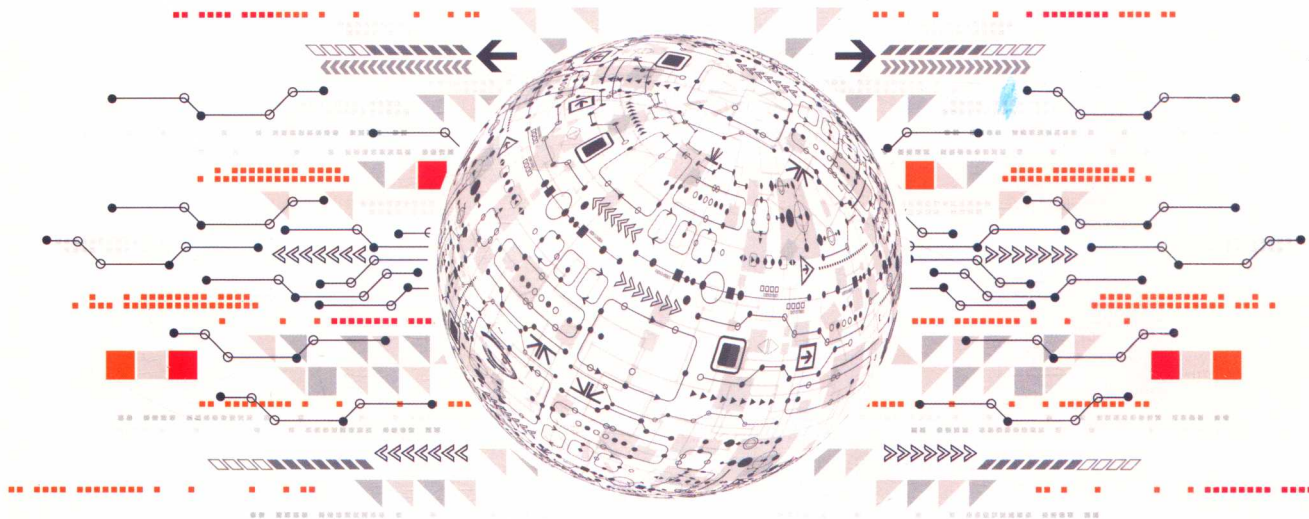




云时代架构系列

Broadview®
www.broadview.com.cn



互联网轻量级 SSM框架解密

Spring、Spring MVC、MyBatis源码深度剖析

李艳鹏 曲源 宋杨 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



云时代架构系列

互联网轻量级 SSM框架解密

Spring、Spring MVC、MyBatis源码深度剖析

李艳鹏 曲源 宋杨 编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

SSM 是 Spring、Spring MVC 和 MyBatis 框架的组合,是目前 Java 领域使用非常广泛也非常稳定的开源 Web 框架。本书以 SSM 的核心代码剖析为基础,突破 Java Web 研发瓶颈的束缚,选取 Spring、Spring MVC 和 MyBatis 框架中易于理解的版本,深入剖析了其中各个模块的实现,从代码中挖掘常用的设计模式,为读者理解 Spring 系列框架的可扩展设计艺术提供了方法论和优秀实践。

本书 Spring 源码剖析篇基于 Spring 4.3.2 版本,剖析了 Spring 上下文、Spring AOP 和 Spring 事务的实现,并通过实例展示了框架陷阱的隐蔽性及学习框架原理的必要性。Spring MVC 源码剖析篇基于 Spring MVC 3.0 版本,这个版本比较简单、核心清晰,便于读者理解透彻,这里主要讲解其中的设计模式及可插拔的设计思路。MyBatis 源码剖析篇基于 MyBatis 3.4 版本,帮助读者对 SQL 语言、JDBC 及数据访问方式有更深入的了解,也能看到工厂、Builder、代理、装饰者等设计模式在 MyBatis 中的大量应用。

本书对于互联网从业者,或者传统行业的 IT 工程师、架构师、技术经理、技术总监,以及想深耕 IT 行业的技术人员都有很强的借鉴性和实用价值。

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。

图书在版编目(CIP)数据

互联网轻量级 SSM 框架解密: Spring、Spring MVC、MyBatis 源码深度剖析 / 李艳鹏等编著. —北京: 电子工业出版社, 2019.3

(云时代架构)

ISBN 978-7-121-35954-5

I. ①互… II. ①李… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字 (2019) 第 014988 号

责任编辑: 张国霞

印 刷: 三河市良远印务有限公司

装 订: 三河市良远印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 33 字数: 740 千字

版 次: 2019 年 3 月第 1 版

印 次: 2019 年 3 月第 1 次印刷

印 数: 4000 册 定价: 109.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819, faq@phei.com.cn。



