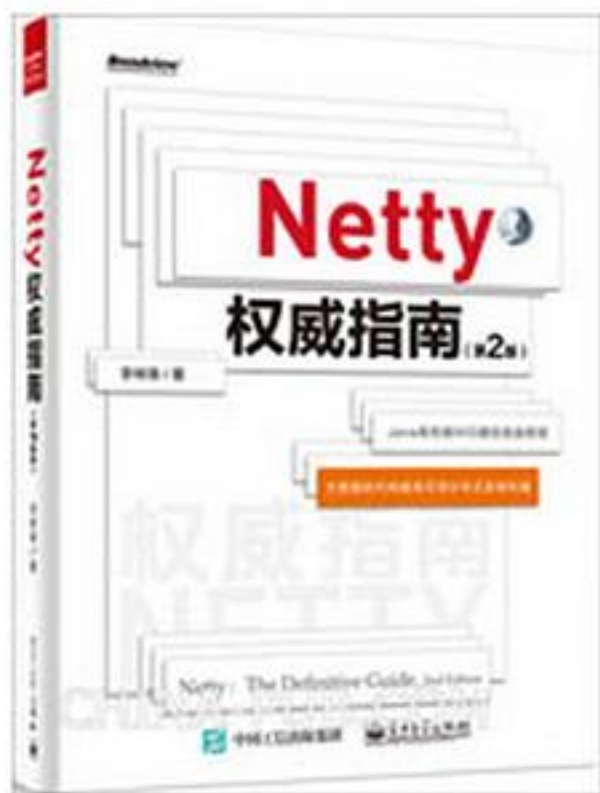


# Netty权威指南（第2版）



[Netty权威指南（第2版）\\_下载链接1](#)

著者:李林锋

出版者:电子工业出版社

出版时间:2015-4-20

装帧:平装

isbn:9787121258015

《Netty 权威指南（第2版）》是异步非阻塞通信领域的经典之作，基于最新版本的Netty 5.0编写，是国内首本深入介绍Netty原理和架构的书籍，也是作者多年实战经验的总结和浓缩。内容不仅包含Java NIO入门知识、Netty

的基础功能开发指导、编解码框架定制等，还包括私有协议栈定制和开发、Netty 核心类库源码分析，以及Netty 的架构剖析。

《Netty 权威指南（第2版）》适合架构师、设计师、软件开发工程师、测试人员以及其他对Java NIO 框架、Netty 感兴趣的相关人士阅读，通过《Netty 权威指南（第2版）》的学习，读者不仅能够掌握Netty 基础功能的使用和开发，更能够掌握Netty 核心类库的原理和使用约束，从而在实际工作中更好地使用Netty。

作者介绍:

李林锋：Netty中国推广者，现华为技术有限公司平台中间件架构与设计部设计师，公司总裁技术创新奖获得者。长期从事高性能通信软件的架构设计和开发工作，有多年在NIO领域的设计、开发和运维经验，精通NIO编程和Netty、Mina等主流NIO框架。目前负责华为软件公司下一代SOA中间件和PaaS平台的架构设计工作。

