.6.1	后端接口实现 .....	315
16.6.2	前端实现 .....	316
16.7	配置邮件发送 .....	319
16.8	员工资料导出 .....	322
16.8.1	后端接口实现 .....	322
16.8.2	前端实现 .....	325
16.9	员工资料导入 .....	325
16.9.1	后端接口实现 .....	325
16.9.2	前端实现 .....	329
16.10	在线聊天 .....	330
16.10.1	后端接口实现 .....	330
16.10.2	前端实现 .....	331
16.11	前端项目打包 .....	334
16.12	小结 .....	335

# 第 1 章

## Spring Boot 入门

### 本章概要

- Spring Boot 简介
- 开发第一个 Spring Boot 程序
- Spring Boot 的简便创建方式

### 1.1 Spring Boot 简介

Spring 作为一个轻量级的容器，在 Java EE 开发中得到了广泛的应用，但是 Spring 的配置烦琐臃肿，在和各种第三方框架进行整合时代码量都非常大，并且整合的代码大多是重复的，为了使开发者能够快速上手 Spring，利用 Spring 框架快速搭建 Java EE 项目，Spring Boot 应运而生。

Spring Boot 带来了全新的自动化配置解决方案，使用 Spring Boot 可以快速创建基于 Spring 生产级的独立应用程序。Spring Boot 中对一些常用的第三方库提供了默认的自动化配置方案，使得开发者只需要很少的 Spring 配置就能运行一个完整的 Java EE 应用。Spring Boot 项目可以采用传统的方案打成 war 包，然后部署到 Tomcat 中运行。也可以直接打成可执行 jar 包，这样通过 `java -jar` 命令就可以启动一个 Spring Boot 项目。总体来说，Spring Boot 主要有如下优势：

- 提供一个快速的 Spring 项目搭建渠道。
- 开箱即用，很少的 Spring 配置就能运行一个 Java EE 项目。
- 提供了生产级的服务监控方案。

- 内嵌服务器，可以快速部署。
- 提供了一系列非功能性的通用配置。
- 纯 Java 配置，没有代码生成，也不需要 XML 配置。

Spring Boot 是一个“年轻”的项目，发展非常迅速，特别是在 Spring Boot 2.0 之后，许多 API 都有较大的变化，本书的写作基于目前最新的稳定版 2.0.4（本书写作时的最新版），因此需要 Java 8 或 9 以及 Spring Framework 5.0.8.RELEASE 或更高版本，同时，构建工具的版本要求为 Maven 3.2+ 或 Gradle 4。

