

Spring For All技术社区

强力推荐

前瞻

本书基于Spring 5版本编写，除讲解Spring 5新特性外，还介绍了REST服务、响应式Web开发、微服务设计、Spring Boot及Spring Cloud等方面的前瞻技术。

全面

本书力求全面介绍Spring框架，涵盖了包括Spring核心、测试、数据访问、Web开发、响应式编程、系统集成及微服务等方面在内的共26章的内容，可以说是Spring技术的“百科全书”。

Spring 5

开发大全

柳伟卫◎著

实用

本书除了讲解Spring的理论知识外，还在每个知识点上辅大量的实战案例，使理论可以联系实际，具备更强的可操作性。

简明

目录标注有“新功能”“实战”“难点”及“重点”等标识，方便读者做好知识点的归纳和总结。

官方

以Spring官方文档和Spring框架源码为主要参考依据，确保知识点的准确性。



北京大学出版社
PEKING UNIVERSITY PRESS

Spring5

开发大全

柳伟卫◎著



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 提 要

自 Spring 诞生以来,以 Spring 技术栈为核心的企业级应用开发方式深入人心,已经成为 Java EE 开发的最佳实践。随着 Spring 5 版本的发布, Spring 自身也发生了巨大的变革。本书力求全面介绍 Spring 框架,涵盖了 Spring 核心、测试、数据访问、Web 开发、响应式编程、系统集成及微服务等方面在内的共 26 章的内容,可以说是 Spring 技术的“百科全书”。同时,本书基于 Spring 5 版本来编写,除了涉及 Spring 5 版本的新特性外,还介绍了 REST 服务、响应式 Web 开发、微服务设计、Spring Boot、Spring Cloud 等方面的前瞻技术。而且除了讲解 Spring 的理论知识外,还在每个知识点上辅以大量的代码案例,使理论可以联系实际,具备更强的可操作性。

本书主要面向的是 Java 开发者,以及对以 Spring 为核心的 Java EE 开发感兴趣的计算机专业的学生、软件开发人员和系统架构师。

图书在版编目(CIP)数据

Spring 5 开发大全 / 柳伟卫著. — 北京:北京大学出版社, 2018.10
ISBN 978-7-301-29882-4

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208436号

书 名 Spring 5 开发大全

SPRING 5 KAIFA DAQUAN

著作责任者 柳伟卫 著

责任编辑 吴晓月

标准书号 ISBN 978-7-301-29882-4

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 010-62752015 发行部 010-62750672 编辑部 010-62570390

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 38.25印张 888千字

2018年10月第1版 2018年10月第1次印刷

印 数 1-4000册

定 价 119.00 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

本书献给我的父母，愿他们健康长寿！

前言

Preface

写作背景

Spring 自诞生以来，一直被广大开发者作为 Java 企业级应用开发的首选。无论是早年流行的 SSH 框架（Spring、Struts、Hibernate），还是近些年盛行的 SSM 组合（Spring、Spring MVC、MyBatis），唯一不变的就是 Spring。伴随互联网十几年的风云变迁，Spring 也不断地进行着技术革命，向着云计算、平台化方向发展。

Spring 至今仍然奉行着最初的宗旨——简化企业级应用的开发。特别是 Spring Boot、Spring Cloud 等项目的诞生，进一步简化了基于 Spring 的企业级、分布式系统的应用开发方式。为此，笔者在 2014 年以开源方式翻译了《Spring Framework 4.x 参考文档》，备受广大开发者关注。在 2017 年，笔者编写了一系列关于 Spring 开发实战的书籍及培训视频教程，包括《Spring Boot 企业级应用开发实战》和《Spring Cloud 微服务架构开发实战》等^①，致力于让广大读者能够真正领略 Spring 所带来的力量和乐趣。

目前，Spring 5 已经发布，并带来了大量的新特性。鉴于市面上关于 Spring 5 的介绍资料比较匮乏，故笔者撰写本书以补空白，助力国内开发者能够享受到新技术所带来的便利。

本书特色

本书具有以下特色。

（1）**全面**：本书全面介绍 Spring 框架，涵盖了 Spring 核心、测试、数据访问、系统集成、Web 开发、响应式编程及微服务等方面在内的共 26 章的内容，可以说是 Spring 技术的“百科全书”。

