

14 | HTTP有哪些优点？又有哪些缺点？

2019-06-28 Chrono

《透视HTTP协议》

课程介绍 >



讲述：Chrono

时长 10:59 大小 12.57M



上一讲我介绍了 HTTP 的五个基本特点，这一讲要说的则是它的优点和缺点。其实这些也应该算是 HTTP 的特点，但这一讲会更侧重于评价它们的优劣和好坏。

上一讲我也留了两道课下作业，不知道你有没有认真思考过，今天可以一起来看看你的答案与我的观点想法是否相符，共同探讨。

不过在正式开讲之前我还要提醒你一下，今天的讨论范围仅限于 HTTP/1.1，所说的优点和缺点也仅针对 HTTP/1.1。实际上，专栏后续要讲的 HTTPS 和 HTTP/2 都是对 HTTP/1.1 优点的发挥和缺点的完善。

领资料

简单、灵活、易于扩展

首先，HTTP 最重要也是最突出的优点是“简单、灵活、易于扩展”。



初次接触 HTTP 的人都会认为，HTTP 协议是很“简单”的，基本的报文格式就是“header+body”，头部信息也是简单的文本格式，用的也都是常见的英文单词，即使不去看 RFC 文档，只靠猜也能猜出个“八九不离十”。

可不要小看了“简单”这个优点，它不仅降低了学习和使用的门槛，能够让更多的人研究和开发 HTTP 应用，而且我在 [第 1 讲](#)时就说过，“简单”蕴含了进化和扩展的可能性，所谓“少即是多”，“把简单的系统变复杂”，要比“把复杂的系统变简单”容易得多。

所以，在“简单”这个最基本的设计理念之下，HTTP 协议又多出了“灵活和易于扩展”的优点。

“灵活和易于扩展”实际上是一体的，它们互为表里、相互促进，因为“灵活”所以才会“易于扩展”，而“易于扩展”又反过来让 HTTP 更加灵活，拥有更强的表现能力。

HTTP 协议里的请求方法、URI、状态码、原因短语、头字段等每一个核心组成要素都没有被“写死”，允许开发者任意定制、扩充或解释，给予了浏览器和服务器的最大程度的信任和自由，也正好符合了互联网“自由与平等”的精神——缺什么功能自己加个字段或者错误码什么的补上就是了。

“请勿跟踪”所使用的头字段 DNT（Do Not Track）就是一个很好的例子。它最早由 Mozilla 提出，用来保护用户隐私，防止网站监测追踪用户的偏好。不过可惜的是 DNT 从推出至今有差不多七八年的历史，但很多网站仍然选择“无视”DNT。虽然 DNT 基本失败了，但这也正说明 HTTP 协议是“灵活自由的”，不会受单方面势力的压制。

“灵活、易于扩展”的特性还表现在 HTTP 对“可靠传输”的定义上，它不限制具体的下层协议，不仅可以使⽤ TCP、UNIX Domain Socket，还可以使⽤ SSL/TLS，甚至是基于 UDP 的 QUIC，下层可以随意变化，而上层的语义则始终保持稳定。

应用广泛、环境成熟

HTTP 协议的另一大优点是“应用广泛”，软硬件环境都非常成熟。

随着互联网特别是移动互联网的普及，HTTP 的触角已经延伸到了世界的每一个角落：从简单的 Web 页面到复杂的 JSON、XML 数据，从台式机上的浏览器到手机上的各种 APP，从看新闻、泡论坛到购物、理财、“吃鸡”，你很难找到一个没有使用 HTTP 的地方。

领资料



不仅在应用领域，在开发领域 HTTP 协议也得到了广泛的支持。它并不限定某种编程语言或者操作系统，所以天然具有“**跨语言、跨平台**”的优越性。而且，因为本身的简单特性很容易实现，所以几乎所有的编程语言都有 HTTP 调用库和外围的开发测试工具，这一点我觉得就不用再举例了吧，你可能比我更熟悉。