## 1.2 开发第一个 Spring Boot 程序

Spring Boot 工程可以通过很多方式来创建，最通用的方式莫过于使用 Maven 了，因为大多数的 IDE 都支持 Maven。

### 1.2.1 创建 Maven 工程

这里不过多说明，Maven 的介绍和安装只介绍三种创建 Maven 工程的方式。

#### 1. 使用命令创建 Maven 工程

首先可以通过 Maven 命令创建一个 Maven 工程，在 cmd 窗口中执行如下命令：

```
1 mvn archetype:generate -DgroupId=org.sang -DartifactId=chapter01
2 -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

命令解释：

- **-DgroupId** 组织 Id（项目包名）。
- **-DartifactId** ArtifactId（项目名称或者模块名称）。
- **-DarchetypeArtifactId** 项目骨架。
- **-DinteractiveMode** 是否使用交互模式。

使用命令将项目创建好之后，直接用 Eclipse 或者 IntelliJ IDEA 打开即可。

#### 2. 在 Eclipse 中创建 Maven 工程

大部分的 IDE 工具都可以直接创建 Maven 工程。在 Eclipse 中创建 Maven 工程的步骤如下：

**步骤01** 创建项目时选择 Maven Project，如图 1-1 所示。

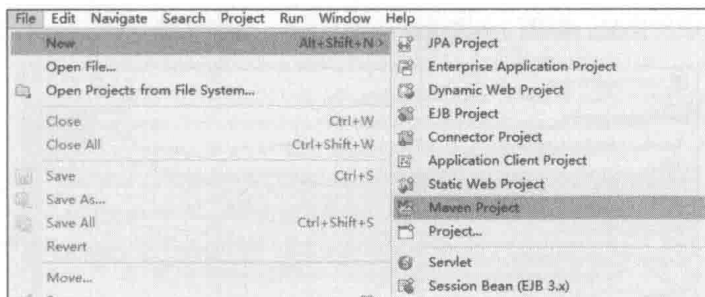


图 1-1

**步骤02** 选中 Use default Workspace location 复选框，如图 1-2 所示。

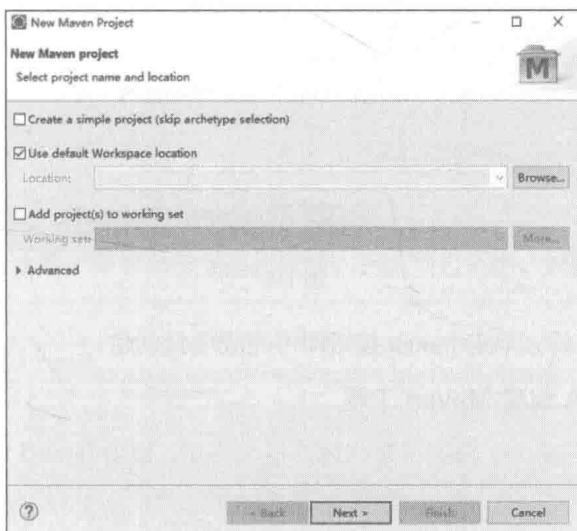


