

脱离“Hello World”阶段
向Spring Boot企业级快速应用开发跃迁

Spring Boot

企业级应用

开发实战

柳伟卫
◎著

非
外
借



北京大学出版社
PEKING UNIVERSITY PRESS

Spring Boot

企业级应用

开发实战

柳伟卫◎著



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 提 要

本书围绕如何整合以目前最新的 Spring Boot 2 版本为核心的技术栈,来实现一个完整的企业级博客系统而展开。读者可以通过学习构建这个博客系统的整个过程,来达到设计和实现一个企业级 Java EE 应用开发的目的。该博客系统是一个类似于 WordPress 的专注于博客功能的博客平台,支持多用户访问和使用。该博客系统所涉及的相关技术有 Spring Boot、Spring、Spring MVC、Spring Security、Spring Data、Hibernate、Gradle、Bootstrap、jQuery、HTML5、JavaScript、CSS、Thymeleaf、MySQL、H2、Elasticsearch、MongoDB 等,技术点较为丰富,内容富有前瞻性。

本书面向实战,除了给出基本的原理外,会辅以大量的案例和源码,利于读者理论联系实践。全书对于技术讲解的安排,是按照渐进式的教学方式来进行的。按照学习的难度,大致可以分为实战入门阶段、实战进阶阶段、实战高级阶段三个部分,内容包括 Spring Boot 概述、Spring 框架核心概念、Spring MVC 及常用 MediaType、集成 Thymeleaf、数据持久化、全文搜索、架构设计与分层、集成 Bootstrap、博客系统的需求分析与设计、集成 Spring Security、博客系统的整体框架实现、用户管理实现、角色管理实现、权限管理实现、文件服务器实现、博客管理实现、评论管理实现、点赞管理实现、分类管理实现、标签管理实现、首页搜索实现等。

本书主要面向的用户是 Java 开发者,以及对 Spring Boot 及企业级开发感兴趣并有一定了解的读者。

图书在版编目(CIP)数据

Spring Boot 企业级应用开发实战 / 柳伟卫著 — 北京:北京大学出版社,2018.3
ISBN 978-7-301-29230-3

I. ①S… II. ①柳… 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028100号

- 书 名** Spring Boot 企业级应用开发实战
SPRING BOOT QIYE JI YINGYONG KAIFA SHIZHAN
- 著作责任者** 柳伟卫 著
- 责任编辑** 尹 毅
- 标准书号** ISBN 978-7-301-29230-3
- 出版发行** 北京大学出版社
- 地 址** 北京市海淀区成府路205号 100871
- 网 址** <http://www.pup.cn> 新浪微博: @北京大学出版社
- 电子信箱** pup7@pup.cn
- 电 话** 邮购部 62752015 发行部 62750672 编辑部 62570390
- 印 刷 者** 北京大学印刷厂
- 经 销 者** 新华书店
- 787毫米×1092毫米 16开本 28.25印张 656千字
2018年3月第1版 2018年3月第1次印刷
- 印 数** 1-3000册
- 定 价** 98.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话:010-62752024 电子信箱:fd@pup.pku.edu.cn

图书如有印装质量问题,请与出版部联系。电话:010-62756370

谨将此书献给我的小女儿伊然，愿她永远天真烂漫！

前言

Preface

写作背景

对于 Spring Boot 知识的整理归纳，最早是在笔者的第一本书《分布式系统常用技术及案例分析》的微服务章节中，作为微服务的技术实现方式来展开的。由于篇幅限制，当时讲解的案例深度和广度也比较有限。其后，笔者又在 GitHub 上，以开源方式撰写了《Spring Boot 教程》系列课程^①，为网友们提供了更加丰富的使用案例。在 2017 年年初，笔者应邀给慕课网做了一个关于 Spring Boot 实战的系列视频课程^②。视频课程上线后受到了广大的 Spring Boot 技术爱好者的关注，课程的内容也引发了热烈的反响。很多该课程的学员，通过学习该课程，不但技术能力提高了，而且在如何采用新技术来实现企业级应用上有了更深刻的理解，最重要的是提升了自己在市场上的价值。