HTTP 广泛应用的背后还有许多硬件基础设施支持，各个互联网公司和传统行业公司都不遗余力地“触网”，购买服务器开办网站，建设数据中心、CDN 和高速光纤，持续地优化上网体验，让 HTTP 运行的越来越顺畅。

“应用广泛”的这个优点也就决定了：无论是创业者还是求职者，无论是做网站服务器还是写应用客户端，HTTP 协议都是必须要掌握的基本技能。

无状态

看过了两个优点，我们再来看看一把“双刃剑”，也就是上一讲中说到的“无状态”，它对于 HTTP 来说既是优点也是缺点。

“无状态”有什么好处呢？

因为服务器没有“记忆能力”，所以就不需要额外的资源来记录状态信息，不仅实现上会简单一些，而且还能减轻服务器的负担，能够把更多的 CPU 和内存用来对外提供服务。

而且，“无状态”也表示服务器都是相同的，没有“状态”的差异，所以可以很容易地组成集群，让负载均衡把请求转发到任意一台服务器，不会因为状态不一致导致处理出错，使用“堆机器”的“笨办法”轻松实现高并发高可用。

那么，“无状态”又有什么坏处呢？

既然服务器没有“记忆能力”，它就无法支持需要连续多个步骤的“事务”操作。例如电商购物，首先要登录，然后添加购物车，再下单、结算、支付，这一系列操作都需要知道用户的身份才行，但“无状态”服务器是不知道这些请求是相互关联的，每次都得问一遍身份信息，不仅麻烦，而且还增加了不必要的数据传输量。

所以，HTTP 协议最好是既“无状态”又“有状态”，不过还真有“鱼和熊掌”两者兼得这样的好事，这就是“小甜饼”Cookie 技术（第 19 讲）。

领资料



明文

HTTP 协议里还有一把优缺点一体的“双刃剑”，就是**明文传输**。

“明文”意思就是协议里的报文（准确地说是 header 部分）不使用二进制数据，而是用简单可阅读的文本形式。

对比 TCP、UDP 这样的二进制协议，它的优点显而易见，不需要借助任何外部工具，用浏览器、Wireshark 或者 tcpdump 抓包后，直接用肉眼就可以很容易地查看或者修改，为我们的开发调试工作带来极大的便利。

当然，明文的缺点也是一样显而易见，HTTP 报文的所有信息都会暴露在“光天化日之下”，在漫长的传输链路的每一个环节上都毫无隐私可言，不怀好意的人只要侵入了这个链路里的某个设备，简单地“旁路”一下流量，就可以实现对通信的窥视。

你有没有听说过“免费 WiFi 陷阱”之类的新闻呢？

黑客就是利用了 HTTP 明文传输的缺点，在公共场所架设一个 WiFi 热点开始“钓鱼”，诱骗网民上网。一旦你连上了这个 WiFi 热点，所有的流量都会被截获保存，里面如果有银行卡号、网站密码等敏感信息的话那就危险了，黑客拿到了这些数据就可以冒充你为所欲为。

不安全

与“明文”缺点相关但不完全等同的另一个缺点是“不安全”。

安全有很多的方面，明文只是“机密”方面的一个缺点，在“身份认证”和“完整性校验”这两方面 HTTP 也是欠缺的。

“身份认证”简单来说就是“**怎么证明你就是你**”。在现实生活中比较好办，你可以拿出身份证、驾照或者护照，上面有照片和权威机构的盖章，能够证明你的身份。



但在虚拟的网络世界里这却是个麻烦事。HTTP 没有提供有效的手段来确认通信双方的真实身份。虽然协议里有一个基本的认证机制，但因为刚才所说的明文传输缺点，这个机制几乎可以说是“纸糊的”，非常容易被攻破。如果仅使用 HTTP 协议，很可能你会连到一个页面一模一样但却是个假冒的网站，然后再被“钓”走各种私人信息。