（2）**前瞻**：本书基于 Spring 5 版本来编写，除了涉及 Spring 5 版本的新特性外，还介绍了 REST 服务、响应式 Web 开发、微服务设计、Spring Boot 及 Spring Cloud 等方面的前瞻技术。

（3）**实用**：本书除了讲解 Spring 的理论知识外，还在每个知识点上辅以大量的实战案例，使理论可以联系实际，具备更强的可操作性。

（4）**简明**：目录标注有“新功能”“难点”及“重点”等标识，方便读者做好知识点的归纳和总结。

（5）**权威**：以 Spring 官方文档和 Spring 框架源码为主要参考依据，确保知识点的权威性。

源代码

本书提供源代码下载，下载地址为 <http://github.com/waylau/spring-5-book>。

^① 有关笔者的书籍、教程介绍，可见<https://waylau.com/books/>。

本书所采用的技术及相关版本

技术版本是非常重要的，因为不同的版本之间存在兼容性问题，而且不同版本中的软件所对应的功能也是不同的。本书所列出的技术在版本上相对较新，并且都是经过笔者实际测试的。读者在自行编写代码时，可以参考本书所列出的版本，从而避免很多因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本书所采用的开发环境，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“附录 D”的内容。

本书示例采用 Eclipse 编写，但示例源码与具体的 IDE 无关，读者可以选择适合自己的 IDE，如 IntelliJ IDEA 和 NetBeans 等。运行本书示例，请确保 JDK 版本不低于 8。

勘误和交流

本书如有勘误，会在 <http://github.com/waylau/spring-5-book> 上进行发布。由于笔者编写能力有限，加上时间仓促，疏漏之处在所难免，欢迎读者批评指正。

读者也可以通过以下方式直接联系笔者。

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的各位工作人员为本书的出版所做出的努力。

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，在此感谢家人对我工作的理解和支持。

感谢 Spring 团队为 Java 社区贡献了这么优秀的框架，由衷地希望 Spring 框架发展得越来越好！

最后，特别感谢 Rod Johnson、Juergen Hoeller 和 Yann Caroff，是他们最早创建了 Spring。

柳伟卫

目录

Contents

第 1 章 Spring 5 概述	1
1.1 Spring 与 Java EE.....	2
1.2 Spring 简史.....	6
1.3 Spring 5 的新特性.....	8
实战 1.4 快速开启第一个 Spring 应用	12
1.5 Gradle 与 Maven 的抉择	17
第 2 章 Spring 框架核心概念	26
2.1 Spring 框架总览.....	27
2.2 IoC 容器.....	32
2.3 AOP 编程.....	93
2.4 资源处理	107
2.5 Bean 验证.....	111
2.6 表达式语言 SpEL.....	113
2.7 数据缓冲器和编解码器	129
2.8 空安全	130
第 3 章 测试	131
3.1 测试概述	132
3.2 测试的类型和范围	134
3.3 如何进行微服务的测试.....	137
第 4 章 单元测试.....	147
4.1 Mock 对象	148
4.2 测试工具类.....	149

第 5 章	集成测试.....	150
5.1	集成测试概述	151
5.2	测试相关的注解	153
5.3	Spring TestContext 框架	166
5.4	Spring MVC Test 框架	186
★新功能	5.5 WebTestClient	201
第 6 章	事务管理.....	205
6.1	事务管理概述	206
6.2	通过事务实现资源同步	210
6.3	声明式事务管理	212
6.4	编程式事务管理	225
6.5	事件中的事务	226
第 7 章	DAO.....	228
7.1	DAO 概述	229
7.2	DAO 常用异常类	229
7.3	DAO 常用注解	230
第 8 章	基于 JDBC 的数据访问	231
8.1	Spring JDBC 概述	232
8.2	JDBC 核心类	234
8.3	控制数据库连接	248
8.4	批处理	251
8.5	SimpleJdbc 类	254
8.6	JDBC 转为对象模型	258
8.7	内嵌数据库	271
8.8	初始化 DataSource	280
第 9 章	基于 ORM 的数据访问	281
9.1	Spring ORM 概述	282
9.2	ORM 集成注意事项	283
9.3	集成 Hibernate	284
9.4	JPA	290

