

## 32 | 全面解析HTTP：从HTTP/0.9到HTTP/3的演变

2023-07-05 Barry 来自北京

《Python实战 · 从0到1搭建直播视频平台》



你好，我是 Barry。

从这节课开始，我们正式开启直播模块的学习。

在开发直播平台前，我们需要先学习 HTTP 相关技术。直播平台的核心功能就是基于网络传输实时音视频数据，而 HTTP 是用于在客户端和服务端之间传输数据的协议，它能优化数据传输效率、实现实时性和稳定性，保护用户隐私和数据安全，实现与服务器的交互和集成。

与此同时，网络知识也是一个技术人员的基本功，但是很多技术人员会忽视掉这一块的内容，在工作和面试中吃了很多的亏，所以这节课我们就以 HTTP 为切入点，系统梳理一下这部分内容。

### 初识 HTTP

HTTP 是一种用于传输超文本（如 HTML、XML、JSON 等）的应用层协议。它是互联网上最常用的协议之一。

因为网络协议是我们的必备知识，这里我也顺便说说其他的传输协议。

FTP，用来做文件传输的协议，使用客户端和服务端之间的连接来传输文件。

SMTP，用于发送电子邮件的协议，它使用客户端和服务端之间的连接来传输邮件。

SSH，这是用于远程登录和管理计算机的协议，它使用客户端和服务端之间的安全连接来传输数据。

HTTP 的发展历程从 20 世纪 80 年代开始算起，当时互联网还处于发展的早期阶段。在 HTTP 的早期版本中，它主要用来在客户端和服务端之间传输静态文本和图像文件。

随着互联网的发展，为了适应不断变化的需求，支持更多功能和特性（例如多媒体、动态内容、安全性、可扩展性等），HTTP 协议也在逐渐扩展和改进。

从最初的 HTTP/0.9 的版本到现在的 HTTP/3，HTTP 始终在不断地创新优化，所以才成为了互联网最常用的协议之一。

我这就来为你梳理一下 HTTP 整个发展历程，看看它在每个阶段都有什么发展变革。

名称	作用和关键参数
HTTP/0.9	于 1991 年发布。它非常简单，只能传输纯文本。主要通过 GET 方法从服务器请求整个 HTML 文档，并以纯文本形式作为响应进行传输。
HTTP/1.0	HTTP 1.0 引入了一些新的特性和功能，如缓存、管道化、请求方法和头部信息等。这些功能的引入提高了 HTTP 的性能和可扩展性。同时支持传输任意类型的文件、引入请求头和响应头、支持状态码、引入 POST 方法等。
HTTP/1.1	HTTP/1.1 是在 1997 年发布的，HTTP/1.1 是 HTTP 协议的一次重要更新，引入了许多新功能和改进。它的目标是提供更高效率、更灵活和更可靠的通信。
HTTP/2	2015 年发布，它引入了一些重要的优化和改进，以提高 HTTP 的性能和可扩展性，如多路复用、头部压缩、服务器推送等。这些改进使 HTTP/2 变得更快、更高效，给用户提供了更好的体验。
HTTP/3	2022 年发布。HTTP/3 的引入旨在解决 HTTP/2 中存在的一些性能瓶颈和问题，并提供更快速和可靠的数据传输。

# HTTP 详解

梳理完 HTTP 的发展历程和每个阶段做出的重要调整，我们还需要更深入地了解每个阶段的细节，这才是掌握 HTTP 相关知识的核心。掌握了这些，不仅能够巩固我们对 HTTP 的应用理解，也能让你轻松应对在技术面试中网络相关的技能考察。

## HTTP/0.9

我们先从 HTTP/0.9 这个版本开始说起，HTTP/0.9 最初诞生主要用来在网络之间传递 HTML 超文本，所以被称为超文本传输协议。

当时它的实现也非常简单，只有请求行，通过单一的请求行已经能够满足要获取的内容。同时，对于相应内容也只是需要返回数据就可以，这里的数据格式也只能是 HTML 数据格式，这也在后来的发展中成为了它的缺点。

在后续的使用场景中，用户涌现出大量新的需求，在浏览器传输文件中不单单只有 HTML。例如还有 JS 和 CSS 脚本、图文、音视频等不同类型的文件。所以需要 HTTP 能够满足更多类型文件的传输，在巨大的需求下 HTTP/1.0 诞生了。