目录: 基础篇走进Java NIO

第1章Java 的I/O 演进之路.2

1.1 I/O 基础入门.....3

1.1.1 Linux 网络I/O 模型简介.....3

1.1.2 I/O 多路复用技术.....6

1.2 Java 的I/O 演进.....8

1.3 总结..... 10

第2章NIO 入门..... 11

2.1 传统的BIO 编程..... 11

2.1.1 BIO 通信模型图..... 12

2.1.2 同步阻塞式I/O 创建的TimeServer 源码分析..... 13

2.1.3 同步阻塞式I/O 创建的TimeClient 源码分析..... 16

2.2 伪异步I/O 编程..... 18

2.2.1 伪异步I/O 模型图..... 19

2.2.2 伪异步I/O 创建的TimeServer 源码分析..... 19

2.2.3 伪异步I/O 弊端分析..... 21

2.3 NIO 编程..... 24

2.3.1 NIO 类库简介.. 24

2.3.2 NIO 服务端序列图..... 28

2.3.3 NIO 创建的TimeServer 源码分析..... 30

2.3.4 NIO 客户端序列图..... 36

2.3.5 NIO 创建的TimeClient 源码分析..... 39

2.4 AIO 编程..... 45

2.4.1 AIO 创建的TimeServer 源码分析..... 46

2.4.2 AIO 创建的TimeClient 源码分析..... 51

2.4.3 AIO 版本时间服务器运行结果..... 56

2.5 4 种I/O 的对比..... 58

2.5.1 概念澄清..... 58

2.5.2 不同I/O 模型对比..... 59

2.6 选择Netty 的理由..... 60

2.6.1 不选择Java 原生NIO 编程的原因..... 61

2.6.2 为什么选择Netty ..... 62

2.7 总结..... 63

入门篇 Netty NIO 开发指南

第3章Netty 入门应用..... 66

3.1 Netty 开发环境的搭建..... 66

3.1.1 下载Netty 的软件包.....	67
3.1.2 搭建Netty 应用工程.....	67
3.2 Netty 服务端开发.....	68
3.3 Netty 客户端开发.....	73
3.4 运行和调试.....	76
3.4.1 服务端和客户端的运行.....	76
3.4.2 打包和部署.....	77
3.5 总结.....	77
第4章TCP 粘包/拆包问题的解决之道.....	79
4.1 TCP 粘包/拆包.....	79
4.1.1 TCP 粘包/拆包问题说明.....	80
4.1.2 TCP 粘包/拆包发生的原因.....	80
4.1.3 粘包问题的解决策略.....	81
4.2 未考虑TCP 粘包导致功能异常案例.....	82
4.2.1 TimeServer 的改造.....	82
4.2.2 TimeClient 的改造.....	83
4.2.3 运行结果.....	84
4.3 利用LineBasedFrameDecoder 解决TCP 粘包问题.....	85
4.3.1 支持TCP 粘包的TimeServer .....	86
4.3.2 支持TCP 粘包的TimeClient.....	88
4.3.3 运行支持TCP 粘包的时间服务器程序.....	90
4.3.4 LineBasedFrameDecoder 和StringDecoder 的原理分析.....	91
4.4 总结.....	92
第5章分隔符和定长解码器的应用.....	93
5.1 DelimiterBasedFrameDecoder 应用开发.....	94
5.1.1 DelimiterBasedFrameDecoder 服务端开发....	94
5.1.2 DelimiterBasedFrameDecoder 客户端开发....	97
5.1.3 运行DelimiterBasedFrameDecoder 服务端和客户端.....	99
5.2 FixedLengthFrameDecoder 应用开发.....	101
5.2.1 FixedLengthFrameDecoder 服务端开发.....	101
5.2.2 利用telnet 命令行测试EchoServer 服务端.....	103
5.3 总结.....	104
中级篇 Netty 编解码开发指南	
第6章编解码技术.....	106
6.1 Java 序列化的缺点 .....	107
6.1.1 无法跨语言....	107
6.1.2 序列化后的码流太大.....	107
6.1.3 序列化性能太低.....	110
6.2 业界主流的编解码框架.....	113
6.2.1 Google 的Protobuf 介绍....	113
6.2.2 Facebook 的Thrift 介绍....	115
6.2.3 JBoss Marshalling 介绍.....	116
6.3 总结.....	117
第7章MessagePack 编解码.....	118
7.1 MessagePack 介绍... ..	118
7.1.1 MessagePack 多语言支持..	119
7.1.2 MessagePack Java API 介绍.....	119
7.1.3 MessagePack 开发包下载..	120
7.2 MessagePack 编码器和解码器开发.....	120
7.2.1 MessagePack 编码器开发.....	120
7.2.2 MessagePack 解码器开发 .....	121
7.2.3 功能测试.....	121
7.3 粘包/半包支持.....	124
7.4 总结.....	127