第 10 章 XML 与对象的转换	296
10.1 XML 解析概述	297
10.2 XML 的序列化与反序列化	297
10.3 常用 XML 解析工具	302
第 11 章 Spring Web MVC	311
11.1 Spring Web MVC 概述	312
11.2 DispatcherServlet	312
11.3 过滤器	323
11.4 控制器	325
11.5 URI 处理	335
11.6 异常处理	337
11.7 异步请求	340
11.8 CORS 处理	344
11.9 HTTP 缓存	348
11.10 MVC 配置	350
11.11 视图处理	358
★新功能 11.12 HTTP/2	362
实战 11.13 基于 Spring Web MVC 的 REST 接口	363
第 12 章 REST 客户端	369
12.1 RestTemplate	370
★新功能 12.2 WebClient	370
实战 12.3 基于 RestTemplate 的天气预报服务	371
第 13 章 WebSocket	379
13.1 WebSocket 概述	380
13.2 WebSocket 常用 API	381
13.3 SockJS	388
13.4 STOMP	392
实战 13.5 基于 STOMP 的聊天室	408
第 14 章 Spring WebFlux	418
★新功能 14.1 响应式编程概述	419
★新功能 14.2 Spring 中的响应式编程	422

★新功能	14.3	DispatcherHandler	426
★新功能	14.4	控制器	427
★新功能	14.5	常用函数	429
★新功能	14.6	WebFlux 相关配置	434
★新功能	14.7	CORS 处理	439

第 15 章

响应式编程中的 WebClient

442

★新功能	重点	15.1	retrieve() 方法	443
★新功能	重点	15.2	exchange() 方法	444
★新功能		15.3	请求主体	444
★新功能		15.4	生成器	446
★新功能		15.5	过滤器	447
★新功能	实战	15.6	基于 WebClient 的文件上传、下载	447

第 16 章

响应式编程中的 WebSocket

451

	16.1	WebSocket 概述	452
	16.2	WebSocket 常用 API	452
★新功能	16.3	WebSocketClient	454

第 17 章

常用集成模式

455

	17.1	Spring 集成模式概述	456
	17.2	使用 RMI	456
	17.3	使用 Hessian	457
	17.4	使用 HTTP	459
	17.5	Web 服务	460
	17.6	JMS	463
	17.7	REST 服务	465

第 18 章

EJB 集成

468

	18.1	EJB 集成概述	469
	18.2	EJB 集成的实现	469

第 19 章

JMS 集成

471

	19.1	JMS 集成概述	472
	19.2	Spring JMS	473

19.3	发送消息	475
19.4	接收消息	477
19.5	JCA 消息端点	480
19.6	基于注解的监听器	481
19.7	JMS 命名空间	484
难点	19.8 基于 JMS 的消息发送、接收	485
第 20 章	JMX 集成	498
20.1	JMX 集成概述	499
20.2	bean 转为 JMX	499
20.3	bean 的控制管理	503
20.4	通知	507
第 21 章	JCA CCI 集成	510
21.1	JCA CCI 集成概述	511
21.2	配置 CCI	511
21.3	使用 CCI 进行访问	513
21.4	CCI 访问对象建模	516
21.5	CCI 中的事务处理	518
第 22 章	使用 E-mail	519
22.1	使用 E-mail 概述	520
22.2	实现发送 E-mail	520
22.3	使用 MimeMessageHelper	522
实战	22.4 实现 E-mail 服务器	524
第 23 章	任务执行与调度	529
23.1	任务执行与调度概述	530
23.2	TaskExecutor	530
23.3	TaskScheduler	532
23.4	任务调度及异步执行	533
23.5	使用 Quartz Scheduler	537
实战	23.6 基于 Quartz Scheduler 的天气预报系统	538

第 24 章 缓存	543
24.1 缓存概述	544
24.2 声明式缓存注解	544
24.3 JCache 注解	548
24.4 基于 XML 的声明式缓存	549
24.5 配置缓存存储	550
实战 24.6 基于缓存的天气预报系统	551
第 25 章 Spring Boot.....	555
25.1 从单块架构到微服务架构	556
25.2 微服务设计原则	559
25.3 Spring Boot 概述	562
实战 25.4 开启第一个 Spring Boot 项目	568
第 26 章 Spring Cloud.....	577
★新功能 26.1 Spring Cloud 概述	578
★新功能 26.2 Spring Cloud 入门配置	579
★新功能 26.3 Spring Cloud 的子项目介绍	582
★新功能 实战 26.4 实现微服务的注册与发现	585
附录	592
附录 A EJB 规范摘要	593
附录 B Bean Validation 内置约束	595
附录 C 提升 Gradle 的构建速度	597
附录 D 本书所采用的技术及相关版本	598
参考文献	599

