

Broadview®
www.broadview.com.cn

深入分析

Java Web

技术内幕

许令波◎著



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

深入分析

Java Web

技术内幕

许令波◎著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书围绕 Java Web 相关技术从三方面全面深入地进行阐述。首先介绍前端知识，主要介绍 Java Web 开发中涉及的一些基本知识，包括 Web 请求过程、HTTP 协议、DNS 技术和 CDN 技术。其次深入介绍 Java 技术，包括 I/O 技术、中文编码问题、Javac 编译原理、class 文件结构解析、ClassLoader 工作机制及 JVM 的内存管理等。最后介绍 Java 服务端技术，主要包括 Servlet、Session 与 Cookie、Tomcat 与 Jetty 服务器、Spring 容器、Ibatis 框架和 Velocity 框架等原理介绍。本书不仅介绍这些技术和框架的工作原理，而且结合示例来讲解，通过通俗易懂的文字和丰富生动的配图，让读者充分并深入理解它们的内部工作原理，同时还结合了设计模式来介绍这些技术背后的架构思维。

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

深入分析 Java Web 技术内幕 / 许令波著. —北京：电子工业出版社，2012.9
ISBN 978-7-121-17990-7

I. ①深… II. ①许… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 194519 号

策划编辑：刘 皎

责任编辑：董 英

印 刷：北京丰源印刷厂

装 订：三河市鹏成印业有限公司

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：29 字数：746 千字

印 次：2012 年 9 月第 1 次印刷

印 数：4000 册 定价：69.00 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。

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

服务热线：(010) 88258888。

专家点评

这是一本有关 Java 的书，里面讲述的大量基础知识对前端开发工程师也非常有帮助。比如中文编码章节，作者以一个实践者的身份详细阐述了编码问题的方方面面。总之，这是一本用心的书，是实践者的思考和总结。国内目前很少看到这类书籍，强烈推荐从事 Web 开发工作的人员购买阅读并实践之。

——王保平，开源前端类库 KISSY、SeaJS 作者

作者在淘宝做了很多 Java Web 方面的改造项目，在 Java Web 的相关技术上有深入的掌握，并积累了丰富的经验。在这本书中作者不仅向读者展示了这类大改造项目所需的知识，还展示了 Java Web 更为全景的技术知识体系，值得 Java Web 开发人员阅读。

——林昊，淘宝资深技术专家、China OSGi User Group 总监

从第一次拜读相关内容开始，就可以感觉到作者并不是简简单单地讲述一个技术或者概念，他的分析和讲解十分深入，并且可以很好地聚焦读者的思路，尤其是在 Java Web、Servlet 规范及字符串处理方面，都有很优秀的内容。在众多向 developerWorks 投稿的国内作者中，无论从文章的质量看，还是从内容的选题方向看，作者的文章都可称是上乘之作。同时，他的多篇文章还得到了广大网站读者的好评，其访问量、评分及评论的数量均名列前茅。

——刘达，developerWorks 中国 Java 专区编辑、技术工程师

读者热评

——摘自 developerWorks 上读者对作者文章的评价

相当不错，读完之后颇有顿悟的感觉~~

——lnwazg

看过 how tomcat works，但是有些东西还是没有弄明白，看了你的这篇介绍，虽然不敢说弄明白了，但是至少让我对 tomcat 工作机制及内部实现有了更进一步的了解！

——android007

总结得非常好，以前看了很多遍源代码，也没这样讲的易懂。

——birds

头一次看着这么全的编解码分析，谢谢分享。

——chenxh.

