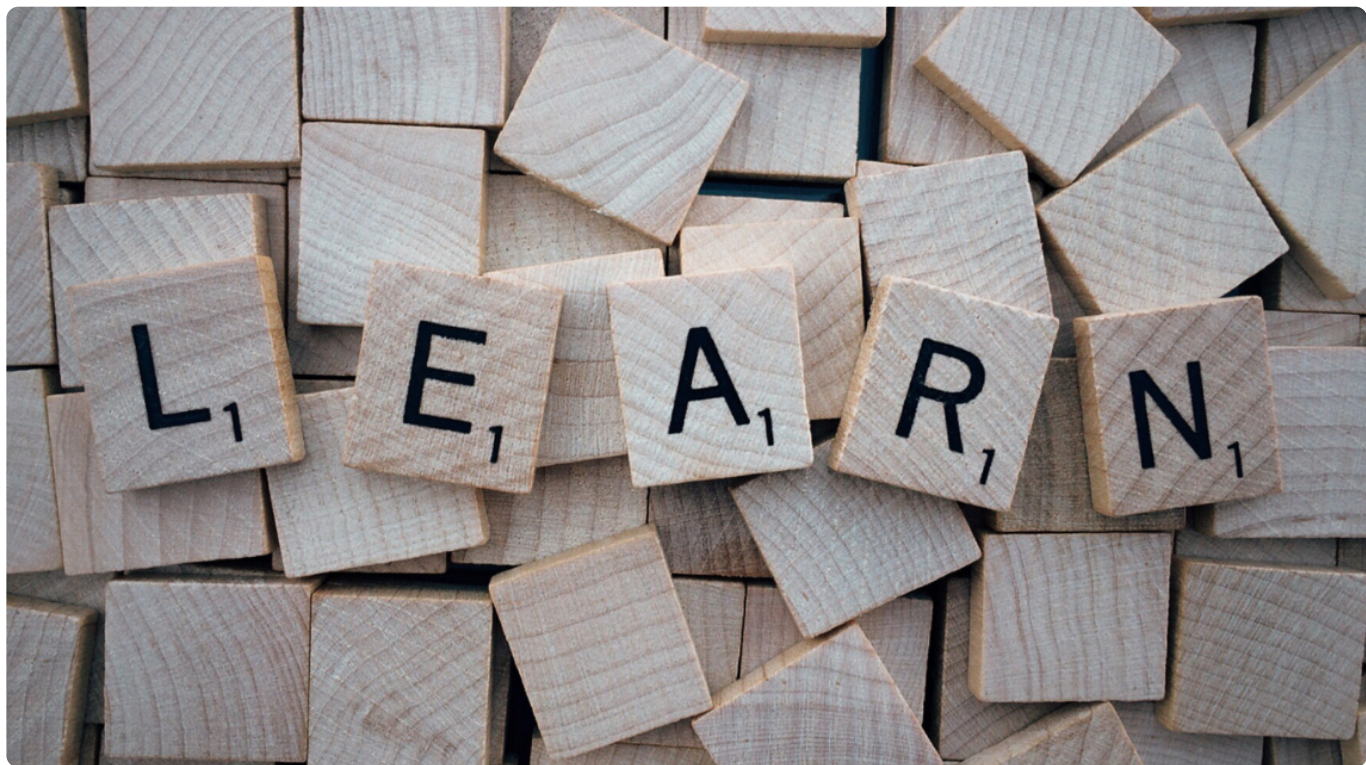


特别放送 | 如何持续保持对学习的兴趣？

2019-07-27 李号双

深入拆解Tomcat & Jetty

[进入课程 >](#)



讲述：李号双

时长 07:01 大小 6.44M



你好，我是李号双。今天我们抛开技术本身的内容，来聊聊专栏或者一门新技术的学习方法，我也分享一下自己是如何啃下 Tomcat 和 Jetty 源码的。

专栏如今已经更新完了五个模块，我们学习了 Tomcat 和 Jetty 的整体架构、连接器、容器和通用组件，这些内容可以说是 Tomcat 和 Jetty 的设计核心。在日常工作的使用中，我们使用到了 Tomcat 和 Jetty 提供的功能，我希望通过学习专栏，还能帮你了解这些功能是如何实现的，以及 Tomcat 和 Jetty 在设计时都考虑了哪些地方。

所以在学习专栏时，你不妨思考这样一个问题，假如让你来设计并实现一个 Web 容器，你会怎么做呢？如何合理设计顶层模块？如何考虑方方面面的需求，比如最基本的功能需求是加载和运行 Web 程序，最重要的非功能需求是高性能、高并发。你可以顺着这两条线先思考下你会怎么做，然后再回过头来看看 Tomcat 和 Jetty 是如何做到的。这样的学习方法其

实就在有意识地训练自己独立设计一个系统的能力，不管是对于学习这个专栏还是其他技术，带着问题再去学习都会有所帮助。

说完关于专栏的学习方法，下面**我必须要鼓励一下坚持学习到现在的你**。专栏从第三模块开始，开始讲解连接器、容器和通用组件的设计和原理，有些内容可能比较偏向底层，确实难度比较大，如果对底层源码不熟悉或者不感兴趣，学习起来会有些痛苦。但是，我之所以设计了这部分内容，就是希望能够揭开 Tomcat 和 Jetty 的内部细节，因为任何一个优秀的中间件之所以可以让用户使用比较容易，其内部一定都是很复杂的。这也从侧面传递出一个信号：美好的东西都是有代价的，需要也值得我们去付出时间和精力。