鉴于 Spring Boot 技术人才在社会上的需求依然很旺盛，而市面上有关 Spring Boot 学习资料，大多停留在“Hello World”级别的案例，缺乏使用 Spring Boot 来构建完整企业级应用实战的能力。故笔者将以往系列课程中的技术做了总结和归纳，采用目前最新的 Spring Boot 2 技术来重新编写了整个教学案例，整理成书，希望能够弥补 Spring Boot 在实战方面的空白，使广大 Spring Boot 爱好者都能受益。

源代码

本书提供源代码下载，下载地址为 <https://github.com/waylau/spring-boot-enterprise-application-development>。

本书所涉及的技术及相关版本

技术版本是非常重要的，特别是对于实战内容而言。因为不同的版本之间，是存在兼容性问题的，而且不同的版本，软件所对应的功能也是不同的。本书所列出的技术，版本上相对比较新，都是经过笔者自己大量实际测试的。这样，读者在自行搭建博客系统时，可以参考本书所列出的版本，从而可以避免很多因为版本兼容性所产生的问题。建议读者将相关开发环境设置成与本书所采用的一致，或者不低于本书所列的配置。详细的版本配置，可以参阅本书的“附录 E”内容。

本书示例采用 Eclipse 编写，但示例源码与具体的 IDE 无关，读者可以自行选择适合自己的 IDE，如 IntelliJ IDEA、NetBeans 等。

^① 有关该教程介绍，可参见 <https://github.com/waylau/spring-boot-tutorial>。

^② 有关该课程介绍，可参见 <http://coding.imooc.com/class/125.html>。

勘误和交流

本书如有勘误，会在 <https://github.com/waylau/spring-boot-enterprise-application-development> 上进行发布。由于笔者能力有限，时间仓促，错漏之处在所难免，欢迎读者批评指正。

读者也可以直接通过以下方式联系。

博客：<https://waylau.com>

邮箱：waylau521@gmail.com

微博：<http://weibo.com/waylau521>

开源：<https://github.com/waylau>

致谢

感谢北京大学出版社工作人员为本书在出版过程中所做出的努力。

感谢我的父母、妻子和两个女儿。由于撰写本书，牺牲了很多陪伴家人的时间。感谢家人对我工作的理解和支持。

最后，感谢 Spring Boot 团队为 Java 社区提供了这么优秀的框架。由衷地希望 Spring Boot 框架发展得越来越好！

柳伟卫

目录

Contents

第 1 章 Spring Boot 概述	1
1.1 传统企业级应用开发之痛与革新	2
1.2 Spring Boot 2 总览	11
1.3 快速开启第一个 Spring Boot 项目	16
1.4 如何进行 Spring Boot 项目的开发及测试	24
第 2 章 Spring 框架核心概念	32
2.1 Spring 框架总览	33
2.2 依赖注入与控制反转	37
2.3 AOP 编程	46
第 3 章 Spring MVC 及常用 MediaType	52
3.1 Spring MVC 简介	53
3.2 JSON 类型的处理	56
3.3 XML 类型的处理	61
3.4 文件上传的处理	63
第 4 章 集成 Thymeleaf	67
4.1 常用 Java 模板引擎	68
4.2 Thymeleaf 标准方言	74
4.3 Thymeleaf 设置属性值	84
4.4 Thymeleaf 迭代器与条件语句	88
4.5 Thymeleaf 模板片段	92
4.6 Thymeleaf 表达式基本对象	95
4.7 Thymeleaf 与 Spring Boot 集成	97
4.8 Thymeleaf 实战	98