文章相当不错。启动 Servlet 的重要的步骤相当不错。

——RecallYatou

这篇对 Spring 分析的文章太经典了。

——BradyZhu

文章写得很深刻，一直在关注你写的东西。

——61WC_agan_tomsong

很详细，案例和图解释得也很到位，感谢分享。

——lyron

最近才读完了 iBatis 的源代码，也很想写一篇文章的冲动，不过看了此篇，感觉没有必要了。作者的技术水平和写作水平都令人佩服。

——527*****@QQ.com

推荐序

经过 10 多年的发展，Java Web 从开发框架到社区都已经很成熟，在这些成熟的框架、工具的帮助下，开发人员的效率得到了很大的提高，但也造成了在原理性、整体性上的相对欠缺，很多人往往知其然、不知其所以然，特别是在解决一些系统问题的时候，不能很好地举一反三。

举个例子，我看到一些开发人员在使用 Web 框架后，基于约定的方法进行业务的代码实现，但不清楚自己写的代码是如何被调用执行的，如果他很清楚 Servlet 规范，以及看过容器的大致实现过程，对解决问题是很有帮助的。

许令波是我认识的一位很关注原理细节的工程师，同时很乐于分享，会把工作中使用到的技术进行分析并写成文章，分享给大家。他写的这本书中涉及的技术正是他自己在实际工作中遇到的问题的学习过程和解决过程的总结。是总结技术所涉及的知识，更是总结如何分析和解决问题的思路，以及这些技术背后的原理，让你知其所以然。

本书中的内容涉及从 HTTP、Servlet、模板渲染、数据层到容器、JVM 等 Java Web 开发的各个方面，这些问题是许令波在日常工作中经常遇到的，我想也是所有 Java Web 开发人员都会遇到的。本书最大的特点就是让 Java Web 开发人员对整个开发过程所涉及技术能有一个完整的脉络图，从前端浏览器到 Java 技术再到 Java 服务端技术，还介绍了实现这些技术用到的设计模式。不仅详细总结了这些技术的工作原理，而且也结合了很多实际案例来进行阐述，将复杂难懂的技术原理通过时序图和架构图的方式展现出来，更加便于读者理解。可以说掌握了本书的知识，就可以成为一个合格的 Java Web 开发人员。

本书文笔流畅，图表清晰易懂，值得推荐给 Java Web 开发人员作为进阶学习的参考书。

吴泽明

淘宝网产品技术部研究员

杭州，2012.7.23

序

回想起我第一次接触电脑的时间应该是在 10 年前了，记得当时连怎么开电脑都不会，当时感觉电脑真是一个让人着迷的东西，但是那时别说拥有一台电脑，就算是能玩上电脑也算是一件奢侈的事情了。人总是有好奇心的，而我也因为追随着这份好奇和电脑一起走过了将近 10 年的光阴，也是这份好奇让我接触了电脑，认识了电脑，到现在了解了电脑。但是到目前为止我仍然有很多好奇的东西，所以我将一直求解下去。

回想我开始学习编程的时候，在大学期间开始构建自己的第一个网页到第一个网站，其中的复杂程度真是难以想象，要构建一个网页，需要学习当时的“网页三剑客”，页面布局需要学习 Dreamweaver、图片处理需要学习 Fireworks、动画制作需要学习 Flash。有时候为了一个导航栏甚至通宵达旦。还有要自己搭建一个本地服务器，要学习 IIS、Apache 等。当时的我竟然能够一个人完成这一系列的事情，现在想想还真是有点佩服自己。

现在回想一下当时自己的学习过程，真是走了很多弯路，浪费了很多时间。当时的学习就像是在一个陌生的城市找路一样，不知道如何才能到达目的地，只能边走边问别人，这个人告诉你一点，那个人告诉你一点，一点一点往前走。但是虽然在往前走，但走的路并不是最近的，甚至有人指的方向是错的。当时缺少一个总揽全局的地图，所以不能画出一条最优的路。虽然走了很多弯路，但是这种不断自学的过程还是大大地提升了我的学习能力，这种好的自学能力也在我以后的学习工作中起到了关键作用。