HTTP 协议也不支持“完整性校验”，数据在传输过程中容易被篡改而无法验证真伪。

比如，你收到了一条银行用 HTTP 发来的消息：“小明向你转账一百元”，你无法知道小明是否真的就只转了一百元，也许他转了一千元或者五十元，但被黑客篡改成了一百元，真实情况到底是什么样子 HTTP 协议没有办法给你答案。

虽然银行可以用 MD5、SHA1 等算法给报文加上数字摘要，但还是因为“明文”这个致命缺点，黑客可以连同摘要一同修改，最终还是判断不出报文是否被篡改。

为了解决 HTTP 不安全的缺点，所以就出现了 HTTPS，这个我们以后再说。

性能

最后我们来谈谈 HTTP 的性能，可以用六个字来概括：“不算差，不够好”。

HTTP 协议基于 TCP/IP，并且使用了“请求 – 应答”的通信模式，所以性能的关键就在这两点上。

必须要说的是，TCP 的性能是不差的，否则也不会纵横互联网江湖四十余载了，而且它已经被研究的很透，集成在操作系统内核里经过了细致的优化，足以应付大多数的场景。

只可惜如今的江湖已经不是从前的江湖，现在互联网的特点是移动和高并发，不能保证稳定的连接质量，所以在 TCP 层面上 HTTP 协议有时候就会表现的不够好。

而“请求 – 应答”模式则加剧了 HTTP 的性能问题，这就是著名的“队头阻塞”（Head-of-line blocking），当顺序发送的请求序列中的一个请求因为某种原因被阻塞时，在后面排队的所有请求也一并被阻塞，会导致客户端迟迟收不到数据。

为了解决这个问题，就诞生出了一个专门的研究课题“Web 性能优化”，HTTP 官方标准里就有“缓存”一章（RFC7234），非官方的“花招”就更多了，例如切图、数据内嵌与合并，域名分片、JavaScript“黑科技”等等。

不过现在已经有了终极解决方案：HTTP/2 和 HTTP/3，后面我也会展开来讲。

小结

领资料



1. HTTP 最大的优点是简单、灵活和易于扩展；
2. HTTP 拥有成熟的软硬件环境，应用的非常广泛，是互联网的基础设施；
3. HTTP 是无状态的，可以轻松实现集群化，扩展性能，但有时也需要用 Cookie 技术来实现“有状态”；
4. HTTP 是明文传输，数据完全肉眼可见，能够方便地研究分析，但也容易被窃听；
5. HTTP 是不安全的，无法验证通信双方的身份，也不能判断报文是否被篡改；
6. HTTP 的性能不算差，但不完全适应现在的互联网，还有很大的提升空间。

虽然 HTTP 免不了这样那样的缺点，但你也不要怕，别忘了它有一个最重要的“灵活可扩展”的优点，所有的缺点都可以在这个基础上想办法解决，接下来的“进阶篇”和“安全篇”就会讲到。

课下作业

1. 你最喜欢的 HTTP 优点是哪个？最不喜欢的缺点又是哪个？为什么？
2. 你能够再进一步扩展或补充论述今天提到这些优点或缺点吗？
3. 你能试着针对这些缺点提出一些自己的解决方案吗？

欢迎你把自己的答案写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，欢迎你把文章分享给你的朋友。





== 课外小贴士 ==

- 01 “简洁至上”也是 Apple 公司前领导人乔布斯所崇尚的设计理念。
- 02 与 DNT 类似的还有 P3P (Platform for Privacy Preferences Project) 字段, 用来控制网站对用户的隐私访问, 同样也失败了。
- 03 出于安全的原因, 绝大多数网站都封禁了 80/8080 以外的端口号, 只允许 HTTP 协议“穿透”, 这也是造成 HTTP 流行的客观原因之一。
- 04 HTTP/1.1 以文本格式传输 header, 有严重的数据冗余, 也影响了它的性能。