第 5 章 数据持久化	107
5.1 JPA 概述	108
5.2 Spring Data JPA	127
5.3 Spring Data JPA 与 Hibernate、Spring Boot 集成	143
5.4 数据持久化实战	147
第 6 章 全文搜索	156
6.1 全文搜索概述	157
6.2 Elasticsearch 核心概念	159
6.3 Elasticsearch 与 Spring Boot 集成	164
6.4 Elasticsearch 实战	168
第 7 章 架构设计与分层	175
7.1 为什么需要分层	176
7.2 系统的架构设计及职责划分	179
第 8 章 集成 Bootstrap	182
8.1 Bootstrap 简介	183
8.2 Bootstrap 核心概念	185
8.3 Bootstrap 及常用前端框架与 Spring Boot 集成	189
8.4 Bootstrap 实战	192
第 9 章 博客系统的需求分析与设计	196
9.1 博客系统的需求分析	197
9.2 博客系统的原型设计	201
第 10 章 集成 Spring Security	206
10.1 基于角色的权限管理	207
10.2 Spring Security 概述	210
10.3 Spring Security 与 Spring Boot 集成	218
10.4 Spring Security 实战	219
第 11 章 博客系统的整体框架实现	227
11.1 如何设计 API	228
11.2 实现后台整体控制层	233
11.3 实现前台整体布局	237

第 12 章 用户管理实现	242
12.1 用户管理的需求回顾	243
12.2 用户管理的后台实现	245
12.3 用户管理的前台实现	254
第 13 章 角色管理实现	264
13.1 角色管理的需求回顾	265
13.2 角色管理的后台实现	266
13.3 角色管理的前台实现	272
第 14 章 权限管理实现	276
14.1 权限管理的需求回顾	277
14.2 权限管理的后台实现	278
14.3 CSRF 防护处理	281
14.4 权限管理的前台实现	282
第 15 章 文件服务器实现	285
15.1 文件服务器的需求分析	286
15.2 MongoDB 简介	286
15.3 MongoDB 与 Spring Boot 集成	291
15.4 文件服务器的实现	293
第 16 章 博客管理实现	303
16.1 博客管理的需求回顾	304
16.2 实现个人设置和头像变更	306
16.3 博客管理的后台实现	316
16.4 博客管理的前台实现	325
第 17 章 评论管理实现	337
17.1 评论管理的需求回顾	338
17.2 评论管理的后台实现	338
17.3 评论管理的前台实现	346
第 18 章 点赞管理实现	350
18.1 点赞管理的需求回顾	351
18.2 点赞管理的后台实现	351
18.3 点赞管理的前台实现	358

第 19 章 分类管理实现	362
19.1 分类管理的需求回顾	363
19.2 分类管理的后台实现	364
19.3 分类管理的前台实现	372
第 20 章 标签管理实现	380
20.1 标签管理的需求回顾	381
20.2 标签管理的后台实现	381
20.3 标签管理的前台实现	383
第 21 章 首页搜索实现	385
21.1 首页搜索的需求回顾	386
21.2 首页搜索的后台实现	387
21.3 首页搜索的前台实现	399
21.4 使用中文分词	405
第 22 章 总结与展望	408
22.1 Spring Boot 企业级应用开发的总结	409
22.2 博客系统的展望	412
附录 A 开发环境的搭建	414
附录 B Thymeleaf 属性	418
附录 C Thymeleaf 表达式工具对象	420
附录 D Bean Validation 内置约束	438
附录 E 本书所涉及的技术及相关版本	440
参考文献	442



第1章

Spring Boot 概述

1.1 传统企业级应用开发之痛与革新

作为一门“长寿”的编程语言，Java 语言经历了 20 多年的发展，已成为开发者首选的利器。在最新的 TIOBE 编程语言排行榜中，Java 位居榜首。回顾历史，Java 语言的排行也一直是名列三甲。图 1-1 展示的是 2017 年 8 月 TIOBE 编程语言排行榜情况。

