

## 01 | Web容器学习路径

2019-05-13 李号双

深入拆解Tomcat & Jetty

[进入课程 >](#)



讲述：李号双

时长 09:58 大小 9.14M



你好，我是李号双。在开篇词里我提到要成长为一名高级程序员或者架构师，我们需要提高自己知识的广度和深度。你可以先突破深度，再以点带面拓展广度，因此我建议通过深入学习一些优秀的开源系统来达到突破深度的目的。

我会跟你一起在这个专栏里深入学习 Web 容器 Tomcat 和 Jetty，而作为专栏更新的第 1 篇文章，我想和你谈谈什么是 Web 容器，以及怎么学习 Web 容器。根据我的经验，在学习一门技术之前，想一想这两个问题，往往可以达到事半功倍的效果。

### Web 容器是什么？

让我们先来简单回顾一下 Web 技术的发展历史，可以帮助你理解 Web 容器的由来。

早期的 Web 应用主要用于浏览新闻等静态页面，HTTP 服务器（比如 Apache、Nginx）向浏览器返回静态 HTML，浏览器负责解析 HTML，将结果呈现给用户。

随着互联网的发展，我们已经不满足于仅仅浏览静态页面，还希望通过一些交互操作，来获取动态结果，因此也就需要一些扩展机制能够让 HTTP 服务器调用服务端程序。

于是 Sun 公司推出了 Servlet 技术。你可以把 Servlet 简单理解为运行在服务端的 Java 小程序，但是 Servlet 没有 main 方法，不能独立运行，因此必须把它部署到 Servlet 容器中，由容器来实例化并调用 Servlet。

而 Tomcat 和 Jetty 就是一个 Servlet 容器。为了方便使用，它们也具有 HTTP 服务器的功能，因此 **Tomcat 或者 Jetty 就是一个“HTTP 服务器 + Servlet 容器”，我们也叫它们 Web 容器。**

其他应用服务器比如 JBoss 和 WebLogic，它们不仅仅有 Servlet 容器的功能，也包含 EJB 容器，是完整的 Java EE 应用服务器。从这个角度看，Tomcat 和 Jetty 算是一个轻量级的应用服务器。

在微服务架构日渐流行的今天，开发人员更喜欢稳定的、轻量级的应用服务器，并且应用程序用内嵌的方式来运行 Servlet 容器也逐渐流行起来。之所以选择轻量级，是因为在微服务架构下，我们把一个大而全的单体应用，拆分成一个个功能单一的微服务，在这个过程中，服务的数量必然要增加，但为了减少资源的消耗，并且降低部署的成本，我们希望运行服务的 Web 容器也是轻量级的，Web 容器本身应该消耗较少的内存和 CPU 资源，并且由应用本身来启动一个嵌入式的 Web 容器，而不是通过 Web 容器来部署和启动应用，这样可以降低应用部署的复杂度。

因此轻量级的 Tomcat 和 Jetty 就是一个很好的选择，并且 Tomcat 它本身也是 Spring Boot 默认的嵌入式 Servlet 容器。最新版本 Tomcat 和 Jetty 都支持 Servlet 4.0 规范。

读到这里，我想你应该对 Web 容器有了基本的认识，可以结合平时工作再去细细体会一下。如果你对 HTTP 协议和 Servlet 依然是一头雾水，不用担心，在预习模块中我还会和你聊聊你应该掌握的 HTTP 协议和 Servlet 的相关知识，帮你打好学习的基础。

## **Web 容器该怎么学？**

Java Web 技术发展日新月异，各种框架也是百花齐放。在从事 Java Web 开发相关的工作时，面对这些眼花缭乱的技术时你是否会感到一丝迷茫？可能有些初学者不知道从哪里开始，我身边还有些已经进入了这个行业，并且有了一定 Java 基础的人，对于系统设计的体会可能还不够深刻，编程的时候还停留在完成功能的层次。这样不仅业务上难有突破，对于个人成长也很不利。

为了打破这个瓶颈，就需要我们在深度上多下功夫，找准一个点，深挖下去，彻底理解它的原理和设计精髓。并且在深入学习 Tomcat 和 Jetty 这样的 Web 容器之前，你还需要掌握一定的基础知识，这样才能达到事半功倍的效果。

下面我列举一些在学习 Web 容器之前需要掌握的关键点，我建议你在学习专栏的同时，再去复习一下这些基础知识。你可以把这些基础知识当作成为架构师的必经之路，在专栏以外也要花时间深入进去。当然为了让你更好地理解专栏每期所讲的内容，重点的基础知识我也会在文章里帮你再梳理一遍。

## 操作系统基础

Java 语言其实是对操作系统 API 的封装，上层应用包括 Web 容器都是通过操作系统来工作的，因此掌握相关的操作系统原理是我们深刻理解 Web 容器的基础。