领资料



透视 HTTP 协议

深入理解 HTTP 协议本质与应用

罗剑锋

奇虎360技术专家


Nginx/OpenResty 开源项目贡献者



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。

分享给需要的人，Ta订阅超级会员，你将得 **50** 元

Ta单独购买本课程，你将得 **20** 元

 生成海报并分享

 赞 16

 提建议

© 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 13 | HTTP有哪些特点？

下一篇 15 | 海纳百川：HTTP的实体数据

领资料



JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费



精选留言 (29)

写留言



Xiao

2019-06-28

老师说的对，以前的时候觉得看书或者文章，人家说什么就是什么，而且很绝对。后来慢慢发现，很多东西都是需要结合业务场景来分析的，在这种业务场景是优点，在另外的业务场景就可能是致命的缺点！那句话：脱离业务场景谈技术就是要流氓！哈哈哈，谢谢老师！

作者回复：自己思考就有收获，尽信书则不如无书。



35



钱

2020-03-29

1：你最喜欢的 HTTP 优点是哪个？最不喜欢的缺点又是哪个？为什么？

最喜欢她的简单，因为简单所以美好，简简单单容易上手，易于理解。

最不喜欢她的不安全和低性能，如果她能做到安全高效，那就不用再去学习别的乱七八糟的协议了，编写RPC接口也就没有其他框架什么事情了，搞微服务应该门槛更低效率更高，当然，她目前做不到，由于考上层以后估计也悬，不过如果基建OK，也许网络带宽不再是瓶颈，她能做到天然的高效传输吧！

2：你能够再进一步扩展或补充论述今天提到这些优点或缺点吗？

简单和易扩展，我认为是矛盾的，很少系统能做到即易扩展又简单，毕竟以后需要什么谁也不

领资料



能未卜先知，留下扩展的余地毕竟会增加复杂度吧！如果设计的不好，复杂度也许会急剧上升，不过HTTP做的很好貌似没有出现这种情况。不知是为什么？是公共的预定义基本OK吗？其实扩展的不是很多吗？还请老师分享一下。

3：你能试着针对这些缺点提出一些自己的解决方案吗？

有无状态已有方案，可以自由选择

安全性后面也会采用加密的方式来解决

低效这个是相对的，靠上必然会低效，如果基建进一步加强，也许能进一步解决，不过这个看需求和增加的带宽那个更大吧！

明文最早的初心就是分享信息，明文我觉得太正常了，现在多了安全性的需求，所以，才渴望更安全的，此时的HTTP用于分享信息已是一部分功能，她还可以实现各种各样的诉求。

如果协议里设置的有类似开关的东西就好了，可以选择是否有状态、可以选择是否明文、可以选择是否加密，使用者仅需要控制开关就能实现相关的功能，不需自己实现，那就更好了。

作者回复：还是这么认真写作业啊，笑。

1.http/2和http/3可以解决你说的问题。

2.我觉得http一开始的文本格式头结构算是误打误撞，正好即简单又容易扩展，有点像xml。

3.请看后面的http/2。



👍 9



レイン小雨

2019-06-29

老师，我想问一下关于“队头阻塞”的概念，正好上周团队有人分享这方面的知识，当时是说浏览器针对一个域名最多同时建立6个连接通道，也就是支持6个http并发，当一个页面中有100个资源文件需要加载的时候，就只能6个6个的串行加载，第七个请求要等到第一个请求结果返回之后才能发出。这么理解队头阻塞的概念对吗？

作者回复：你说的这6个连接是为了解决队头阻塞而实现的并发连接。

而“第七个”实际上是排队等待，还不是队头阻塞。

马上就会讲到这个问题了，再等几天。

共 4 条评论 >

👍 9

领资料



业余草

2019-06-28

老师不仅技术好，文章写的也很好。比较喜欢看这种风格的文章，可以加个微信好友吗？我微

信: xttblog

作者回复: 抱歉, 个人微信用的比较少, 可以在GitHub上交流。

共 3 条评论 >

👍 8



Geek_d59aa7

2020-07-02

老师, 问个沙雕问题, 什么叫明文传输, 难道http里的信息不是转成二进制传输的吗, 看网上的意思好像明文传输是直接传输字母和数字?? 不是二进制?

