

# 前言

在过去的近二十年来，笔者主要是在 Java 领域从事软件开发、创作和教学，有幸见证了 Java 技术的整个发展历程。Java 技术的发展与这二十年来整个软件技术乃至工业技术的发展，都遵循一种共同的产品开发趋势：那就是从独立、不可重用并且不可拆卸的产品逐渐发展成为依靠各种框架和可重用组件搭建出来的巨大的组合产品。

Spring 框架为 Java Web 开发提供了全方位的支持，最初它主要是通过 Spring MVC 分支框架来控制 Web 应用处理客户请求的流程，为数据验证、异常处理和国际化提供简单易用的 API。

随着 Web 技术的普及，一些超大型网站（如淘宝和京东等）的日均客户访问量超过了千万，这对网站的并发性能和运行性能提出了新的挑战。为了克服这些挑战，在 Java Web 开发领域，Spring 框架以开放的姿态吸收或自行研发了一些列新的技术解决方案和软件，比如：

- 服务器和客户端交换 JSON 格式的数据，从而更加方便灵活地表达各种复杂的数据结构。
- 通过前后端分离，来减轻服务器端的负荷，让大量客户主机分担一部分处理简单业务逻辑（如客户端数据验证）以及展示业务数据的任务。
- 为操纵数据库的 CRUD（Create、Retrieve、Update 和 Delete，新增、查询、更新和删除）操作提供统一的访问方式。
- 通过 SSE（Server-Sent Event，服务器端发送事件）技术使得浏览器能依靠轮询来捕获服务器端发送数据的事件，并把接收到的数据显示到网页上，从而产生服务器主动向客户端推送数据的效果。
- 通过 Spring Data API 和支持 ORM（Object-Relation Mapping，对象-关系映射）的独立的持久化层来访问数据库。
- 通过 WebFlux 响应式编程来实现 Web 服务器与客户端的异步非阻塞通信。
- 通过 WebSocket 实现 Web 服务器与客户端的双向通信。
- 通过 Web 服务技术实现 B2B（Business To Business，企业到企业）方式的远程服务调用。
- 通过微服务技术实现分布式的 Web 应用，把 Web 应用的各个模块分布到不同的主机节点上，从而扩充软件系统的 CPU、内存和硬盘等硬件资源，来提高应用的运行性能和并发性能。

本书介绍了用 Spring 来开发 Java Web 应用的各种技术，包括：Spring MVC 框架、数据验证、拦截器、异常处理机制、Web 应用的国际化、服务器端异步处理客户请求、AOP 面向切面编程、通过 Spring JDBC API 访问数据库、通过 Spring Data API 访问数据库、RESTful 风格编程、WebFlux 响应式编程、Spring 和 CXF 整合开发 Web 服务，以及用 Spring Cloud 框架开发微服务等。

本书不仅介绍了各种框架软件的使用法，还对目前比较流行的技术解决方案所蕴含的思想做了深刻的阐述，包括：MVC 设计模式、依赖注入、控制反转、前后端分离、服务器端推送、Token 机制、AOP 编程思想、对象-关系映射、响应式编程模型、RESTful 架构、分布式的 Web 服务架构，以及分布式的微服务架构等。

随着软件的不断更新，软件的 API 的用法也在不断变化中，这使得应用程序也要做相应的调整。为了让读者能轻松地紧跟软件技术发展的步伐，本书花了不少篇幅，用形象的生

活化的例子来帮助读者理解各种技术中蕴含的思想。只有领悟了这些思想，才能在技术的发展中把握主动权，依照高屋建瓴的格局来运用 Spring 以及和它集成的软件，开发出便于维护、扩展和性能卓越的 Java Web 应用。

## 本书的组织结构和主要内容

本书按照由浅入深、前后照应的顺序来安排内容，主要包含以下内容：

- 第 1 章至第 10 章：详细介绍 Spring MVC 框架的各种用法。
- 第 11 章：介绍 AOP (Aspect Oriented Programing) 面向切面编程技术，以及利用 AOP 来输出日志。
- 第 12 章至第 15 章：侧重介绍模型层的开发，并介绍如何开发包含视图、控制器和模型层的完整范例；在模型层分离出访问数据库的 DAO (Data Access Object, 数据访问对象) 层，在 DAO 层通过 Spring JDBC API 以及 Spring Data API 访问数据库；创建采用 RESTful 风格的 Web 应用，为操纵数据库的 CRUD 操作提供统一的访问方式。
- 第 16 章：介绍 Spring WebFlux 框架的用法，创建支持异步非阻塞通信的 Web 应用。
- 第 17 章：利用 Spring WebSocket API 实现 Web 服务器与客户端的双向通信，并介绍了一个实用的聊天室范例。
- 第 18 章：通过 Spring 和 CXF 的整合框架来开发 Web 服务，实现分布式的 B2B 方式的通信。
- 第 19 章：通过 Spring Cloud 框架来开发微服务，实现分布式的 Java Web 应用。

本书每一章都提供了具体的范例程序，所有的范例程序都由笔者亲自设计和编写，在本书配套源代码包中提供了完整的源代码。

## 这本书是否适合您

这本书适合所有的 Java 开发人员。通过阅读本书，读者既能掌握最新的 Spring 开发技术，又能领悟各种最新 Java Web 开发技术中蕴含的深刻思想，还能把握技术发展的趋势。本书既可以作为纯粹的 Spring 学习用书，也可以作为实用的 Spring 开发参考工具书。

笔者在 Java 领域陆陆续续创作了多本被广大读者一致认可的经典之作，如《Java 面向对象编程》、《Java 网络编程核心技术详解》、《Tomcat 与 Java Web 开发技术详解》和《精通 JPA 与 Hibernate: Java 对象持久化技术详解》。这些书和本书按照循序渐进的演绎方式，阐述了 Java 开发领域的大多数技术，是笔者近二十年来倾心为广大 Java 开发人员烹制的 Java 技术大餐。笔者期望通过这些书帮助 Java 开发人员既能轻松地上手，掌握开发 Java 应用的实用技能，而且能巩固核心知识、概念和原理，真正提高内在的 Java 开发技术功底。

## 写作规范

为了节省文章的篇幅，在本书中显示范例的源代码时，有时做了一些省略。对于 Java 类，省略显示 package 语句和 import 语句。本书大部分范例创建的 Java 类都位于 mypack 包下。对于 JavaBean 类，还省略显示了属性的 getXXX() 和 setXXX() 方法。

在本书提供的 SQL 语句中，表名和字段名都采用大写形式，而 SQL 关键字，如 select、from、insert、update 和 delete 等，都采用小写形式。

## 致谢

本书在编写过程中得到了 Spring 软件开发组织、Apache 软件开发组织和 Oracle 公司在技术上的大力支持。此外 JavaThinker.net 网站的网友为本书的编写提供了有益的帮助，清华大学出版社的编辑老师为本书做了精雕细琢的润色，进一步提升了书的品质。在此表示衷心

的感谢！尽管我们尽了最大努力，但本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正。

A handwritten signature in black ink, consisting of stylized Chinese characters. The characters appear to be '孙明' (Sun Ming) with a flourish underneath.