对于 Web 容器来说，操作系统方面你应该掌握它的工作原理，比如什么是进程、什么是内核、什么是内核空间和用户空间、进程间通信的方式、进程和线程的区别、线程同步的方式、什么是虚拟内存、内存分配的过程、什么是 I/O、什么是 I/O 模型、阻塞与非阻塞的区别、同步与异步的区别、网络通信的原理、OSI 七层网络模型以及 TCP/IP、UDP 和 HTTP 协议。

总之一句话，基础扎实了，你学什么都快。关于操作系统的学习，我推荐你读一读《UNIX 环境高级编程》这本经典书籍。

## Java 语言基础

Java 的基础知识包括 Java 基本语法、面向对象设计的概念（封装、继承、多态、接口、抽象类等）、Java 集合的使用、Java I/O 体系、异常处理、基本的多线程并发编程（包括线程同步、原子类、线程池、并发容器的使用和原理）、Java 网络编程（I/O 模型 BIO、NIO、AIO 的原理和相应的 Java API）、Java 注解以及 Java 反射的原理等。

此外你还需要了解一些 JVM 的基本知识，比如 JVM 的类加载机制、JVM 内存模型、JVM 内存空间分布、JVM 内存和本地内存的区别以及 JVM GC 的原理等。

这方面我推荐的经典书籍有[《Java 核心技术》](#)、[《Java 编程思想》](#)、[《Java 并发编程实战》](#)和[《深入理解 Java 虚拟机：JVM 高级特性与最佳实践》](#)等。

## Java Web 开发基础

具备了一定的操作系统和 Java 基础，接下来就可以开始学习 Java Web 开发，你可以开始学习一些通用的设计原则和设计模式。这个阶段的核心任务就是了解 Web 的工作原理，**同时提高你的设计能力**，注重代码的质量。我的建议是可以从学习 Servlet 和 Servlet 容器开始。我见过不少同学跳过这个阶段直接学 Web 框架，这样做的话结果会事倍功半。

为什么这么说呢？Web 框架的本质是，开发者在使用某种语言编写 Web 应用时，总结出的一些经验和设计思路。很多 Web 框架都是从实际的 Web 项目抽取出来的，其目的是用于简化 Web 应用程序开发。

我以 Spring 框架为例，给你讲讲 Web 框架是怎么产生的。Web 应用程序的开发主要是完成两方面的工作。

设计并实现类，包括定义类与类之间的关系，以及实现类的方法，方法对数据的操作就是具体的业务逻辑。

类设计好之后，需要创建这些类的实例并根据类与类的关系把它们组装在一起，这样类的实例才能一起协作完成业务功能。

就好比制造一辆汽车，汽车是由零件组装而成的。第一步是画出各种零件的图纸，以及定义零件之间的接口。第二步把把图纸交给工厂去生产零件并组装在一起。因此对于 Web 应用开发来说，第一步工作是具体业务逻辑的实现，每个应用都不一样。而第二步工作，相对来说比较通用和标准化，工厂拿到零件的图纸，就知道怎么生产零件并按照零件之间的接口把它们组装起来，因此这个工作就被抽取出来交给 Spring 框架来做。

Spring 又是用容器来完成这个工作的，容器负责创建、组装和销毁这些类的实例，而应用只需要通过配置文件或者注解来告诉 Spring 类与类之间的关系。但是容器的概念不是 Spring 发明的，最开始来源于 Servlet 容器，并且 Servlet 容器也是通过配置文件来加载



Servlet 的。你会发现它们的“元神”是相似的，在 Web 应用的开发中，有一些本质的东西是不变的，而很多“元神”就藏在“老祖宗”那里，藏在 Servlet 容器的设计里。

Spring 框架就是对 Servlet 的封装，Spring 应用本身就是一个 Servlet，而 Servlet 容器是管理和运行 Servlet 的，因此我们需要先理解 Servlet 和 Servlet 容器是怎样工作的，才能更好地理解 Spring。

## 本期精华

今天我谈了什么是 Web 容器，以及该如何学习 Web 容器。在深入学习之前，你需要掌握一些操作系统、Java 和 Web 的基础知识。我希望你在学习专栏的过程中多温习一下这些基础知识，有扎实的基础，再结合专栏深入学习 Web 容器就比较容易了。

等你深刻理解了 Web 容器的工作原理和设计精髓以后，你就可以把学到的知识扩展到其他领域，你会发现它们的本质都是相通的，这个时候你可以站在更高的角度来学习和审视各种 Web 框架。虽然 Web 框架的更新比较快，但是抓住了框架的本质，在学习的过程中，往往会更得心应手。