我和你一样我们都身处 IT 行业，这个行业技术更新迭代非常快，因此我们需要以一个开放的心态持续学习。而学习恰恰又是一个反人性的过程，甚至是比较痛苦的，尤其是有些技术框架本身比较庞大，设计得非常复杂，我们在学习初期很容易遇到“挫折感”，一些技术点怎么想也想不明白，往往也会有放弃的想法。我同样经历过这个过程，**我的经验是找到适合自己的学习方法非常重要，同样关键的是要保持学习的兴趣和动力。**

举个我学习 Spring 框架的例子，记得当时我在接触 Spring 框架的时候，一开始就钻进一个模块开始啃起了源代码。由于 Spring 框架本身比较庞杂，分很多模块，当时给我最直观的感受就是看不懂，我不明白代码为什么要这么写，为什么设计得这么“绕”。这里面的问题是，首先**我还没弄清楚森林长什么样子，就盯着树叶看**，很可能是盲人摸象，看不到全貌和整体的设计思路。第二个问题是我还没学会用 Spring，就开始研究它是如何设计的，结果可想而知，也遇到了挫折。后来我逐渐总结出一些学习新技术的小经验：在学习一门技术的时候，一定要先看清它的全貌，我推荐先看官方文档，看看都有哪些模块、整体上是如何设计的。接着我们先不要直接看源码，而是要动手跑一跑官网上的例子，或者用这个框架实现一个小系统，关键是要学会怎么使用。只有在这个基础上，才能深入到特定模块，去研究设计思路，或者深入到某一模块源码之中。这样在学习的过程中，按照一定的顺序一步步来，就能够即时获得**成就感**，有了成就感你才会更加专注，才会愿意花更多时间和精力去深入研究。因此要保持学习的兴趣，我觉得有两个方面比较重要：

第一个是我们需要**带着明确的目标去学习**。比如某些知识点是面试的热点，那学习目标就是彻底理解和掌握它，当被问到相关问题时，你的回答能够使得面试官对你刮目相看，有时候往往凭着某一个亮点就能影响最后的录用结果。再比如你想掌握一门新技术来解决工作上的问题，那你的学习目标应该是不但要掌握清楚原理，还要能真正的将新技术合理运用到实际工作中，解决实际问题，产生实际效果。我们学习了 Tomcat 和 Jetty 的责任链模式，是不

是在实际项目中的一些场景下就可以用到这种设计呢？再比如学习了调优方法，是不是可以在生产环境里解决性能问题呢？总之技术需要变现才有学习动力。

第二个是**一定要动手实践**。只有动手实践才会让我们对技术有最直观的感受。有时候我们听别人讲经验和理论，感觉似乎懂了，但是过一段时间便又忘记了。如果我们动手实践了，特别是在这个过程中碰到了一些问题，通过网上查找资料，或者跟同事讨论解决了问题，这便是你积累的宝贵经验，想忘记都难。另外适当的动手实践能够树立起信心，培养起兴趣，这跟玩游戏上瘾有点类似，通过打怪升级，一点点积累起成就感。比如学习了 Tomcat 的线程池实现，我们就可以自己写一个定制版的线程池；学习了 Tomcat 的类加载器，我们也可以自己动手写一个类加载器。

专栏更新到现在，内容最难的部分已经结束，在后面的实战调优模块，我在设计内容时都安排了实战环节。毕竟调优本身就是一个很贴近实际场景的话题，应该基于特定场景，去解决某个性能问题，而不是为了调优而调优。所以这部分内容也更贴近实际工作场景，你可以尝试用我前面讲的方法，带着问题学习后面的专栏。

调优的过程中需要一些知识储备，比如我们需要掌握操作系统、JVM 以及网络通信的原理，这些原理在专栏前面的文章也讲到过。虽然涉及很多原理也很复杂，并不是说要面面俱到，我们也不太容易深入到每个细节，所以最关键的是要弄懂相关参数的含义，比如 JVM 内存的参数、GC 的参数、Linux 内核的相关参数等。

除此之外，调优的过程还需要借助大量的工具，包括性能监控工具、日志分析工具、网络抓包工具和流量压测工具等，熟练使用这些工具也是每一个后端程序员必须掌握的看家本领，因此在实战环节，我也设计了一些场景来带你熟悉这些工具。

说了那么多，就是希望你保持对学习的热情，树立明确的目标，再加上亲自动手实践。专栏学习到现在这个阶段，是时候开始动手实践了，希望你每天都能积累一点，每天都能有所进步。

最后欢迎你在留言区分享一下你学习一门新技术的方法和心得，与我和其他同学一起讨论。

深入拆解 Tomcat & Jetty

从源码角度深度探索 Java 中间件

李号双

eBay 技术主管



新版升级：点击「👤 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 33 | Cluster组件：Tomcat的集群通信原理

精选留言 (1)

写留言



QQ怪

2019-07-27

跟着老师一路走来，的确感觉一个新的技术起先要懂个大体的框架，知道其是干什么的，为了解决什么问题的，知根知底是很必要的；然后就是要会用，会用之后对看源码和懂源码帮助很大，深有感悟



3