## HTTP/1.0

HTTP/1.0 与 HTTP/0.9 相比做了很大的变革。我们这就一起来看一下，它都做了哪些全面的升级。

首先，HTTP 1.0 引入了请求头和响应头，主要是为了提供更多的信息来帮助客户端和服务端更好地处理请求和响应，使得 HTTP 协议更加易于使用和维护。

其次，在请求方法上也开始支持 GET、POST、Head 请求方法。

最后，通过支持缓存机制，来减轻服务器的负担，也引入了状态码，把请求结果反馈做得更加完善。

虽然 HTTP/1.0 做了很大的优化和改变，但在后续应用时还是发现一些核心的问题。

首先，它只有在上一个请求返回之后，才能够再去执行下一个，这样会引起队头阻塞。

其次，它只能支持一个服务器，只有一个域名。

最后，在数据传送方面存在一些资源浪费，每次请求都需要提交完整的头部对象进行传递。

## HTTP/1.1

所以，HTTP/1.0 后来逐渐无法满足技术发展需求，这时 HTTP/1.1 应运而生。它在 1.0 的基础上又做了大量更新迭代。

第一，在效能优化方面，HTTP/1.1 引入了持久连接，大大减少了 TCP 建立连接时消耗的次数，节省了服务器的资源消耗，实现了访问提效。

第二，通过在请求头中新增 Host 字段，解决服务器只能支持一个域名的问题。通过 Host 字段来表示当前的域名地址，服务器可以通过 Host 值来区分处理不同的域名。

第三，新增了 e-tags 和 Cache-control，这也是一次重要的变革。这里说个题外话，我们面试当中经常会被问到这两个参数的用途，它们分别用来判断资源是否发生了变化以及如何处理缓存和过期策略。通过使用 e-tags 和 Cache-control，HTTP 1.1 能够更好地支持缓存和优化网络传输，提高了 Web 性能和可靠性。

第四，添加了 Pipeline。这允许客户端在第一个请求的没完全传输之前，就发送第二个请求，这么做降低了通信的延迟。虽然 Pipeline 在 HTTP/2 才投入稳定的应用，但其实它在 1.1 的阶段就曾投入使用。Pipeline 机制主要还是为了优化性能。具体的执行效果图如下所示。

shikey.com转载分享



还有其他的一些优化，例如 HTTP/1.1 通过引入 Chunk transfer 机制来实现对动态内容的支持。这个阶段还新增了 5 种新的请求方法，我给你整理在了后面的表格里，你可以看一下。

名称	作用和关键参数
OPTIONS	用于获取指定资源的支持请求方法信息
PUT	用于创建或更新指定资源
DELETE	用于删除指定资源
TRACE	用于进行消息路由跟踪，以便客户端可以追踪请求/响应的传输路径
CONNECT	用于通过使用代理服务器进行 TCP 通信

shikey.com转载分享

我们看到了在 HTTP/1.1 阶段发生的巨大的更新，在提效节能方面进行了多维度优化。但是宽带利用率和队头阻塞问题还是没有得到充分解决。

shikey.com转载分享

## HTTP/2.0

HTTP/2.0 的推出原因不言而喻，目的是为了对 HTTP 进行持续的版本迭代和升级。那么 HTTP/2 相比 HTTP/1.1 又做了哪些迭代呢？

第一，HTTP/2 采用二进制格式而非文本格式，二进制协议解析起来更高效、“线上”更紧凑，更重要的是错误更少。

第二，为了解决 HTTP/1.x 中存在的队头阻塞问题，HTTP/2 提出了多路复用的概念。多路复用主要解决队头阻塞问题，具体来说就是实现在同一条 connection 之上可以混合发送分属于不同 stream 的 frame，这样就能达到发送多个 request 的功能的效果。

第三，HTTP/2 使用 Header 压缩机制，持续降低开销。我们都知道，HTTP1 相关版本的 header 带有大量信息，而且每次都要重复发送，这样会造成大量的资源消耗。而 HTTP/2 采用 HPACK 算法进行压缩，节省了资源消耗。