推荐序一

很高兴能为艳鹏的新书写序, 之前一直与艳鹏合作写书, 非常佩服其知识深度和广度, 以及对写作持之以恒的态度。

这是一本非常难得的 SSM 源码书。SSM 框架非常流行, 掌握这些技术是大势所趋, 但是想要成为 SSM 专家却并不容易, 而阅读本书是一个很好的捷径。

SSM, 是 Spring+Spring MVC+MyBatis 的缩写, 是继 SSH 之后, 目前比较主流的 Java EE 企业级框架, 适用于搭建各种大型的企业级应用系统。Spring 依赖注入 DI 来管理各层的组件, 使用 AOP (面向切面编程) 管理事务、日志、权限等。Spring MVC 代表 Model (模型)、View (视图)、Controller (控制) 接收外部请求并进行分发和处理。MyBatis 是基于 JDBC 的框架, 主要用来操作数据库, 并且将业务实体和数据表联系起来。

本书在核心源码分析和企业运用上, 由浅入深、由易到难地进行系统分析和讲解, 涉及 Spring、Spring MVC 和 MyBatis 的设计理念和整体架构、容器的基本实现、标签的解析、Bean 的加载、容器的功能扩展、IoC、AOP、事务、Spring 消息服务等内容, 能很好地指导读者使用 SSM 编写企业级应用, 并针对在编写代码的过程中如何优化代码给出切实可行的建议, 从而帮助读者全面提升实战能力。

本书语言简洁, 示例丰富, 可帮助读者迅速掌握使用 SSM 进行开发所需的各种技能, 适用于有一定 Java 编程经验或者想学习 SSM 的读者, 还适用于 Java 开发人员、测试人员等。

高级架构师、《分布式服务架构: 原理、设计与实战》

《可伸缩服务架构: 框架与中间件》

主要作者 杨彪



推荐序二

在互联网软件开发领域，Java 作为工业生产语言，常年蝉联 TOBLE 排行榜榜首。良好的社区活跃度和广泛的从业基础，也推动着 Java 的蓬勃发展，涌现出一批又一批的优秀框架。

在早期，程序员依靠 Struts+DAO 层打天下，大都经历过学习 EL 和手拼 SQL 的苦日子。Spring 在发布后很快就在社区流行，尽管是否使用 EJB 是当时 Java 圈备受争议的话题，但 SSH 三件套最终发展为企业应用的必选项。

进入移动互联网时代后，随着前后端分离、微服务等技术的冲击，不管是进行大规模服务化部署，还是进行小作坊快速上线，SSM 逐渐成为事实上的业界标准。程序员面试被问到 SSM 的实现原理、架构设计，似乎不啃上几遍源码，都不好意思跟面试官交流。

在面试“造火箭”、工作“螺丝钉”的大环境下，很多程序员虽然对这些框架使用得非常娴熟，但对底层原理及架构设计缺少足够的积累与认知，知其然却不知其所以然。面对复杂的类库继承关系、纯英文的源码及注释，很多工程师在学习时遇到重重障碍，我当时也将 Tomcat 源码读了 3 遍，但每每都很难打个通关。

如何帮助程序员快速读懂框架源码，熟悉其背后的设计哲学，掌握其实现上的技巧，既能在面对高阶面试官时侃侃而谈，又能在实际工作中灵活运用，而不是一次次鼓起勇气，却又在源码的漩涡中退却呢？

本书分 3 篇对 SSM 做了深入的源码剖析。每篇都先介绍其框架的主要优点，在互联网开发中的功能定位及模块划分；进而对各个模块进行抽丝剥茧的分析并且给出核心类库的 UML 图，同时对关键代码进行注释、解读，为读者呈现框架的精华部分；最后结合一个实战案例，对企业生产中的优秀实践进行复盘。

本书作者都是业内有多年实际操作经验的专家，在互联网领域积累了大量实战开发经验。本书是他们知识和经验的总结，是他们智慧与理念的结晶，相信各位读者可以通过本书解决源码阅读的痛点，迅速吃透 SSM，在工作中真正做到深入浅出、言必有据。

爱奇艺技术产品中心高级技术经理、支付中心技术负责人 张冲



推荐序三

随着互联网的飞速发展，从项目迭代到框架更新的速度之快都让人应接不暇，互联网人能做的就是快速跟进，而抓住纷繁复杂事务的本质和规律，会让我们走得更好。

软件是一个密切关注实践的领域，就像我们研究很多设计模式及设计思想，学习源码也是我们提升实践能力的好方式。而企业级应用至今很难绕开 B/S 结构，无论是在 PC 端还是在手机端，Spring 这样的开源框架都是经典的核心解决方案，并且 Spring Boot、Spring Cloud 等框架都有着很深的影响力，在 Java 软件工程实现领域是很难绕开的核心框架。