作者回复: 明文是相对于密文而言的, 意思是可以直接看到内容, 而密文是经过加密的, 看到的是乱码。

可以参考后面的安全篇。



👍 3



欢乐的小马驹

2020-03-11

老师好, 小贴士里面提到了: 绝大多数的网站都封禁了80/8080以外的端口号, 只允许HTTP进行穿透。

1、最初为什么要封禁其他端口只留80/8080呢? 2、穿透是什么意思?

作者回复:

1.因为80/8080只用于http协议传输网页, 禁掉其他端口就降低了被嗅探、攻击的可能性, 提高安全性。

2.穿透是个通俗的说法, 就是通过80端口可以一路畅通, 一直传到后端去处理, 而不会在服务器被挡住。



👍 3



小丽

2019-07-13

老师讲的非常透彻, 有些话看似简单, 却值得琢磨, 希望老师多讲一点关于身份认证和安全的東西

作者回复: 下周就会进入安全篇了, 详细讲解https、ssl/tls。



👍 3

领资料





-W.LI-

2019-06-28

老师好!传输的时候序列化方式属于HTTP范畴么?现在大多用的json形式可是cloud和dubbo在高并发情况下dubbo性能比cloud好。网上说是序列化方式不一样。也不晓得是不是

作者回复: http不管body的格式, 所以可以自己选择序列化方式。

json的优点是易读易解析, 但明文显然格式容易, 开销大, 所以要求高性能都用专门的二进制序列化。

我用c/c++比较多, 对Java不太了解, 不好评价, 抱歉了。

共 3 条评论 >



3



Xiao

2019-06-28

不觉得明文是缺点, 因为http本身就是一套标准规范, 而且前面也说它是非常灵活的, 所以个人觉得明文也是一种场景, 用户也可以选择密文传输!

作者回复: 明文是把“双刃剑”, 很多时候是优点, 但换个场景就是缺点了。

什么事情都是这样, 没有绝对, 就看应用的场景。



3



凉人。

2020-03-17

大概最喜欢应用广泛的特点。这点太适用了。

作者回复: 这也是我最喜欢的特点。



2



LIKE

2021-03-29

HTTP. 身份认证, 可以通过加入通信双方约定的密钥做摘要算法, 完成身份认证。

作者回复: 这个就是要用HTTPS了。



1

领资料





彩色的沙漠
2019-06-28

虽然使用了HTTPS金融领域还是需要对数据加密和完整性检验，加密对body数据好加密对于header部分不好加密，需要对KV一个一个加密有点麻烦，不过在传输的时候一般把字段都用body传输，header里面不使用自定义字段。

作者回复: 用了https不就都全加密了吗?

你的办法也对，保密的部分放在body里，反正http格式很灵活，收到以后随便解释。



1



cake
2021-11-01

“无状态”服务器是不知道这些请求是相互关联的，每次都得问一遍身份信息，不仅麻烦，而且还增加了不必要的数据传输量 老师请问下，“每次都得问一遍身份信息”这个在没有cookie时是怎么实现的呢

作者回复: 其实和cookie的做法是类似的，也是用自定义头字段，只是cookie给标准化了。



Leolee
2021-04-21

作业：

- 1、我最喜欢HTTP的应用广泛，学会了就可以受用很久；我不喜欢HTTP的不安全，因为不安全已经是致命的缺陷了，不然也不会出现HTTPS和HTTP2了；
- 2、PASS（python的pass，留待以后再写）；
- 3、PASS（python的pass，留待以后再写）。