第8章Google Protobuf 编解码.....	128
8.1 Protobuf 的入门.....	129
8.1.1 Protobuf 开发环境搭建.....	129
8.1.2 Protobuf 编解码开发.....	131
8.1.3 运行Protobuf 例程.....	133
8.2 Netty 的Protobuf 服务端开发..	133
8.2.1 Protobuf 版本的图书订购服务端开发.....	134
8.2.2 Protobuf 版本的图书订购客户端开发.....	136
8.2.3 Protobuf 版本的图书订购程序功能测试....	139
8.3 Protobuf 的使用注意事项.....	140
8.4 总结.....	142
第9章JBoss Marshalling 编解码.....	143
9.1 Marshalling 开发环境准备.....	143
9.2 Netty 的Marshalling 服务端开发.....	144
9.3 Netty 的Marshalling 客户端开发.....	147
9.4 运行Marshalling 客户端和服务端例程.....	149
9.5 总结.....	150
高级篇 Netty 多协议开发和应用	
第10章HTTP 协议开发应用.....	154
10.1 HTTP 协议介绍.....	155
10.1.1 HTTP 协议的URL .....	155
10.1.2 HTTP 请求消息 (HttpRequest) .....	155
10.1.3 HTTP 响应消息 (HttpResponse) .....	158
10.2 Netty HTTP 服务端入门开发.....	159
10.2.1 HTTP 服务端例程场景描述.....	160
10.2.2 HTTP 服务端开发.....	160
10.2.3 Netty HTTP 文件服务器例程运行结果....	166
10.3 Netty HTTP+XML 协议栈开发.....	170
10.3.1 开发场景介绍.....	171
10.3.2 HTTP+XML 协议栈设计.....	174
10.3.3 高效的XML 绑定框架JiBx .....	175
10.3.4 HTTP+XML 编解码框架开发.....	183
10.3.5 HTTP+XML 协议栈测试.....	199
10.3.6 小结.....	201
10.4 总结.....	202
第11章WebSocket 协议开发.....	203
11.1 HTTP 协议的弊端.....	204
11.2 WebSocket 入门.....	204
11.2.1 WebSocket 背景.....	205
11.2.2 WebSocket 连接建立.....	206
11.2.3 WebSocket 生命周期.....	207
11.2.4 WebSocket 连接关闭.....	208
11.3 Netty WebSocket 协议开发.....	209
11.3.1 WebSocket 服务端功能介绍.....	209
11.3.2 WebSocket 服务端开发....	210
11.3.3 运行WebSocket 服务端...	218
11.4 总结.....	219
第12章私有协议栈开发....	221
12.1 私有协议介绍.....	221
12.2 Netty 协议栈功能设计.....	223
12.2.1 网络拓扑图..	223
12.2.2 协议栈功能描述.....	224
12.2.3 通信模型.....	224
12.2.4 消息定义.....	225

12.2.5 Netty 协议支持的字段类型.....	226
12.2.6 Netty 协议的编解码规范.	227
12.2.7 链路的建立..	229
12.2.8 链路的关闭..	230
12.2.9 可靠性设计..	230
12.2.10 安全性设计	232
12.2.11 可扩展性设计.....	232
12.3 Netty 协议栈开发..	233
12.3.1 数据结构定义.....	233
12.3.2 消息编解码..	237
12.3.3 握手和安全认证.....	241
12.3.4 心跳检测机制.....	245
12.3.5 断连重连.....	248
12.3.6 客户端代码..	249
12.3.7 服务端代码..	251
12.4 运行协议栈.....	252
12.4.1 正常场景.....	252
12.4.2 异常场景：服务端宕机重启.....	253
12.4.3 异常场景：客户端宕机重启.....	256
12.5 总结.....	256
第13 章服务端创建.....	258
13.1 原生NIO 类库的复杂性.....	259
13.2 Netty 服务端创建源码分析....	259
13.2.1 Netty 服务端创建时序图.	260
13.2.2 Netty 服务端创建源码分析.....	263
13.3 客户端接入源码分析.....	272
13.4 总结.....	275
第14 章客户端创建.....	276
14.1 Netty 客户端创建流程分析....	276
14.2.1 Netty 客户端创建时序图.	276
14.2.2 Netty 客户端创建流程分析.....	277
14.2 Netty 客户端创建源码分析....	278
14.2.1 客户端连接辅助类Bootstrap.....	278
14.2.2 客户端连接操作.....	281
14.2.3 异步连接结果通知.....	283
14.2.4 客户端连接超时机制.....	284
14.3 总结.....	286
源码分析篇 Netty 功能介绍和源码分析	
第15 章ByteBuf 和相关辅助类.....	288
15.1 ByteBuf 功能说明.	288
15.1.1 ByteBuf 的工作原理.....	289
15.1.2 ByteBuf 的功能介绍.....	294
15.2 ByteBuf 源码分析.	308
15.2.1 ByteBuf 的主要类继承关系.....	309
15.2.2 AbstractByteBuf 源码分析.....	310
15.2.3 AbstractReferenceCountedByteBuf 源码分析.....	319
15.2.4 UnpooledHeapByteBuf 源码分析.....	321
15.2.5 PooledByteBuf 内存池原理分析.....	326
15.2.6 PooledDirectByteBuf 源码分析.....	329
15.3 ByteBuf 相关的辅助类功能介绍.....	332
15.3.1 ByteBufHolder.....	332
15.3.2 ByteBufAllocator .....	333
15.3.3 CompositeByteBuf .....	334
15.3.4 ByteBufUtil ..	336