图 1-2

**步骤03** 选择项目骨架，保持默认设置即可，如图 1-3 所示。

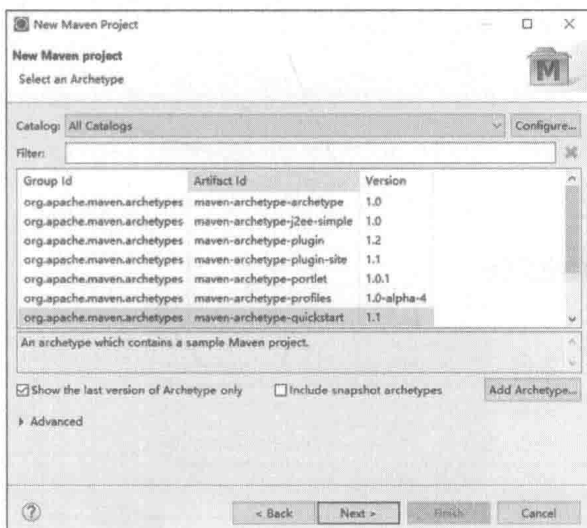


图 1-3

**步骤04** 输入项目信息，如图 1-4 所示。

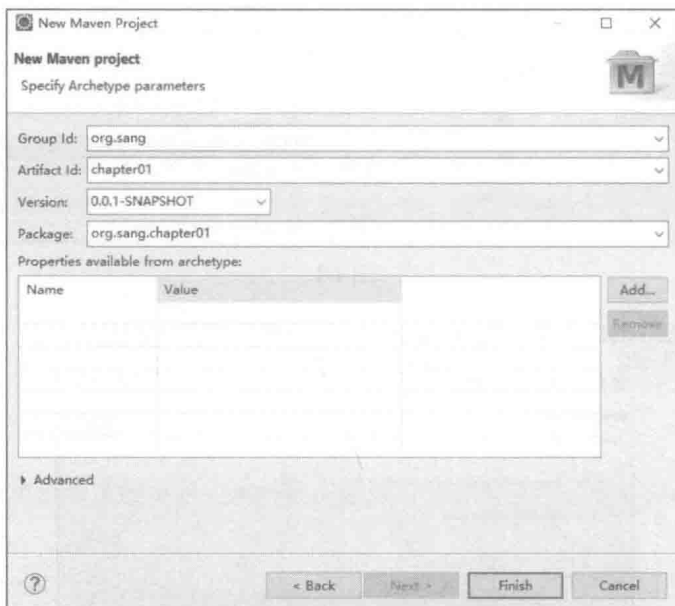


图 1-4

完成以上 4 个步骤之后，单击 **Finish** 按钮即可完成项目创建。

### 3. 使用 IntelliJ IDEA 创建 Maven 工程

IntelliJ IDEA 作为后起之秀，得到了越来越广泛的应用。使用 IntelliJ IDEA 创建 Maven 工程的步骤如下：

**步骤01** 创建项目时选择 **Maven**，但是不必选择项目骨架，直接单击 **Next** 按钮即可，如图 1-5 所示。

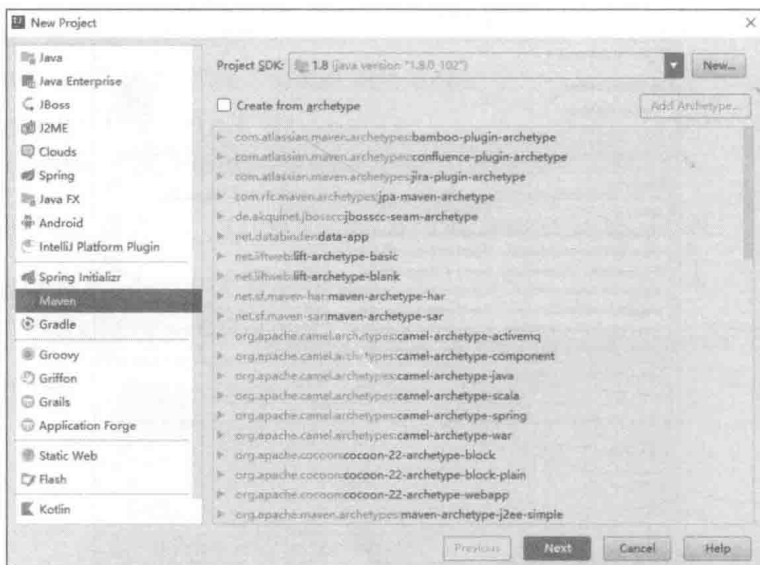


图 1-5



**步骤02** 输入组织名称、模块名称、项目版本号等信息，如图 1-6 所示。

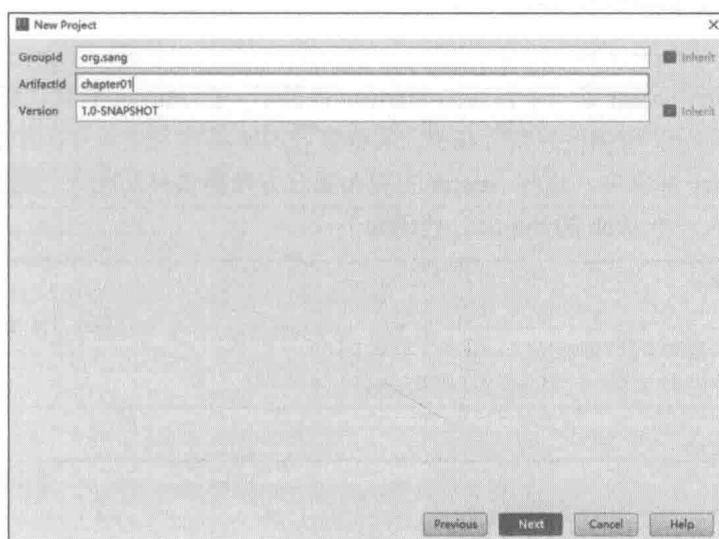


图 1-6

**步骤03** 选择项目位置，然后单击 **Finish** 按钮，完成项目创建，如图 1-7 所示。



图 1-7

这里一共向读者介绍了三种创建 Maven 工程的方式，创建成功之后，接下来添加项目依赖。

## 1.2.2 项目构建

### 1. 添加依赖

首先添加 `spring-boot-starter-parent` 作为 `parent`，代码如下：

```
1 <parent>  
2 <groupId>org.springframework.boot</groupId>
```