IT 行业知识变化很快，需要不断地学习新东西，所以学习知识的能力比掌握知识本身更重要。这也是目前大公司招聘标准中很重要的一条。记得当时我的老人在招聘我进入淘宝时，面试时就问我如何学习一门新技术的问题。当你在学习的过程中碰到了很多难题，然后克服这些难题，很多这种过程积累起来就是你无形的宝贵财富。因为你遇到的问题肯定也是其他人遇到的问题，从发现问题、分析问题再到解决问题的过程远比这个问题本身更有价值。

爱因斯坦说过：“发现问题比解决问题更重要。”对 IT 人员来说，发现 Bug 和重现 Bug 比解决这个 Bug 更有难度。这就好比一个外国人问周总理中国有多少厕所，总理回答说只有两个厕所：男厕所和女厕所。但是，什么人在什么时间、什么地点需要上厕所，考虑这样的情形恐怕需要多少厕所就很难计算了。同样电脑中也只有 0 和 1 两个选择，电脑中的程序也同样如此，每写一行代码就能增加甚至一个数量级的出错的概率。但是我们还是要学习如何避免出现 Bug，这就要求我们能有总理的看问题的思维，将复杂的问题简单化，发现问题背后的本质，找到解决问题的背后的一些通用逻辑，按照这种思路来解决问题可能会让你事半功倍。

如何让学习知识的过程事半功倍，尤其是我们程序员如何做到，从我这么多年的学习过程来说，有一些经验可以分享给大家，这也是我写这本书的初衷，我真正想分享的不是我掌握的知识，而更多的是我如何学习这个知识的过程，或者说我是如何学习这个知识的，以及我对这些知识的一些总结和提炼。

虽然要掌握整个 Web 开发中涉及的所有知识是一件非常困难的事情，尤其是要掌握这些知识的实现原理，不仅知其然还要知其所以然。所以掌握学习它们的方法至关重要。如何快速高效地阅读它们的源码，有很多同学看到我在 developerWorks 发表的文章时来信问我如何阅读各种框架的源码，很多同学都说不知道从哪里入手来看。其实，当你掌握了一些技巧，加上你的一点耐心，这并不是很难的。

本书虽然介绍了很多开源框架，但是始终都在告诉你如何才能更深入和简单地掌握这个框架，告诉你学习的方法，而并不是告诉你这个框架有哪些类、怎么用这些零碎的知识。打个比喻，本书并不是告诉你 $1+1=2$ ， $1+2=3$ ， $2+2=4$ 这个结果，然后你可以根据这个方式得出 $1+1+2=4$ ，你要计算其他数必须根据它给你的公式才能计算，而是告诉你加减乘除的算法规则，然后你就可以根据这个规则自己做运算了。

另外本书为什么要选择介绍 Web 开发中这些技术的实现原理，因为只有你掌握它们的实现原理，你就能够快速解决一些意想不到的问题。例如，当你理解了 ClassLoader 的工作机制后，遇到 ClassNotFoundException 时，你就能快速地判断，到底为什么会报这个错误，可能是哪个地方出错导致的。

另外还有一个很重要的原因是，如果你很想进入淘宝、腾讯、百度这样的大型互联网企业工作，不掌握本书讲到的这些技术的实现原理，是很难通过技术面试的。因为面试官不仅希望你学会用这些技术，还要求你说出个所以然来。所以掌握这些技术的实现原理可以

为你的职业发展提供更好的机会。

本书组织结构

本书从结构上主要分为三个部分：第一部分为基础知识，主要介绍 Java Web 开发中涉及的一些基本知识，如一次 HTTP 请求是什么样的，HTTP 协议本身是如何工作的；第二部分将深入介绍 Java 技术，帮助读者了解 Java 是如何工作的，在会用的基础上进一步理解 Java；第三部分是 Java 服务端技术，主要介绍 Web 服务器的处理流程，包括 Servlet 容器的工作原理和 Web 框架是如何运转的，也就是从 Web 服务器接收到请求到返回请求这个过程中涉及的知识。