15.4 总结.....	337
第16 章Channel 和Unsafe .....	338
16.1 Channel 功能说明.	338
16.1.1 Channel 的工作原理.....	339
16.1.2 Channel 的功能介绍.....	340
16.2 Channel 源码分析.	343
16.2.1 Channel 的主要继承关系类图.....	343
16.2.2 AbstractChannel 源码分析.....	344
16.2.3 AbstractNioChannel 源码分析.....	347
16.2.4 AbstractNioByteChannel 源码分析.....	350
16.2.5 AbstractNioMessageChannel 源码分析.....	353
16.2.6 AbstractNioMessageServerChannel 源码分析.....	354
16.2.7 NioServerSocketChannel 源码分析.....	355
16.2.8 NioSocketChannel 源码分析.....	358
16.3 Unsafe 功能说明...	364
16.4 Unsafe 源码分析...	365
16.4.1 Unsafe 继承关系类图.....	365
16.4.2 AbstractUnsafe 源码分析.	366
16.4.3 AbstractNioUnsafe 源码分析.....	375
16.4.4 NioByteUnsafe 源码分析.	379
16.5 总结.....	387
第17 章ChannelPipeline 和ChannelHandler.....	388
17.1 ChannelPipeline 功能说明.....	389
17.1.1 ChannelPipeline 的事件处理.....	389
17.1.2 自定义拦截器.....	391
17.1.3 构建pipeline	392
17.1.4 ChannelPipeline 的主要特性.....	393
17.2 ChannelPipeline 源码分析.....	393
17.2.1 ChannelPipeline 的类继承关系图.....	393
17.2.2 ChannelPipeline 对ChannelHandler 的管理.....	393
17.2.3 ChannelPipeline 的inbound 事件.....	396
17.2.4 ChannelPipeline 的outbound 事件.....	397
17.3 ChannelHandler 功能说明.....	398
17.3.1 ChannelHandlerAdapter 功能说明.....	399
17.3.2 ByteToMessageDecoder 功能说明.....	399
17.3.3 MessageToMessageDecoder 功能说明.....	400
17.3.4 LengthFieldBasedFrameDecoder 功能说明.....	400
17.3.5 MessageToByteEncoder 功能说明.....	404
17.3.6 MessageToMessageEncoder 功能说明.....	404
17.3.7 LengthFieldPrepender 功能说明.....	405
17.4 ChannelHandler 源码分析.....	406
17.4.1 ChannelHandler 的类继承关系图.....	406
17.4.2 ByteToMessageDecoder 源码分析.....	407
17.4.3 MessageToMessageDecoder 源码分析.....	410
17.4.4 LengthFieldBasedFrameDecoder 源码分析.....	411
17.4.5 MessageToByteEncoder 源码分析.....	415
17.4.6 MessageToMessageEncoder 源码分析.....	416
17.4.7 LengthFieldPrepender 源码分析.....	417
17.5 总结.....	418
第18 章EventLoop 和EventLoopGroup.....	419
18.1 Netty 的线程模型..	419
18.1.1 Reactor 单线程模型.....	420
18.1.2 Reactor 多线程模型.....	421
18.1.3 主从Reactor 多线程模型	422