我读过 Spring 的 IoC、AOP 及 MVC，至今仍受益良多。好的框架能让你知道过去、现在和未来，Spring 可以算其中一个。

我发现本书抓住了中高级 Java 工程师的一些痛点并给出了很好的解决方案。

（1）高屋建瓴。框架基础及领域模型才是一个框架的灵魂。本书深入浅出地阐释了灵魂思想，使框架的能力及发展轨迹有章可循。

（2）重点突出。Spring 的核心功能全部突出，包括 IoC、AOP、MVC 及相关的 RPC 调用等。一个经典框架很难面面俱到，但是人们用得顺手的核心功能，一定是千锤百炼出来的。Spring 的核心设计模式在本书中都有重点阐释。

（3）注重系统化。作者花了较大的篇幅在 MyBatis 上，这也是本书的一大亮点。作者深入浅出地将 SQL 映射、SQL 解析、执行器、缓存机制等深层次内容呈现给我们，在面临自动化 SQL 生成及各种异构数据库适应的时候，很多延伸框架及解决方案就已经在我们的脑海中了。

程序员在自我修炼的过程中，若想获得超乎寻常的视野，则需要扎实地了解一下过去和现在，需要能够站在开发这些框架的大师角度去审视软件工程领域的优秀实践。这需要大量富有奉献、分享精神的作者引领我们前行，很高兴本书的作者们欣承此责。希望本书作者们能出更多这样的书，满足广大开源爱好者的强烈发展诉求。

新生支付有限公司副总裁 王志成



前言

SSM 是 Spring、Spring MVC 和 MyBatis 框架的组合，是目前 Java 领域使用非常广泛也非常稳定的开源 Web 框架，具有易搭建、开箱即用、配置丰富、扩展度高、运行稳定、开源社区活跃等优点。

本书以 SSM 的核心代码剖析为基础，突破 Java Web 研发瓶颈的束缚，选取 Spring、Spring MVC 和 MyBatis 框架中易于理解的版本，深入剖析了其中各个模块的实现，从代码中挖掘常用的设计模式，为读者理解 Spring 系列框架的可扩展设计艺术提供了方法论和优秀实践。

软件是一个密切关注实践的领域，源码是我们提升实践能力的优质学习资源，我们学习了各种设计模式，最终需要在源码中进行落地。当然，我们也需要从优秀的源码中挖掘设计模式及设计模式的应用场景，学习其中的设计艺术。

本书 Spring 源码剖析篇基于 Spring 4.3.2 版本，剖析了 Spring 上下文、Spring AOP 和 Spring 事务的实现，并通过实例展示了框架陷阱的隐蔽性及学习框架原理的必要性。

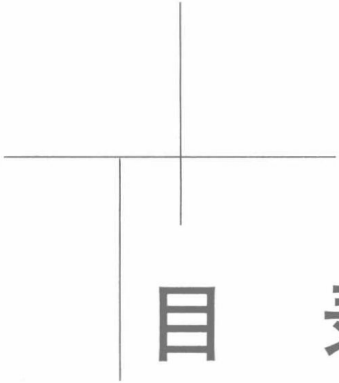
Spring MVC 源码剖析篇基于 Spring MVC 3.0 版本，这个版本相对简单、核心清晰，便于读者理解透彻，并主要讲解其中的设计模式及可插拔的设计思路。

MyBatis 源码剖析篇基于 MyBatis 3.4 版本。轻量化、易集成和 SQL 资源易管理等特性为 MyBatis 带来了大量的用户，本篇致力于使读者对 SQL 语言、JDBC 及数据访问方式有更深入的了解，也能看到工厂、Builder、代理、装饰者等设计模式在 MyBatis 中的大量应用。

我们在实际项目中会用到很多中间件，在搭建大型项目工程的时候，多数开发人员主要关注业务逻辑的实现，甚至不关心核心的非功能需求的质量，许多框架高级特性也经常

被忽略，导致项目质量不过关。本书通过源码剖析的方式，带领读者挖掘优秀框架的经典设计，窥探框架中高级特性的实现方式，让开源爱好者和应用开发者快速了解 SSM 框架的内部设计细节、设计思路、编程技巧及高级功能特性等内容，为在实际项目中更加熟练地使用框架并巧妙地避开框架内的陷阱提供帮助。

