

扩展：进阶学习 Netty 的方向与资料

最后一小节，给大家推荐一些比较靠谱的进阶学习 Netty 的资料

1. 官网与 github

毋庸置疑，通常情况下，官网一直是学习一门技术最好的资料

<https://netty.io/wiki/user-guide-for-4.x.html> 这里是官方给出的 4.x 版本的 Netty 的一个学习指引，大家可以温习一遍。

<https://netty.io/wiki/new-and-noteworthy-in-4.1.html> 4.1x 版本的新的特性，大家也可以感受一下 Netty 的版本变化。

另外，大家也可以了解一下目前有哪些开源项目使用了 Netty：

<https://Netty.io/wiki/related-projects.html>

关于 Netty 详细的 demo，也可以在官网找到，比如，你想使用 Netty 来实现一个 Http 服务器，或者实现一个 Websocket 服务器，或者实现 redis 协议等等，都可以在 [官方提供的 demo](https://github.com/netty/netty/tree/4.1/example/src/main/) (<https://github.com/netty/netty/tree/4.1/example/src/main/>) 中找到。

最后 Netty 的作者之一 [normanmaurer](https://github.com/normanmaurer) (<https://github.com/normanmaurer>) 大大也亲自操刀写了一本 Netty 入门的书，叫做《Netty 实战》，国内也有译本，讲的比较全面，大家也可以当做参考书来读一读。再歪一句，[normanmaurer](https://github.com/normanmaurer) (<https://github.com/normanmaurer>) 大大

也是一个非常热心并且幽默风趣的人，在 Netty 官方邮件列表非常活跃，github 上提问题也经常秒回，下面，放出一张他的 ppt，大家应该可以感受到他的幽默 ^ ^

Want to know more?



Buy my book [Netty in Action](#) and make me **RICH**.



<http://www.manning.com/maurer>

\$ KA-CHING \$

2. 源码解析博客

要精通一门技术，除了会熟练使用之外，撸源码也是一条必经之路。

笔者在过去一段时间陆续写过很多篇关于 Netty 源码解析的文章，这里我也做下归类

关于 Netty 服务端启动的流程可以参考

[Netty 源码分析之服务端启动全解析](#)

<https://juejin.im/post/5bc92271e51d450ea4024c75>

关于服务端是如何处理一条新的连接的，可以参考

[Netty 源码分析之新连接接入全解析](#)

<https://juejin.im/post/5bcd01ccf265da0aa664f99b>

关于 Netty 里面 NIO 到底干了啥事，为啥可以做到一个 NIO 线程就可以处理上万连接，异步机制又是如何实现的，可以参考以下三篇文章

1. [Netty 源码分析之揭开 reactor 线程的面纱 \(一\)](https://juejin.im/post/5bce54826fb9a05d1f2247a0)
(<https://juejin.im/post/5bce54826fb9a05d1f2247a0>)
2. [Netty 源码分析之揭开 reactor 线程的面纱 \(二\)](https://juejin.im/post/5bcfa22fe51d4534763cc5bd)
(<https://juejin.im/post/5bcfa22fe51d4534763cc5bd>)
3. [Netty 源码分析之揭开 reactor 线程的面纱 \(三\)](https://juejin.im/post/5bd1120cf265da0af7755b6c)
(<https://juejin.im/post/5bd1120cf265da0af7755b6c>)

关于事件传播机制 Netty 又是如何实现的，为什么 inBound 和 outBound 的传播顺序与添加顺序的对应规则不同，Netty 又是如何来区分 inBound 和 outBound 的，Netty 的 pipeline 默认又有哪两个 handler，他们的作用分别是什么，一切尽在以下两篇文章

1. [Netty 源码分析之 pipeline \(一\)](https://juejin.im/post/5bd26334e51d457a496de685)
(<https://juejin.im/post/5bd26334e51d457a496de685>)
2. [Netty 源码分析之 pipeline \(二\)](https://juejin.im/post/5bd79c39f265da0aa81c47cf)
(<https://juejin.im/post/5bd79c39f265da0aa81c47cf>)

Netty 中拆包器的原理是什么？可以参考 [Netty 源码分析之拆包器的奥秘](https://juejin.im/post/5bda41646fb9a0225703923d) (<https://juejin.im/post/5bda41646fb9a0225703923d>)

基于长度域的拆包器又是如何来实现的，可以参考 [Netty 源码分析之 LengthFieldBasedFrameDecoder](https://juejin.im/post/5bedf8636fb9a049b07cec90)
(<https://juejin.im/post/5bedf8636fb9a049b07cec90>)

最后，我们在本小册接触最频繁的 writeAndFlush() 方法，它又是如何实现异步化的，可以参考 [Netty 源码分析之 writeAndFlush 全解析](https://www.jianshu.com/p/feaeaab2ce56) (<https://www.jianshu.com/p/feaeaab2ce56>)

关于源码解析的文章差不多就这么多，另外，如果你对我之前遇到的线上与 Netty 相关的问题排查，以及一些调优相关的经验和实践感兴趣的话，也可以读一读下面几篇文章

1. [Netty 堆外内存泄露排查盛宴](https://juejin.im/post/5b8dbbd4518825430810d760)
(<https://juejin.im/post/5b8dbbd4518825430810d760>)
2. [海量连接服务端 jvm 参数调优杂记](https://www.jianshu.com/p/051d566e110d)
(<https://www.jianshu.com/p/051d566e110d>)
3. [一次 Netty "引发的" 诡异 old gc 问题排查过程](https://www.jianshu.com/p/702ef10102e4)
(<https://www.jianshu.com/p/702ef10102e4>)

3. 源码解析视频

如果你觉得文章看不下去，更偏向视频的话，那么我这里也有之前录的一个源码解析视频，手把手带你撸源码：[Java读源码之 Netty 深入剖析](https://coding.imooc.com/class/evaluation/230.html)

(<https://coding.imooc.com/class/evaluation/230.html>)

4. 微信公众号

最后，也是我个人的微信公众号，最近比较火的三篇小闪对话系列也许能够以一种轻松幽默地方式帮你了解 IM 集群，负载均衡相关的知识。