18.1.4 Netty 的线程模型.....	423
18.1.5 最佳实践.....	424
18.2 NioEventLoop 源码分析.....	425
18.2.1 NioEventLoop 设计原理..	425
18.2.2 NioEventLoop 继承关系类图.....	426
18.2.3 NioEventLoop.....	427
18.3 总结.....	436
第19 章Future 和Promise .....	438
19.1 Future 功能.....	438
19.2 ChannelFuture 源码分析.....	443
19.3 Promise 功能介绍.	445
19.4 Promise 源码分析.	447
19.4.1 Promise 继承关系图.....	447
19.4.2 DefaultPromise .....	447
19.5 总结.....	449
架构和行业应用篇 Netty 高级特性	
第20 章Netty 架构剖析.....	452
20.1 Netty 逻辑架构.....	452
20.1.1 Reactor 通信调度层.....	453
20.1.2 职责链ChannelPipeline ...	453
20.1.3 业务逻辑编排层 (Service ChannelHandler) .....	454
20.2 关键架构质量属性.....	454
20.2.1 高性能.....	454
20.2.2 可靠性.....	457
20.2.3 可定制性.....	460
20.2.4 可扩展性.....	460
20.3 总结.....	460
第21 章Java 多线程编程在Netty 中的应用.....	461
21.1 Java 内存模型与多线程编程..	461
21.1.1 硬件的发展和多任务处理.....	461
21.1.2 Java 内存模型.....	462
21.2 Netty 的并发编程实践.....	464
21.2.1 对共享的可变数据进行正确的同步.....	464
21.2.2 正确使用锁..	465
21.2.3 volatile 的正确使用.....	467
21.2.4 CAS 指令和原子类.....	470
21.2.5 线程安全类的应用.....	472
21.2.6 读写锁的应用.....	476
21.2.7 线程安全性文档说明.....	477
21.2.8 不要依赖线程优先级.....	478
21.3 总结.....	479
第22 章高性能之道.....	480
22.1 RPC 调用性能模型分析.....	480
22.1.1 传统RPC 调用性能差的三宗罪.....	480
22.1.2 I/O 通信性能三原则.....	481
22.2 Netty 高性能之道..	482
22.2.1 异步非阻塞通信.....	482
22.2.2 高效的Reactor 线程模型	482
22.2.3 无锁化的串行设计.....	485
22.2.4 高效的并发编程.....	486
22.2.5 高性能的序列化框架.....	486
22.2.6 零拷贝.....	487
22.2.7 内存池.....	491
22.2.8 灵活的TCP 参数配置能力.....	494

22.3 主流NIO 框架性能对比.....	495
22.4 总结.....	497
第23 章可靠性.....	498
23.1 可靠性需求.....	498
23.1.1 宕机的代价..	498
23.1.2 Netty 可靠性需求.....	499
23.2 Netty 高可靠性设计.....	500
23.2.1 网络通信类故障.....	500
23.2.2 链路的有效性检测.....	507
23.2.3 Reactor 线程的保护.....	510
23.2.4 内存保护.....	513
23.2.5 流量整形.....	516
23.2.6 优雅停机接口.....	519
23.3 优化建议.....	520
23.3.1 发送队列容量上限控制...	520
23.3.2 回推发送失败的消息.....	521
23.4 总结.....	521
第24 章安全性.....	522
24.1 严峻的安全形势....	522
24.1.1 OpenSSL Heart bleed 漏洞.....	522
24.1.2 安全漏洞的代价.....	523
24.1.3 Netty 面临的安全风险....	523
24.2 Netty SSL 安全特性.....	525
24.2.1 SSL 单向认证.....	525
24.2.2 SSL 双向认证.....	532
24.2.3 第三方CA 认证.....	536
24.3 Netty SSL 源码分析.....	538
24.3.1 客户端.....	538
24.3.2 服务端.....	541
24.3.3 消息读取.....	544
24.3.4 消息发送.....	545
24.4 Netty 扩展的安全特性.....	546
24.4.1 IP 地址黑名单机制.....	547
24.4.2 接入认证.....	548
24.4 总结.....	550
第25 章Netty 未来展望....	551
25.1 应用范围.....	551
25.2 技术演进.....	552
25.3 社区活跃度.....	552
25.4 Road Map .....	552
25.5 总结.....	553
附录A Netty 参数配置表....	554
• • • • •	(收起)

[Netty权威指南（第2版）\\_下载链接1](#)