除了介绍在 Java Web 开发中用到的框架或系统外，本书还会在介绍这些框架的同时介绍这个框架用到的经典设计模式，因为只有掌握了这些设计模式才能更好地理解这些框架的设计原理。另外用具体的示例来讲解设计模式也能让我们更好地理解设计模式在框架设计中的作用。

目标读者

如果你是学校刚毕业的学生或者刚刚准备学习 Web 开发并且不知道如何入手的人，那么这本书比较适合你；如果你已经工作 1~2 年，已经熟悉了 Java Web 开发的基本流程并且想进一步提高自己，那么这本书更适合你。

如果你已经知道了如何去学习 Java Web 开发技术，正准备入门去实际开发，也就是说你是一个开发新手，那么这本书不太适合你。但是当你知道了如何去开发一个 Web 应用想知道它们如何工作时，欢迎你再回来看看本书，它能帮助你进一步提高。

总的来说，本书适合以下读者群：

- ◎ 对 Web 技术感觉迷茫，不知道如何开始学习，对整个 B/S 工作机制不了解的同学可以参考本书。
- ◎ Java 技术爱好者，想深入学习 Java 技术内部实现细节的人。
- ◎ 有一定开发基础，但是不了解 Web 中一些容器和框架的内部工作原理的人。

- ◎ 大型互联网工程师，对性能优化和分布式数据管理有兴趣，这里介绍了淘宝的一些实践经验。
- ◎ 开源代码爱好者，喜欢研究开发代码的 Coder 可以从本书中找到一些如何分析源码的方法。

本书不会教你如何开发 Web 应用程序，也不会介绍 Struts、Spring、iBatis 等框架如何使用。我想这些框架的使用参考手册图书市场上有一大把，本书没有必要再重复介绍。但是如果你已经掌握如何使用并且不满足只会使用，想知道它们是如何工作的，想打开这些黑盒子，想以后告诉人家这些黑盒子里到底有些什么东西，对每种技术有强烈的好奇心，如果你是这样的人，本书值得你拥有。

本书特点

- ◎ 本书按照通常的学习习惯设计，为你展示从浏览器发起请求到浏览器最终显示出页面整个过程，让你对 Web 开发的整个过程有个总体的理解。
- ◎ 本书虽然讲解的都是比较深入的技术，但是实践示例和比较恰当的比喻将帮你更好地理解。
- ◎ 本书将结合淘宝网中真实使用示例应用程序来讲解技术，让读者有更好的直观认识。

读者讨论

在本书出版后的任何时间有任何问题，你都可以通过 xulingbo0201@163.com 发送邮件给我，或者到 <http://xulingbo.net> 上向我提交你的建议和想法，我会对所有问题给予回复。

致谢

感谢我的父母，在我高考失败后仍然给我机会让我选择自己想做的事，支持我选择了自己喜欢的电脑行业，在家庭并不富裕的情况下给我配置了第一台电脑，让我有机会继续追求自己的梦想，是你们的支持和鼓励让我在做自己一直喜欢做的事。

感谢我的莹，从大学就一直陪伴在我身边，有你在我身边是我一直不断努力的最大动力，在本书写作过程中，你完成初稿的审阅工作，同时也给了我很多鼓励和建议。

感谢电子工业出版社的策划编辑刘皎和责任编辑董英，你们严谨认真的工作态度让我非常敬佩。

感谢吴泽明（范禹）老大为本书写的序，不仅带我进入淘宝，而且也一直帮助我持续进步，感谢王保平、林昊和刘达在繁忙的工作中为我写推荐语。

感谢在本书写作过程中提出宝贵意见的同事们，他们的花名是：小凡、小邪、丹臣、哲别、景升、文通、向飞、凌弃、路奇、济城、大仁、常彬、旭天、韩章、小赌、雁声、索尼、凤豪、柳擎、华黎、空望、嗷嗷、渐飞、普智、胜衣、叔度、文景、撒迦、狄龙、祝幽、单通、承泽等。