不知道你有没有遇到过这样的场景，当你在看一个框架的技术细节时，会突然恍然大悟：对啊，就是应该这么设计！如果你有这种感觉，说明你的知识储备起到了作用，你对框架的运用也会更加自如。

## 课后思考

请你分享一下你对 Web 容器的理解，或者你在学习、使用 Web 容器时遇到了哪些问题？

不知道今天的内容你消化得如何？如果还有疑问，请大胆的在留言区提问，也欢迎你把你的课后思考和心得记录下来，与我和其他同学一起讨论。如果你觉得今天有所收获，欢迎你把它分享给你的朋友。

---

# 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 开篇词 | Java程序员如何快速成长？

下一篇 02 | HTTP协议必知必会

## 精选留言 (31)

写留言



落幕

2019-05-14

12

和一群优秀的人共同进步，这种感觉很好。

展开



蔡伶

2019-05-15

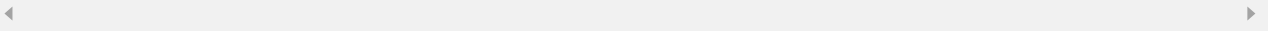
9

打卡

先说下听完老师课程的感受：经典不会随着时间而消逝。java和servlet规范已经发布20多年、操作系统和网络协议以及html更是经过了几十年的洗礼，现在依然是业内最核心的技术基础，毫不动摇。...

展开 ▾

作者回复: 说的很好👍



G

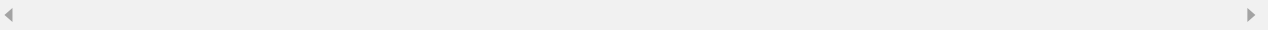
2019-05-13

👍 6

你说的所有spring. 都应该说springMVC

展开 ▾

作者回复: 嗯嗯, SpringMVC是Spring的子集, 我就说成Spring了~



今夜秋风和

2019-05-14

👍 2

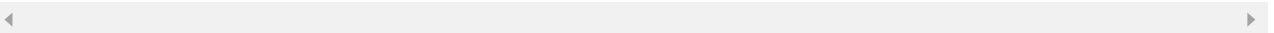
应用程序的上下文, 这个概念总是感觉理解不透彻

展开 ▾

作者回复: 简单可以这里理解:

我们把Spring的IOC容器理解为一个工厂, 这个工厂负责创建组装你的Bean。

但是我们怎么向IOC容器中放入Bean呢? 可能通过配置文件或者注解或者其他方式, 于是容器除了做创建、组装Bean的工作, 还需要去做解析配置文件或者注解的工作, 于是把容器换个说法, 叫应用上下文。



无梦何以展...

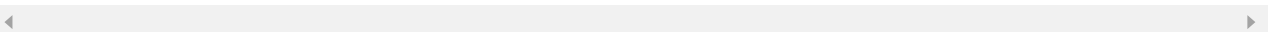
2019-05-14

👍 2

期待不断更新, 希望老师能够讲清楚, 让大家都理解

展开 ▾

作者回复: 后面的内容基本是先回顾基础知识, 再谈Tomcat&Jetty如何在真实场景下运用这些基础知识, 这样理解起来比较容易。





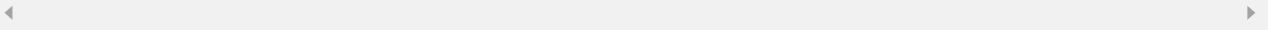
yy\_java

2019-05-18

👍 1

请问老师，操作系统基础 除了您推荐的那本书以外还有其他薄点的书籍推荐吗？

作者回复: 😊，我理解这本书是有点厚，但还是建议读经典的比较好，你不需要一口气读完，专栏讲到了某个点再去看看相关的部分~



Geek\_f318c...

2019-05-16

👍 1

打卡第一天。我们通过深入的学习Tomcat&Jetty来完成自我提升。今天主要回答了两个问题，第一，Web容器是什么？简而言之，Web容器就是Servlet容器加HTTP服务器，它使得我们请求的页面可以与客户端动态的交互。还介绍了Tomcat，Jetty与Weblogic，Jboss的区别。后者不但包含了HTTP服务器和Servlet容器，它还包含了EJB，是完整的JavaEE应用服务器。第二，回答了如何学习Web容器。我总结为，打好基础，发散思维...  
展开 ▾



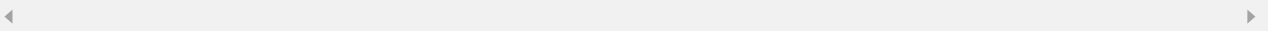
Monday

2019-05-16

👍 1

操作系统还是我的痛，但也不是一两天就能补充得了的。只好边学本专栏边学操作系统了。希望不要因为操作系统的缘故拖了学习本专栏的后腿 😊 😊

作者回复: 不会的，咱们这边都有具体应用场景，根据实际运用场景开学理解起来会容易些



Jaswine

2019-05-15

👍 1

带着问题学知识，学习的时候问自己几个为什么，为什么这么设计？为什么不那么设计？在自己解答这些为什么的时候，最后发现都是计算机基础知识决定的，操作系统，网络，数据结构和算法