标签

netty



Java

nio

网络编程

java

netty权威指南

编程

计算机

## 评论

看过Netty in Action之后，这本书的价值只在于源码分析中的一部分

-----  
完全不适合对于netty陌生的新手读，先读完netty in action在回头来看这本书，有一些地方可以查漏补缺。缺点就不说了，豆瓣评论中说道的各种缺点仍然适用于第二本。属于在国内科技书籍当中出版质量也比较低的一本书。但是还是有参考意义。。给个两分意思一下。

-----  
入门型，还行吧，比第一版良心一点。

-----  
随便翻译一下文档就敢叫权威指南啊，书中程序还各种错误，真他妈地

-----  
有助于了解和熟悉Netty

---

工具书，随时翻一翻

---

开卷有益

---

不知道为啥豆瓣评分没超过7分，个人一口气看了十章，厚是厚了点，基础和深度都还不错，看起来很快，如果觉得啰嗦的地方可以跳过，但是确实是本好书

---

代码有些老化，还是直接看文档吧

---

大概在16年看过一次，最近重读后感觉作者写的其实还不错，看起来干货不多其实是作者完全提炼后却没展开说的缘故

---

很好的工具书，netty网络上的资源比较少，这本书上有丰富的例子，入门很不错。

---

只适合入门

---

对netty的功能讲解非常全面，而且涉及到对netty的源码讲解，对于想学习源码的同学是一本不错的读物。书中的示例非常详实，当然，也有点过于详实了。给了5星，虽说示例代码过于详实，但还是非常推荐阅读的，因为对于初学者而言，没有什么比更详尽的能够手动运行的示例要更适合的了。

---

深入浅出，代码比较简单，还需要自己多试，看源码

---

目前看到前十章，有些水。

---

前面讲网络相关基础，后面分别对讲粘包拆包，编解码，后面讲接口及其源码，最后讲了一下netty架构方面的设计

---

之前对io模型了解很少1，2章细读。3章东西比较少。4章了解了tcp粘包和拆包。5-25章略读，只看原理，跳过源码解析部分。对tcp协议相关的系统调用(UNIX网络编程相关内容得熟悉)还要多了解一些才能完全理解netty中的原理。

---

内容没说的那么差，代码是贴的有点多了

---

只是大体翻了一下，没有细读。整体上来讲我觉得还是不错的，不像一些评论那么不友好。问题是章节安排详略不当，比如编解码部分，必要性不大，原理讲清楚了一个案例就可以了。还有源码分析部分，近三分之一篇幅，书太厚了。我喜欢In Action系列的风格，要点清楚，不扣细节，注重实战。细节的东西有需要的读者自己去补充。作为技术同行，深知做技术的辛苦，能系统地写一本技术书，必定踩过很多坑，熬过很多不眠夜。支持原创。

---

作为入门还不错，整理的部分内容清晰易懂。不过标题起的太大了。。

---

[Netty权威指南（第2版）\\_下载链接1](#)

## 书评

作者纯粹是为了凑字数的。里面竟然把一大段日志输出分成4-5页放上来。太奇葩了。  
作者纯粹是为了凑字数的。里面竟然把一大段日志输出分成4-5页放上来。太奇葩了。  
作者纯粹是为了凑字数的。里面竟然把一大段日志输出分成4-5页放上来。太奇葩了。

---

1、如前面同志所说，凑字数明显

2、源代码的排版很成问题，看着极度痛苦，字体大小不一，缩进前后不一致，作者号称代码洁癖，但是变量名命名并不规范 3、大部分内容点到为止，更可以说是蜻蜓点水 4、建议去并发网上去看看本书相关的帖子即可，别浪费钱

强烈不推荐看，简直就是浪费时间和金钱。大段大段的代码，大量重复的代码，排版怎一个恶心，作者还自称有代码洁癖，我看简直就是睁眼放硫化氢！  
另外，内容上来说，完全没有什么干货，还有很多无关的东西，有的都是官方文档上面的，而且很多东西都是点到即止。比如说序列化，第...

不如一本cookbook，当api看，太啰嗦，当深入浅出看，很多东西只贴代码，不讲原理，很多关键内容直接跳过，有可学习精彩的部分，但质量不高内容和粗糙的铺陈方式，带来的阅读痛苦。略失望。  
平息下怒气，就当入门书看还是比较有收获，netty核心讲的比较无力，书内容和结构比较...