第四，HTTP/2 让服务器可以将响应主动“推送”到客户端缓存中。在 HTTP/2 中，服务器可以在发送响应之前，主动向客户端推送一些资源，这些资源可以是页面中的常见资源，比如图片、样式表、脚本等。通过这种方式，服务器可以在客户端请求资源之前就提前发送响应，减少客户端的等待时间，提高了网页的加载速度。

同时，HTTP/2 还有增加了一些新特性。

引入了优先级（Priority）机制，允许客户端在请求中指定资源的优先级。

引入了流控制（Flow Control）机制，通过在发送方和接收方之间设置缓冲区来控制数据的流量。

采用二进制协议，将 HTTP 协议的文本格式改为二进制格式，减少了传输中的开销，提高传输效率。

不过，HTTP/2 依然存在一些缺点，例如在 TCP 传输过程中，由于单个数据包的丢失而造成的阻塞。这是因为在 HTTP/2 中，多个请求是跑在一个 TCP 管道中的，如果其中任意一路数据流中出现了丢包的情况，那么就会阻塞该 TCP 连接中的所有请求，这就是我们常说的 TCP 上的队头阻塞。

## HTTP/3.0



综合前面的梳理，可以看出尽管 HTTP/2 在 HTTP/1 的基础上做了很多革新，但是依然有一些缺陷。接下来，我们就看看 HTTP/3 又做了哪些变革。

HTTP/3.0 的核心目标是提升性能、加快响应速度、保障交互安全性。HTTP/3.0 的核心是 QUIC 协议，它是基于 UDP 的协议，这代表着在传输效率上相比之前的版本有了大幅提升，这属于颠覆性的变革，不需要再像之前的版本那样，经历复杂的连接环节。

HTTP 3.0 还支持更强的安全性，支持使用 TLS 1.3 协议进行加密和保护数据传输，TLS 1.3 是最新的安全传输协议。这与之之前相比极大的保护了用户的隐私和数据安全，能让用户更加信赖去使用。

在解决多路复用丢包的线头阻塞问题方面，QUIC 保留了 HTTP2.0 多路复用的特性。在之前的多路复用过程中，同一个 TCP 连接上有多个 stream，只要有一个丢包，那在这个 stream 前后的 stream 都会有影响。而在 QUIC 上的 stream 之间并没有依赖关系，所以只关注当下丢包的 stream 就可以，这个方式有效避免了线头阻塞的问题。

另外，HTTP/3 还实现了有序交付。我们都知道 UDP 是不可靠传输协议，为了实现有序交付，QUIC 在每个数据包都设有一个 offset 字段，也就是偏移量，这样接收端就能根据 offset 字段为异步到达的数据包做排序。

## 总结

又到了课程的尾声，我们一起来回顾总结一下这节课的重点。

HTTP 是超文本传输协议，是一种用于在客户端和服务端之间传输数据的协议。除了在线直播平台项目里需要应用到 HTTP 的相关技术知识，我们日常开发里也少不了和 HTTP 打交道。

针对 HTTP 的发展，我们从最初的 HTTP/0.9 到 HTTP/3 进行了全面解读，了解了在每个阶段 HTTP 主要解决的问题，在不同版本都做了哪些突破和优化，又存在哪些不足。你可以结合后面的表格回顾复习一下。

HTTP 版本	主要解决的核心问题	采用的解决方案
HTTP/0.9	解决 HTML 文件传输问题	确立了客户增请求、服多端响应的通信流程
HTTP/1.0	实现不同类型文件传输	设立头部字段
HTTP/1.1	解决创建/断开 TCP 开销大的问题	建立长连接进行复用
HTTP/2.0	解决并发数有限问题	采用二进制分帧的方式来实现
HTTP/3.0	解决 TCP 丢包阻塞问题	采用 UDP 协议, QUIC 中一个连接上的多个 stream 之间没有依赖



最后再分享一点 HTTP 相关知识的学习技巧：你需要明确在不同版本间主要做了哪些创新和解决哪些问题。至于网络部分的考核，更看重你是否能深入掌握 HTTP 的核心原理，希望你在课后参考这个思路再仔细复习一下，这样学习效果会更好。