展开 ▾

作者回复: 嗯嗯，除了学，还要思考和总结







贤蛋蛋

2019-05-15

👍 1

请问为什么说http是超文本传输协议，文本两字的含义是什么？http2.0所说的二进制帧，为什么说是二进制，和1.1格式上的本质区别是什么？再往下一层到TCP能否都看成二进制帧？

展开 ▾

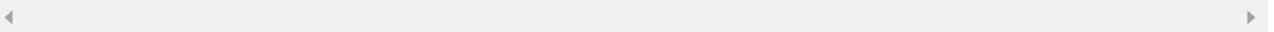
作者回复: 文本可以理解为只有文字信息的文档，超文本是带有超链接的文档，可以链接到另一个文档，或一张图...

HTTP1.1是文本协议，HTTP2.0是二进制协议。

文本协议的协议数据是由ASCII字符组成的，比如文章里的HTTP请求的例子：请求行、请求头和请求体，我们一眼就看出什么意思。这是因为协议里的每个Byte都是用ASCII字符来解释的。

二进制协议的每个Byte完全由协议本身来定义，比如一个Byte有8个Bit，这8个Bit可能有不同的意思（比如代表长度或者其他标志位），不一定代表一个ASCII字符。

TCP是二进制协议。



凌霄

2019-05-14

👍 1

遇到过一个偶发的tomcat8问题，请求到tomcat后，nio长连接，到了20秒后超时后才自动断开连接，返回结果内容正常，抓包发现和正常的比少了最后的回车换行。

展开 ▾

作者回复: 你的应用程序设置的响应长度Content-Length与实际响应数据长度不符，可能长了那么一丢丢，这样Tomcat一直在等你的数据呢。



JackJin

2019-05-13

👍 1

还停留在使用容器的阶段，并不清楚其原理，例如：一个请求到一个响应返回，其涉及到的设计模式，以及为什么这样做，这样做的好处是什么；我能在容器的基础上做一些自定义的扩展吗？希望在专栏收获到这些。

展开 ▾

作者回复: 对的，这个专栏会学习Tomcat&Jetty为什么设计成这样，设计者是怎么考虑问题的；

Tomcat和Jetty做为中间件，可扩展性非常强，你可以通过它们定制自己的Web容器！

◀ ▶



**Vainycos**

2019-06-03



web容器，既然是容器，那就一定是盛装某种东西的器具。而在web中，盛装的自然是关于web的东西，除了前端的html页面之外，还包括后端的逻辑处理代码，前端页面打开的时候，需要有后端的数据支撑，所以容器装了html页面和后端的代码处理(简单的自我理解，不够严谨)



**行者**

2019-06-02



这算不算 太阳底下无新事 的另类体现。

展开 ▾



**chibohe**

2019-05-27



如果我的服务全都是rpc调用，不涉及http调用，可以不部署在tomcat或者jetty容器中吗？还有你说的Spring都应该指的是SpringMVC吧？

作者回复: 可以的，比如Dubbo。文中的Spring可以理解为SpringMVC。

◀ ▶



**水户洋平◆...**

2019-05-22



从毕业到现在已经工作小一年了，感觉自己一直在从事与业务层面的开发。对于深入的知识总是停留在会用的层面，比如我用的服务器都是tomcat，但是不知道为什么用它，该怎么去优化它。希望可以在这个专栏中去提升自己，让自己达到一个新的高度。



**太阳是块坚...**

2019-05-22



醍醐灌顶！！

展开 ▾



**李青**

2019-05-21



基础很重要，没有servlet基础知识的铺垫，学习web框架，永远只是停留在用的阶段。



**刘三通**

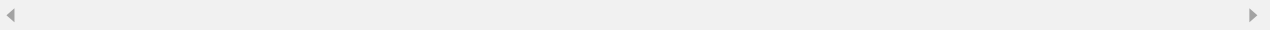
2019-05-19



Spring应用本身就是一个Servlet容器

展开 ▾

作者回复: Spring是Bean容器，不是Servlet容器，它不负责加载和实例化Servlet。



**飞向云端**

2019-05-19



什么叫内嵌方式运行servlet容器，老师有时间普及一下。

展开 ▾

作者回复: 就是你的程序比如SpringBoot直接调用Web容器的提供的API去创建一个Web容器（HTTP服务器和Servlet容器），同时你的程序注册一个Servlet到Servlet容器中，比如SpringMVC的DispatcherServlet，这样请求到达时，Servlet容器负责调用你的Servlet。