第1章

Spring 5 概述

1.1 Spring 与 Java EE

Spring 诞生之初是以 J2EE 的挑衅者身份而为广大 Java 开发者所熟知的。特别是当时 J2EE 平台中的 EJB(Enterprise Java Beans)标准, 由于其设计本身的缺陷, 导致在开发过程中使用非常复杂, 代码侵入性很强。又由于 EJB 是依赖于容器实现的, 因此进行单元测试也变得极其困难, 最终的后果是大多数开发者对 Java 企业级开发望而却步。

为此, Rod Johnson 为 Java 世界带来了 Spring, 它的目标就是要简化 Java 企业级开发。

1.1.1 Java 平台发展简史

作为一门最受欢迎的编程语言, Java 语言在经历了 20 多年的发展后, 已然成为开发者首选的“利器”。在 2018 年年初的 TIOBE 编程语言排行榜中, Java 位居榜首。图 1-1 所示的是 2018 年 4 月 TIOBE 编程语言排行榜情况。回顾历史, Java 语言的排行也一直是名列三甲。

Apr 2018	Apr 2017	Change	Programming Language	Ratings	Change
1	1		Java	15.777%	+0.21%
2	2		C	13.589%	+6.62%
3	3		C++	7.218%	+2.66%
4	5	^	Python	5.803%	+2.35%
5	4	v	C#	5.265%	+1.69%
6	7	^	Visual Basic .NET	4.947%	+1.70%
7	6	v	PHP	4.218%	+0.84%
8	8		JavaScript	3.492%	+0.64%
9	-	^^	SQL	2.650%	+2.65%
10	11	^	Ruby	2.018%	-0.29%
11	9	v	Delphi/Object Pascal	1.961%	-0.86%
12	15	^	R	1.806%	-0.33%

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。

1991 年, 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, 并命名为“Oak”(橡树), 以他办公室外面的树来命名。然而, 由于当时的机顶盒项目并没有竞标成功, 于是 Oak 被阴差阳错地应用到万维网。1994 年, 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 HotJava), 该浏览器可以直接用来运行 Java 小程序(Java Applet)。1995 年, Oak 改名为 Java。由于 Java

2018 年 2 月 26 日, Eclipse 基金会社区正式将 Java EE 更名为 Jakarta EE, 也就是说, 下个 Java 企业级发布版本将可能会命名为 Jakarta EE 9。这个名称来自 Jakarta——一个早期的 Apache 开源项目。但该改名行为, 并未得到 Java 社区的支持。Java EE Guardians 社区负责人 Reza Rahman 就 Java EE 重命名的问题做了一项 Twitter 调查, 结果显示, 68% 的 Java 开发者认为应该保留 Java EE 名称。在本书中, 为了避免混淆, 仍然统一采用 Java EE 命名来代表 J2EE 或 Jakarta EE。

2018 年 3 月 20 日, Java 10 如期发布, 包含了 109 项新特性。

1.1.2 Java EE 现状

Java EE 平台目前已经发展到了第 8 个大版本。同时, Oracle 公司将 Java EE 8 移交给了开源组织 Eclipse 基金会来进行管理, 这意味 Java EE 未来将会更加开放, 版本更新速度也会加快。Eclipse 基金会将未来开源版本的 Java EE 命名为 EE4J (Eclipse Enterprise for Java)。

1. EE4J 的使命

EE4J 创建之初就包含了如下使命。

① EE4J 的使命是创建标准 API, 这些 API 的实现及用于 Java 运行时的技术兼容性工具包, 支持服务器端和云本机应用程序的开发、部署和管理。EE4J 基于 Java EE 标准, 并使用 Java EE 8 作为创建新标准的基准。

② EE4J 能够使用灵活的许可授权和平台演化的开放治理流程。开放的过程不依赖于单一的供应商或领导者, 鼓励参与和创新, 并为整个社区的集体利益服务。

③ EE4J 通过使用通用流程和通用兼容性要求来定义一套完整的标准, 以确定其组成项目之间的通用性。EE4J 通过在 Java EE 8 和 EE4J 版本之间提供兼容性, 为现有用户和新用户提供兼容性。