感谢 developerWorks 上所有给我提出问题和建议的网友们。

许令波

2012.7

目 录

第 1 章 深入 Web 请求过程	1
1.1 B/S 网络架构概述.....	2
1.2 如何发起一个请求.....	4
1.3 HTTP 协议解析.....	6
1.3.1 查看 HTTP 信息的工具.....	8
1.3.2 浏览器缓存机制.....	9
1.4 DNS 域名解析.....	12
1.4.1 DNS 域名解析过程.....	12
1.4.2 跟踪域名解析过程.....	15
1.4.3 清除缓存的域名.....	18
1.4.4 几种域名解析方式.....	19
1.5 CDN 工作机制.....	20
1.5.1 CDN 架构.....	20
1.5.2 负载均衡.....	21
1.6 总结.....	24
第 2 章 深入分析 Java I/O 的工作机制	25
2.1 Java 的 I/O 类库的基本架构.....	25
2.1.1 基于字节的 I/O 操作接口.....	26
2.1.2 基于字符的 I/O 操作接口.....	27
2.1.3 字节与字符的转化接口.....	28
2.2 磁盘 I/O 工作机制.....	29
2.2.1 几种访问文件的方式.....	29
2.2.2 Java 访问磁盘文件.....	33

2.2.3	Java 序列化技术	34
2.3	网络 I/O 工作机制	36
2.3.1	TCP 状态转化	37
2.3.2	影响网络传输的因素	39
2.3.3	Java Socket 的工作机制	39
2.3.4	建立通信链路	40
2.3.5	数据传输	41
2.4	NIO 的工作方式	41
2.4.1	BIO 带来的挑战	41
2.4.2	NIO 的工作机制	42
2.4.3	Buffer 的工作方式	45
2.4.4	NIO 的数据访问方式	47
2.5	I/O 调优	49
2.5.1	磁盘 I/O 优化	49
2.5.2	TCP 网络参数调优	50
2.5.3	网络 I/O 优化	52
2.6	设计模式解析之适配器模式	56
2.6.1	适配器模式的结构	56
2.6.2	Java I/O 中的适配器模式	57
2.7	设计模式解析之装饰器模式	57
2.7.1	装饰器模式的结构	58
2.7.2	Java I/O 中的装饰器模式	58
2.8	适配器模式与装饰器模式的区别	59
2.9	总结	59
第 3 章	深入分析 Java Web 中的中文编码问题	60
3.1	几种常见的编码格式	60
3.1.1	为什么要编码	60
3.1.2	如何“翻译”	61
3.2	Java 中需要编码的场景	63
3.2.1	I/O 操作中存在的编码	63

3.2.2	内存操作中的编码	65
3.3	Java 中如何编解码	66
3.3.1	按照 ISO-8859-1 编码	68
3.3.2	按照 GB2312 编码	69
3.3.3	按照 GBK 编码	70
3.3.4	按照 UTF-16 编码	70
3.3.5	按照 UTF-8 编码	71
3.3.6	UTF-8 编码代码片段	71
3.3.7	几种编码格式的比较	73
3.4	Java Web 中涉及的编解码	73
3.4.1	URL 的编解码	75
3.4.2	HTTP Header 的编解码	78
3.4.3	POST 表单的编解码	78
3.4.4	HTTP BODY 的编解码	79
3.5	JS 中的编码问题	80
3.5.1	外部引入 JS 文件	80
3.5.2	JS 的 URL 编码	81
3.5.3	其他需要编码的地方	83
3.6	常见问题分析	83
3.6.1	中文变成了看不懂的字符	83
3.6.2	一个汉字变成一个问号	84
3.6.3	一个汉字变成两个问号	84
3.6.4	一种不正常的正确编码	85
3.7	总结	86
第 4 章	Javac 编译原理	87
4.1	Javac 是什么	88
4.2	Javac 编译器的基本结构	88
4.3	Javac 工作原理分析	90
4.3.1	词法分析器	91
4.3.2	语法分析器	98