Aug 2017	Aug 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.961%	-6.05%
2	2		C	6.477%	-4.83%
3	3		C++	5.550%	-0.25%
4	4		C#	4.195%	-0.71%
5	5		Python	3.692%	-0.71%
6	8	^	Visual Basic .NET	2.569%	+0.05%
7	6	v	PHP	2.293%	-0.88%
8	7	v	JavaScript	2.098%	-0.61%
9	9		Perl	1.995%	-0.52%
10	12	^	Ruby	1.965%	-0.31%
11	14	^	Swift	1.825%	-0.16%
12	11	v	Delphi/Object Pascal	1.825%	-0.45%
13	13		Visual Basic	1.809%	-0.24%
14	10	v	Assembly language	1.805%	-0.56%
15	17	^	R	1.766%	+0.16%
16	20	^	Go	1.645%	+0.37%
17	18	^	MATLAB	1.619%	+0.08%
18	15	v	Objective-C	1.505%	-0.38%
19	22	^	Scratch	1.481%	+0.43%
20	26	^	Dart	1.273%	+0.30%

图1-1 TIOBE 编程语言排行榜

然而，作为当今企业级应用的首选编程语言，Java 的发展也并非一帆风顺。

1.1.1 Java 大事件

最初，SUN 公司准备用一种新的语言来设计用于智能家电类（如机顶盒）的程序开发。“Java 之父”James Gosling 创造出了这种全新的语言，并被命名为“Oak”（橡树），以他的办公室外的树来命名。然而，由于当时机顶盒的项目并没有成功拿下，于是 Oak 被阴差阳错地应用到了万维网。作为原型，SUN 公司的工程师编写了一个小型万维网浏览器 WebRunner（后来改名为 Hot-

Java)，该浏览器可以直接用来运行 Java 小程序（即 Java Applet）。同年，Oak 改名为 Java。由于 Java Applet 程序可以产生一般网页所不能实现的效果，从而引起业界对 Java 的热捧，于是当时很多操作系统都预装了 Java 虚拟机。

1997年4月2日，JavaOne 会议召开，参与者逾1万人，创下当时全球同类会议规模最大之纪录。

1998年12月8日，Java 2 企业平台 J2EE 发布，正式进军企业级应用开发领域。

1999年6月，随着 Java 的快速发展，SUN 公司将 Java 分为3个版本，即标准版（J2SE）、企业版（J2EE）和微型版（J2ME）。从版本的划分可以看出当时 Java 语言的野心，企图统治桌面应用、服务器端应用及移动端应用。

2004年9月30日 J2SE 1.5 发布，成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性，J2SE 1.5 更名为 Java SE 5.0。

2005年6月，JavaOne 大会召开，SUN 公司发布了 Java SE 6。此时，Java 的各种版本已经更名，以取消其中的数字“2”，J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

2009年4月20日，Oracle 公司以74亿美元收购了 SUN 公司，从此 Java 归属于 Oracle 公司。

2011年7月28日，Oracle 公司发布 Java 7 的正式版。该版本新增了诸如 try-with-resources 语句、增强 switch-case 语句支持字符串类型等特性。

2011年6月中旬，Oracle 公司正式发布了 Java EE 7。该版本的目标在于提高开发人员的生产力，满足苛刻的企业需求。

2014年3月19日，Oracle 公司发布 Java 8 的正式版。该版本中的 Lambdas 表达式、Streams 流式计算框架等广受开发者关注。

由于 Java 9 中计划开发的模板化项目（或称为 Jigsaw）存在较大的技术难度，JCP 执行委员会内部成员也无法达成共识，所以造成了该版本的发布一再延迟。Java 9 及 Java EE 8 终于在 2017 年 9 月份发布，并将 Java EE 8 移交给了开源组织 Eclipse 基金会。