本书对于互联网从业者，或者传统行业的 IT 工程师、架构师、技术经理、技术总监，以及想深耕 IT 行业的技术人员都有很强的借鉴性和实用价值。



目 录

第 1 篇 深入剖析 Spring 源码

第 1 章	Spring 基础介绍	2
1.1	Spring 的核心结构	2
1.2	Spring 的领域模型	6
第 2 章	Spring 上下文和容器	7
2.1	Spring 上下文的设计	7
2.2	Spring 容器 BeanFactory 的设计	11
2.3	Spring 父子上下文与容器	13
第 3 章	Spring 加载机制的设计与实现	18
3.1	Spring ApplicationContext 的加载及源码实现	18
3.2	Spring XML 文件标签加载解析及自定义	27
3.3	Spring 注解的加载及自动注入	31
第 4 章	Spring Bean 探秘	45
4.1	Spring Bean 的定义和注册设计	45
4.2	Spring Bean 的定义模型	47
4.3	Spring Bean 的运行（获取、创建）实现	48
4.4	Spring Bean 的依赖注入的实现	59
4.5	Spring Bean 的初始化	70

第 5 章	Spring 代理与 AOP	74
5.1	Spring 代理的设计及 JDK、CGLIB 动态代理	75
5.2	Spring AOP 的设计	82
5.3	Spring AOP 的加载和执行机制	83
5.3.1	Spring AOP 的加载及源码解析	83
5.3.2	Spring AOP 的创建执行及源码解析	88
5.4	Spring 事务管理设计及源码	101
5.5	Spring 事务传播机制	115
第 6 章	Spring 实战	118
6.1	对 Spring 重复 AOP 问题的分析	118
6.2	Spring Bean 循环依赖的问题	125
 第 2 篇 深入剖析 Spring MVC 源码 		
第 7 章	MVC 简介	138
7.1	MVC 的体系结构和工作原理	138
7.1.1	控制器	139
7.1.2	视图	139
7.1.3	模型	140
7.2	Web MVC 的体系结构和工作原理	140
第 8 章	Spring Web MVC 工作流	142
8.1	组件及其接口	142
8.1.1	DispatcherServlet	143
8.1.2	处理器映射	143
8.1.3	处理器适配器	144
8.1.4	处理器与控制器	145
8.1.5	视图解析器	145
8.1.6	视图	146
8.2	组件间的协调通信	146

第 9 章	DispatcherServlet 的实现	148
9.1	深入剖析 GenericServlet 和 HttpServlet	150
9.1.1	HTTP 和 Servlet 规范简介	150
9.1.2	Servlet 和 GenericServlet 详解	152
9.1.3	HttpServlet 详解	152
9.2	深入剖析 DispatcherServlet	159
9.2.1	HttpServletBean 详解	161
9.2.2	FrameworkServlet 详解	162
9.2.3	DispatchServlet 详解	166
9.3	根共享环境的加载	182
9.3.1	基于 Servlet 环境监听器的实现结构	182
9.3.2	多级 Spring 环境的加载方式	189
第 10 章	基于简单控制器的流程实现	194
10.1	通过 Bean 名称 URL 处理器映射获取处理器执行链	194
10.1.1	抽象处理器映射	196
10.1.2	抽象 URL 处理器映射	199
10.1.3	抽象探测 URL 处理器映射	209
10.1.4	Bean 名称 URL 处理器映射	210
10.2	通过处理器适配器把请求转接给处理器	211
10.2.1	简单控制处理适配器的设计	211
10.2.2	表单控制器处理 HTTP 请求的流程	212
10.3	对控制器类体系结构的深入剖析	214
10.3.1	Web 内容产生器	215
10.3.2	抽象控制器类	217
10.3.3	基本命令控制器	218
10.3.4	抽象表单控制器	222
10.3.5	简单表单控制器	229
第 11 章	基于注解控制器的流程实现	230
11.1	默认注解处理器映射的实现	230