由于 EE4J 是基于 Java EE 8 技术标准的, 且与 Java EE 8 的 API 是完全兼容的, 因此用户在从 Java EE 8 切换到 EE4J 项目时不会有难度。同时, EE4J 由强大的供应商和强大的社区作为支撑, 所提供的创新解决方案更能够满足现有用户的新需求, 吸引新用户。

2. 不再使用 JCP

长期以来, Java 规范的制定都是由 JCP (Java Community Process) 来执行。JCP 是一种针对 Java 技术开发标准技术规范的机制。它向所有人开放, 任何人都可以参与审核, 并提供 Java 规范请求 (JSR) 反馈。任何人都可以注册成为 JCP 成员, 并加入 JSR 专家组, 成员甚至可以提交自己的 JSR 提案。

2018 年 1 月, Oracle 公司表示将来不再支持或建议使用 JCP 来增强 Java EE, 而是建议并支持使用 EE4J 推动的过程对 Java EE 8 规范进行功能增强。在发给 EE4J 社区的邮件中, Oracle WebLogic Server 产品管理高级主管 Will Lyons 传达了这则消息。

重点 1.1.3 Spring 与 Java EE 的关系

早年,“Spring 之父” Rod Johnson 对传统的 Java EE 系统框架臃肿、低效、脱离现实的种种现状提出了质疑,并积极寻求探索革新。Rod Johnson 在 2002 年编著的 *Expert One-on-One J2EE Design and Development* 一书中,可以说一针见血地指出了当时 Java EE 架构在实际开发中的种种弊端。

在该书中, Rod Johnson 表明了如下的观点。

① Java EE 不能“包治百病”。任何技术都不可能是“银弹”,即便当时 Java EE 已经是企业级开发的最好选择了,但仍不能说 Java EE 可以解决任何问题。

②小心“正统”的开发方式。特别是 Sun 公司所推崇的 Java EE 开发方式,在实际开发中也并不完全适用,甚至有时在误导开发者。Rod Johnson 坦言,“所谓‘正统’的开发方式,都是面向规范的,而不是面向实际要解决的问题的。”这必然就导致了像 EJB 这种复杂规范无法真实落地的情况出现。

Rod Johnson 正是洞察到了传统 Java EE 开发上的弊端,从而推出了 Spring 框架,致力于解决 Java EE 开发上的问题。

虽然 Spring 喊出了“Without EJB”(不需要 EJB)的口号,但本质上,它并非想要完全挑战整个 Java EE 平台。Spring 力图冲破 Java EE 传统开发的困境,从实际需求出发,着眼于构建轻便、灵巧并易于开发、测试和部署的轻量级开发框架。

Spring 在很大程度上是为了改进当时以 EJB 为核心的 Java EE 开发方式。Rod Johnson 对 EJB 各种臃肿的结构进行了逐一的分析和否定,并分别以简洁实用的方式替换。EJB 是一种复杂的技术,虽然很好地解决了一些问题,但在许多情况下增加了比其商业价值更大的复杂性问题。

传统 Java EE 应用的开发效率是低下的,应用服务器厂商对各种技术的支持并没有真正统一,导致 Java EE 应用并没有真正实现“Write Once, Run Anywhere”(一次编写,各处运行)的承诺。Spring 作为开源的中间件,独立于各种应用服务器,甚至无须应用服务器的支持,也能提供应用服务器的功能,如声明式事务、事务处理等。

Spring 致力于 Java EE 应用各层的解决方案,而不是仅仅专注于某一层方案。可以说, Spring 是企业应用开发的“一站式”选择,并贯穿表现层、业务层及持久层。然而, Spring 并不想取代那些已有的框架,而是要与它们无缝整合。

虽然从表面上来看,有些人会认为 Spring 和 Java EE 是竞争关系,实际上 Spring 是对 Java EE 的改进和补充。Spring 本身也集成非常多的 Java EE 平台规范,如 Servlet API (JSR 340)、WebSocket API (JSR 356)、Concurrency Utilities (JSR 236)、JSON Binding API (JSR 367)、Bean Validation (JSR 303)、JPA (JSR 338)、JMS (JSR 914)、Dependency Injection (JSR 330)、Common Annotations (JSR 250) 等。

简言之, Spring 的目标就是要简化 Java EE 开发。如今, Spring 俨然成为 Java EE 的代名词,成为构建 Java EE 应用的事实标准。大多数 Java 项目都会采用 Spring 作为框架的首选。