图 1-2 所示为 Java EE 8 整体架构图。

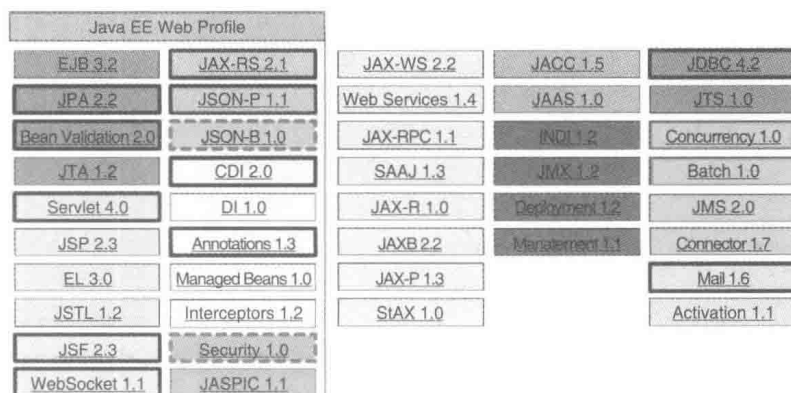


图1-2 Java EE 8整体架构图

1.1.2 Java 企业级应用现状

作为 Java 企业级应用开发的规范——Java EE，从诞生之初就饱受争议。特别是 EJB（Enterprise Java Beans）作为 Java 企业级应用开发的核心，由于其设计的复杂性，使之在 J2EE 架构中的表现一直不是很好。EJB 大概是 J2EE 架构中唯一没有兑现其能够简化开发并提高生产力承诺的组件。

正当 Java 开发者无法忍受 EJB 的臃肿不堪时，Spring 应运而生。Spring 框架打破了传统 EJB 开发模式中以 Bean 为重心的强耦合、强侵入性的弊端，采用依赖注入和 AOP（Aspect Oriented Programming，面向切面编程）等技术，来解耦对象间的依赖关系，无须继承复杂 Bean，只需要 POJOs（Plain Old Java Objects，简单的 Java 对象），就能快速实现企业级应用的开发。为此，“Spring 之父” Rod Johnson 还特意撰写了《Expert one-on-one J2EE Development without EJB》一书，来向 EJB 宣战，从而业界掀起了以 Spring 为核心的轻量级应用开发的狂潮。

Spring 框架最初的 Bean 管理是通过 XML 文件来描述的。然后随着业务的增加，应用里面存在了大量的 XML 配置，这些配置除包括 Spring 框架自身的 Bean 配置外，还包括了其他框架的集成配置等，到最后 XML 文件变得臃肿不堪，难以阅读和管理。同时，XML 文件内容本身不像 Java 文件那样能够在编译期事先做类型校验，所以也就很难排查 XML 文件中的错误配置。

以下展示的是在 Spring 应用中非常常见的 XML 配置管理 Bean 的方式。

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean name="john-classic" class="com.example.Person">
    <property name="name" value="Way Lau"/>
    <property name="spouse" ref="jane"/>
  </bean>

  <bean name="john-modern"
    class="com.example.Person"
    p:name="Way Lau"
    p:spouse-ref="jane"/>

  <bean name="jane" class="com.example.Person">
    <property name="name" value="Jane Doe"/>
  </bean>
</beans>
```

1.1.3 革新

针对上面传统企业级应用开发过程中的痛点，Java 及 Spring 框架也都提出了变革。例如，Java 5 引入的注解技术，就能更好描述 Java 程序；EJB 3 也向 Hibernate 等框架吸收了大量的精华（甚至请 Hibernate 的作者来设计 EJB 3），从而大大改善其实体 Bean 的重耦合的现状；从 Spring 3 开始，也引入了 Java 配置的方式来管理 Bean，从而大量减少了 XML 的使用，甚至是零配置。

1.1.4 约定大于配置

实现程序的零配置，其核心思想就是“约定大于配置”（Convention over Configuration）。约定大于配置是一个简单的概念，即系统、类库、框架应该假定合理的默认值，而无须提供不必要的配置。在大部分情况下，用户会发现使用框架提供的默认值会让自己所涉及的项目运行得更快。

零配置并不是完全没有配置，而是通过约定来减少配置，特别是减少 XML 文件的数量。

实现约定大于配置主要从以下几个方面入手。

1. 约定代码结构或命名规范来减少配置数量

如果模型中有个名为 Sale 的类，那么数据库中对应的表就会默认命名为“sale”。只有在偏离这一约定时，如将该表命名为“products_sold”，才需编写有关这个名称的配置。

例如，EJB3 持久化，将一个特殊的 Bean 持久化，设计者所需要做的只是将这个类标注为 @Entity。框架将会假定表名和列名是基于类名和属性名。系统也提供了一些钩子，当有需要时设计者可以重新编写这些名称。下面是一个表与实体映射的例子。