4.3.3	语义分析器	103
4.3.4	代码生成器	113
4.4	设计模式解析之访问者模式	116
4.4.1	访问者模式的结构	117
4.4.2	Javac 中访问者模式的实现	118
4.5	总结	119
第 5 章	深入 class 文件结构	120
5.1	JVM 指令集简介	120
5.1.1	类相关的指令	122
5.1.2	方法的定义	123
5.1.3	属性的定义	124
5.1.4	其他指令集	125
5.2	class 文件头的表示形式	133
5.3	常量池	137
5.3.1	UTF8 常量类型	140
5.3.2	Fieldref、Methodref 常量类型	141
5.3.3	Class 常量类型	141
5.3.4	NameAndType 常量类型	142
5.4	类信息	142
5.5	Fields 和 Methods 定义	143
5.6	类属性描述	147
5.7	Javap 生成的 class 文件结构	148
5.7.1	LineNumberTable	150
5.7.2	LocalVariableTable	151
5.8	总结	153
第 6 章	深入分析 ClassLoader 工作机制	154
6.1	ClassLoader 类结构分析	155
6.2	ClassLoader 的等级加载机制	156
6.3	如何加载 class 文件	159
6.3.1	加载字节码到内存	159

6.3.2	验证与解析	161
6.3.3	初始化 Class 对象	161
6.4	常见加载类错误分析	161
6.4.1	ClassNotFoundException	161
6.4.2	NoClassDefFoundError	162
6.4.3	UnsatisfiedLinkError	163
6.4.4	ClassCastException	164
6.4.5	ExceptionInInitializerError	165
6.5	常用的 ClassLoader 分析	166
6.6	如何实现自己的 ClassLoader	170
6.6.1	加载自定义路径下的 class 文件	170
6.6.2	加载自定义格式的 class 文件	172
6.7	实现类的热部署	174
6.8	Java 应不应该动态加载类	176
6.9	总结	177
第 7 章	JVM 体系结构与工作方式	178
7.1	JVM 体系结构	178
7.1.1	何谓 JVM	178
7.1.2	JVM 体系结构详解	181
7.2	JVM 工作机制	183
7.2.1	机器如何执行代码	183
7.2.2	JVM 为何选择基于栈的架构	184
7.2.3	执行引擎的架构设计	185
7.2.4	执行引擎的执行过程	186
7.2.5	JVM 方法调用栈	191
7.3	总结	195
第 8 章	JVM 内存管理	196
8.1	物理内存与虚拟内存	197
8.2	内核空间与用户空间	198
8.3	Java 中哪些组件需要使用内存	199

8.3.1	Java 堆	199
8.3.2	线程	199
8.3.3	类和类加载器	200
8.3.4	NIO	200
8.3.5	JNI	201
8.4	JVM 内存结构	201
8.4.1	PC 寄存器	202
8.4.2	Java 栈	202
8.4.3	堆	203
8.4.4	方法区	203
8.4.5	运行时常量池	204
8.4.6	本地方法栈	204
8.5	JVM 内存分配策略	204
8.5.1	通常的内存分配策略	205
8.5.2	Java 中内存分配详解	205
8.6	JVM 内存回收策略	210
8.6.1	静态内存分配和回收	210
8.6.2	动态内存分配和回收	211
8.6.3	如何检测垃圾	211
8.6.4	基于分代的垃圾收集算法	213
8.7	内存问题分析	222
8.7.1	GC 日志分析	222
8.7.2	堆快照文件分析	225
8.7.3	JVM Crash 日志分析	225
8.8	实例 1	231
8.9	实例 2	233
8.10	实例 3	235
8.11	总结	240
第 9 章	Servlet 工作原理析	241
9.1	从 Servlet 容器说起	241