11.2	注解处理器适配器的架构设计	237
11.3	深入剖析注解控制器的处理流程	238
11.3.1	解析处理器方法	241
11.3.2	解析处理器方法的参数	253
11.3.3	绑定、初始化领域模型和管理领域模型	272
11.3.4	调用处理器方法	278
11.3.5	处理方法返回值和隐式模型到模型或视图的映射	281
11.3.6	如何更新模型数据	286
第 12 章	基于 HTTP 请求处理器实现 RPC	288
12.1	深入剖析 RPC 客户端的实现	289
12.2	深入剖析 RPC 服务端的实现	299
第 13 章	深入剖析处理器映射、处理器适配器及处理器的实现	311
13.1	处理器映射的实现架构	311
13.1.1	处理器映射实现类	312
13.1.2	处理器映射抽象类	313
13.1.3	对处理器映射类的代码剖析	315
13.2	处理器适配器的实现架构	322
13.3	深入剖析处理器	325
13.3.1	简单控制器	325
13.3.2	注解控制器	339
13.3.3	HTTP 请求处理器	342
13.4	拦截器的实现架构	344
第 14 章	视图解析和视图显示	353
14.1	基于 URL 的视图解析器和视图	353
14.1.1	内部资源视图解析器和内部资源视图	365
14.1.2	瓦块视图解析器和瓦块视图	371
14.1.3	模板视图解析器和模板视图	373
14.1.4	XSLT 视图解析器和 XSLT 视图	377

14.2	更多的视图解析器	378
14.2.1	Bean 名称视图解析器	378
14.2.2	内容选择视图解析器	379
14.2.3	资源绑定视图解析器	383
14.2.4	XML 视图解析器	385
 第 3 篇 深入剖析 MyBatis 源码		
第 15 章 MyBatis 介绍		388
15.1	MyBatis 的历史	388
15.2	MyBatis 子项目	389
15.3	MyBatis 的自身定位	389
15.3.1	JPA 持久化框架	390
15.3.2	MyBatis 的功能	390
15.3.3	MyBatis 与 JPA 的异同	390
15.4	MyBatis 的架构	391
15.4.1	模块	391
15.4.2	MyBatis 的项目包	392
 第 16 章 构建阶段		394
16.1	关键类	394
16.2	关键时序	395
16.3	构建的入口：SqlSessionFactoryBuilder 和 SqlSessionFactory	396
16.4	配置（Configuration）和配置构造器（XmlConfigBuilder）	397
16.4.1	XmlConfigBuilder 的初始化	397
16.4.2	完整的 mybatis-3-config.dtd	399
16.4.3	解析配置文件构建 Configuration 配置	399
16.5	SQL 简介	418
16.6	SQL 映射的构建	419
16.6.1	通过 XML 定义的 SQL Mapper	419
16.6.2	Configuration 类中与 SQL Mapping 相关的类	420
16.6.3	XmlMapperBuilder 是如何工作的	421

16.6.4	映射注解器定义的 SQL Mapper	438
16.6.5	小结	440
第 17 章	执行阶段	441
17.1	关键类	441
17.2	关键接口及默认实现初始化	442
17.2.1	SqlSession 及其关联类的构建过程	442
17.2.2	StatementHandler 语句处理器	446
17.3	DQL 语句是如何执行的	448
17.3.1	查询接口	448
17.3.2	关键时序	449
17.3.3	程序执行查询的入口: DefaultSqlSession#selectList(statement)	450
17.3.4	生成执行语句: getMappedStatement()	450
17.3.5	执行器查询: Executor#query()	451
17.3.6	JDBC 执行语句: SimpleStatementHandler#query()	455
17.3.7	结果集处理: DefaultResultSetHandler#handlerResultSets()	455
17.4	DML 语句是如何执行的	460
17.4.1	操作接口	460
17.4.2	关键时序	460
17.4.3	程序执行更新的入口: DefaultSqlSession#update()	461
17.4.4	执行器执行方法: Executor#update()	461
17.4.5	SQL 语句执行: SimpleStatementHandler#update()	464
17.4.6	结果集主键逻辑: Jdbc3KeyGenerator#processAfter()	464
17.5	小结	466
第 18 章	专题特性解析	467
18.1	动态 SQL 支持	467
18.1.1	XmlScriptBuilder 解析配置	467
18.1.2	NodeHandler 构建 SqlNode 树	468
18.1.3	SqlNode 处理 SQL 语句	471
18.2	MyBatis 的缓存支持	477
18.2.1	本地缓存	478

18.2.2	二级缓存	482
18.3	结果集支持：Object、List、Map 和 Cursor	491
18.4	自定义扩展点及接口	496
第 19 章	作为中间件如何承上启下	498
19.1	MyBatis 与底层的 JDBC	498
19.1.1	java.sql.DataSource	498
19.1.2	java.sql.Connection	499
19.1.3	java.sql.Statement	500
19.1.4	java.sql.ResultSet	502
19.2	MyBatis 的主流集成方式	502
19.2.1	mybatis-spring 简介	502
19.2.2	Spring 对 JDBC 的支持	502
19.2.3	mybatis-spring 与 Spring	504