## 思考题

你觉得 HTTP 和 HTTPS 存在哪些区别？欢迎你总结分享。

期待你在留言区和我交流互动，也推荐你把这节课分享给身边的更多朋友。

© 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。

## 精选留言 (3)

shike.com转载分享



peter

2023-07-06 来自北京

请教老师几个问题：

Q1: Python开发的网站，服务器可以用tomcat吗？

Q2: Python开发的网站，可以用docker等虚拟机部署吗？

Q3: http传XML或JSON，是把XML或JSON放到消息体中吗？什么场景中会用http来传XML或JSON？（我头脑中固定印象就是传html的，认识有局限）

Q4: 二进制格式与文本格式的区别？

文中提到“HTTP/2 采用二进制格式而非文本格式”，不是很理解。文本格式最后不也是要变成二进制传输吗？

shike.com转载分享



Q5: host怎么理解?

Host难道是可以指明多个服务器吗?

Q6: HTTP3用UDP,丢包怎么解决?

作者回复: Q1: Python开发的网站可以使用Tomcat作为服务器。不过, 通常Python开发使用Nginx、Apache作为服务器多一点。

Q2: 是的, Python开发的网站可以使用Docker等虚拟机进行部署。

Q3: 当通过HTTP传输XML或JSON时, 通常是将它们放在HTTP消息的请求体或响应体中。HTTP协议支持传输各种数据类型, 包括文本、二进制等。在Web开发中, 经常使用HTTP传输XML或JSON, 以便在不同的应用程序之间交换数据。

Q4: 二进制格式和文本格式在HTTP/2中都有使用。在HTTP/2中, 可以使用二进制格式来压缩数据, 以减少传输数据量, 提高性能。而文本格式通常用于表示文本数据, 例如HTML、JSON、XML等。虽然二进制格式最终在传输时需要转换成二进制形式, 但它们在处理和存储数据时具有更高的效率和性能。

Q5: 在HTTP中, "Host"头字段是用于指定正在请求的服务器名称和端口号。它可以用于指定多个服务器, 以便客户端可以根据主机名来发送请求。在多域名支持的场景中, 可以通过Host头字段来指定不同的服务器, 以便同一个IP地址上运行多个域名, 这个课程中我有详细说过。

Q6: 虽然HTTP3基于UDP协议, 但UDP本身是无连接的协议, 可能出现数据包丢失的情况。为了解决这个问题, HTTP/3采用了流控制和重传机制来确保数据的可靠传输。同时, HTTP/3还支持流优先级和流量控制等特性, 以确保高优先级的请求能够优先处理, 从而避免丢包对用户体验的影响。第6个问题可能其他同学也有疑问, 这里统一答复。



**coderHOW**

2023-07-05 来自广东

自己没有总结过, 老师帮总结很完全, 很硬核。

对于http和https区别, 就是一般使用的4层模型就到了5层模型, 原来的http和tcp直接握手的变为http和tcp之间加了一个ssl/tls层, 专门作为加密认证等, 就不会改变原来的层原有的作用, 我感觉就是http加了一层装饰器

shikey.com转载分享

作者回复: 感谢同学对课程内容的赞美, 你的总结非常的不错, 对于你的总结我可以补充一条, HTTP S不仅仅是一个简单的加密和认证层, 而是一套完整的安全协议, HTTPS协议在实现过程中会进行更多的安全措施, 例如身份验证、数据完整性保护等。我们的内容可以组合。期待你下次分享, 我们一起加油!!!





Forest

2023-07-05 来自四川

你觉得 HTTP 和 HTTPS 存在哪些区别？

-----  
都是用于网络上传输数据的两种协议；

区别：

安全：HTTP 传输的数据是明文的；HTTPS 使用 SSL 或 TLS 协议对数据进行加密。HTTPS 使用证书对进行身份验证，确保建立的连接是安全的

端口：HTTP 使用 80 端口进行通信，HTTPS 使用 443 端口进行通信

数据完整：HTTPS 提供了数据完整性的保护，通过使用加密算法和证书的加持，可以有效的确保数据在传输过程中没有被篡改

作者回复：总结的非常全面，为你点赞，继续加油，希望你在学习过程中多多总结，多多分享，我们一起加油。



shikey.com转载分享

shikey.com转载分享