1.  
这本书给人的感觉就是想全面覆盖Netty的各个方面，出发点是好的，但是确实太过繁琐；  
2.书中随时可见大段大篇幅的代码，这个对于阅读体验来说确实很差，建议在后面的再版中优化一下，保留关键的代码片段，并通过其他方式比如github开放完整的源码来参考； 3.不知道为什么...

虽然书里有一些api罗列、源码分析一些不讨喜的部分，但是整体还是很好的，适合于对Netty有初步的使用能力、希望系统化了解学习的朋友们。  
也许给一星的都是大牛吧，这种级别对他们太low了，不过大牛直接看看英文文档，扫描源码不就好了吗，不需要再看国人写的书了吧。

不太推荐再看这本书了 这本书是以netty5作为基础讲解了  
但是，目前netty官方推荐的版本是4 版本5已不做支持 还是找最新的看吧  
而且书中很多代码片段都木有  
源码讲解部分，只贴代码段，都不晓得是哪个包、哪个类的 看着极其费劲儿  
请原谅我作为一个入门菜鸟的水平 建议还是去官...

ByteBuffer.allocate(1024)是分配了1MB缓冲区，这是怎么算的？难道不是1KB吗？这么明显的错误也能写出来~~~~~

-----  
书很一般，里面讲的东西其实再官网文档里都找得到。前面几章都在介绍Java bio/nio/aio（百度一下一大把），中间介绍了下netty基础（官网文档都能找到的东西），然后就是各种代码例子（其实去下载netty源码包，example下面都有这些类似的例子），最后几章在介绍netty的远吗（还...

-----  
除了有源代码的讲解以外。其他几乎没有任何实质性的东西。  
真的不值得推荐，还不如去看看netty in action。  
其实真的好希望国内有一本详解netty的书出来就好。  
另外，不知道为何书评要这么长的字！！！评论这么长有何用？？垃圾豆瓣。

-----  
可能是因为关于NIO的书不多了,这个本书的名字又这牛,所以买了这本书,但是看完,感觉真是太烂了.希望后来者不要买了,还不如看java thinking中关于NIO的描述.别说是权威指南,连入门都没讲清楚,以后这个人的书大家要谨慎购买.

-----  
豆瓣的第一次书评就送给这本书了-----读过的最烂的书，没有之一。权威指南！  
华为技术大牛！ 伟大！ 伟大！  
评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！ 评论太短了！

-----  
如果是对netty一点都不了解的人，这本书可以作为一本入门的书，内容写的还是比较浅显的。但是很遗憾，作者写书的时候能明显看出赶工的痕迹： 1. 大段的源码编排不合理，导致阅读的时候不便理解，而且前后不连贯。 2. 配图应该用的就是《UNIX网络编程》的电子版截图下来的吧， ...

-----  
排版不行，内容一般，你能不能把源码排版好一点，就算不能你把源代码放出来（网上找了半天找不到啊），  
电子工业出版社的编辑能不能负点责任？前3章有点敷衍，后面还不错，但是不够深入！既然提到了游戏服务器，AMF怎么也要讲一下吧？

-----  
”《Netty 权威指南（第2版）》是异步非阻塞通信领域的经典之作“，这个Title是谁给你封的？  
大段大段的代码。一个功能的代码，本来就占两页，说是为了改进效率在某个函数就增

加了一行代码，就又重新打印了两页代码，并且控制台的输出结果愣是占了三四页。  
你才是大神。 恩， ...

-----

周日看完全书。

刚看到豆瓣评分和评论，认为7分上下算正常，现在这4.1的评分比较扯。日志输出占两页、有些地方的类图模糊、不如看netty in

action，这些我都同意，但这样就否定全书值得商榷。既然有这么多骂的，那就不再重复，这里谈谈亮点，让看评论的人多点不同意见的参考： ...

-----

人无完人，书也是一样，为什么大家要纠结它的缺点，忽略它的优点呢？在我初次使用netty的时候，我完全找不到方向，在看了这本书以后，收获很多，弥补了我的基础，又让知识上一个台阶，本书兼顾了初级入门，和这个框架的使用方法，又带了分析，我很喜欢这书

-----

[Netty权威指南（第2版） 下载链接1](#)