```
@Entity // 实体
public class User implements UserDetails {

    private static final long serialVersionUID = 1L;

    @Id // 主键
    @GeneratedValue(strategy=GenerationType.IDENTITY) // 自增策略
    private Long id; // 实体一个唯一标识

    @Column(nullable = false, length = 20) // 映射为字段，值不能为空
    private String name;

    @Column(nullable = false, length = 50, unique = true)
    private String email;

    // 以下省略 getter/setter 方法
}
```

例如，Maven 项目约定，在没有自定义的情况下，源代码假定是在 /src/main/java 目录，资源文件假定是在 /src/main/resources 目录。测试代码假定是在 /src/test 目录。项目假定会产生一个 JAR

文件。Maven 假定想要把编译好的字节码放到 `/target/classes`，并且在 `/target` 创建一个可分发的 JAR 文件。Maven 对约定优于配置的应用不仅仅是简单的目录位置，Maven 的核心插件使用了一组通用的约定，以用来编译源代码，打包可分发的构件，生成 Web 站点，以及许多其他的过程。Maven 的力量来自它的“武断”，它有一个定义好的生命周期和一组知道如何构建和装配软件的通用插件。如果遵循这些约定，开发者只需将自己开发的源代码放到正确的目录，而后 Maven 将会帮你处理剩下的事情。

例如，Yeoman 创建项目，只需一行代码：

```
yeoman init angular
```

就会创建整个详细结构，包括渲染路由的框架、单元测试等。

例如，HTML 5 Boilerplate (<http://html5boilerplate.com>) 项目，提供了制作 APP 的默认模板及文件路径规范，无论是网站或者富 UI 的 APP，都可以采用这个模板作为起步。HTML5 Boilerplate 的模板核心部分不过 30 行，但是每一行都可谓千锤百炼，可以用最小的消耗解决一些前端的顽固问题。以下是 HTML5 Boilerplate 初始化项目的目录结构。

```
C:.\
|
|  .gitattributes
|  .gitignore
|  .htaccess
|  404.html
|  apple-touch-icon-114x114-precomposed.png
|  apple-touch-icon-144x144-precomposed.png
|  apple-touch-icon-57x57-precomposed.png
|  apple-touch-icon-72x72-precomposed.png
|  apple-touch-icon-precomposed.png
|  apple-touch-icon.png
|  CHANGELOG.md
|  CONTRIBUTING.md
|  crossdomain.xml
|  favicon.ico
|  humans.txt
|  index.html
|  LICENSE.md
|  README.md
|  robots.txt
|
|  └─css
|      main.css
|      normalize.css
|
|  └─doc
|      crossdomain.md
|      css.md
|      extend.md
```

```

faq.md
html.md
js.md
misc.md
TOC.md
usage.md
img
.gitignore
js
main.js
plugins.js
vendor
jquery-1.9.1.min.js
modernizr-2.6.2.min.js

```

2. 采用更简洁的配置方式来替代 XML

很多配置方式都比 XML 简洁。

例如，hibernate.properties 的配置方式如下。

```

hibernate.connection.driver_class = org.postgresql.Driver
hibernate.connection.url = jdbc:postgresql://localhost/mydatabase
hibernate.connection.username = myuser
hibernate.connection.password = secret
hibernate.c3p0.min_size=5
hibernate.c3p0.max_size=20
hibernate.c3p0.timeout=1800
hibernate.c3p0.max_statements=50
hibernate.dialect = org.hibernate.dialect.PostgreSQL82Dialect

```

例如，Apache Shiro 的 ini 配置方式如下。

```

[users]
root = secret, admin
guest = guest, guest
presidentskroob = 12345, president
darkhelmet = ludicrousspeed, darklord, schwartz
lonestarr = vespa, goodguy, schwartz

[roles]
admin = *
schwartz = lightsaber:*
goodguy = winnebago:drive:eagle5

```

又如，Hibernate 通过以下 Java 代码来进行配置。

```

Configuration cfg = new Configuration()

```