HTTP优点：

简单、灵活、易于扩展，应用广泛、软硬件环境成熟、无状态

HTTP缺点：

明文传输、不安全、性能不算差但也不够好

HTTP优缺点并存：

明文传输、无状态

简单、灵活、可扩展

HTTP协议很简单，报文格式就是header+body，头部信息也是简单的文本格式，用的也是常见的英文单词。学习使用的门槛很低。正是基于简单这个最基本的设计理念，才有了HTTP最锋利的剑：灵活可扩展！简单相当于一张白纸，你想怎么写都行。

领资料



HTTP里的请求方法、URI、状态码、原因短语、头字段等每一个核心组件都没有被写死，允许任意定制、扩充和解释，缺什么功能自己写就可以了。

HTTP不限制具体的下层协议，下层协议可以随意变化，上层的语义却可以始终保持稳定。

应用广泛、软硬件环境成熟

HTTP已经遍布互联网世界，很能找到一个没有使用HTTP的地方。无论应用领域还是开发领域都得到了广泛支持。HTTP天然具有跨语言、跨平台的特性，几乎所有编程语言都有HTTP调用库和外围的开发测试工具。

很多互联网和传统行业公司都购买服务器开办网站、建设数据中心、CDN和高速光纤，持续优化上网体验。

无状态

无状态令服务器不需要额外的资源来记录状态信息，负担轻一些。

无状态令服务器没有状态的差异，很容易组成集群，让负载均衡把请求转发到任意一台服务器，不会因为昨天不一致导致处理出错，直接买多几台服务器就可以轻松实现高并发高可用。无状态的缺点就是没有记忆能力，也就无法支持需要连续多个步骤的事务操作，每一次都要问一次身份信息，不仅麻烦，也增加了不必要的数据传输量。但这个问题可以通过Cookies解决。

明文

是优点也是缺点；

明文的意思就是协议里的报文header部分，不使用二进制数据，而使用简单可阅读的文本形式。优点就是不需要借助任何外部工具，抓包后直接就可以查看或者修改，开发调试起来很便利。

缺点就是毫无隐私可言，只需要侵入了这个链路里的某个设备，简单地旁路一下浏览，就可以实现对通信的窥视。

不安全

HTTP没有提供有效的手段来确认通信双方的真实身份，虽然有基本的认证机制，但在明文传输面前几乎等同无效。

HTTP也不支持完整性校验，数据在传输过程中很容易被篡改而无法验证真伪。

正是因为这个缺陷，所以出现了HTTPS。

性能

不算差、不够好

因为请求-应答模式决定了按顺序发送的请求序列中有一个请求因为某种原因被阻塞时，后面一长串的请求也会被阻塞，会导致客户端迟迟收不到数据。

其他

出于安全原因，绝大多数网站都封禁了80/8080意外的端口号，只允许HTTP进行穿透，这也是导致HTTP流行的客观原因之一。80/8080是HTTP端口。



作者回复: great, keep going.



coral

2021-04-06

1. 你最喜欢的 HTTP 优点是哪个？最不喜欢的缺点又是哪个？为什么？

最喜欢明文的传输，一目了然，上手几乎没有门槛，拿来就能用。不喜欢的是明文可能会被截获带来安全性的问题。当然现在已经有了https（还没有搞清楚它是怎么加密的）很好奇为什么在一开始http设计的时候没有把这个考虑进去。是因为早期的互联网，比较着重在公开的信息，没有这么多隐私的需求吗？

2. 你能够再进一步扩展或补充论述今天提到这些优点或缺点吗？

通讯方式必须一来一回，且需要由客户端发起。服务器无法主动发送信息。这个特点减小了服务器的工作量，但增加了状态变化的监听、即时推送的难度。（但这个需求已经随着网络带宽的增加越来越普遍了）

3. 你能试着针对这些缺点提出一些自己的解决方案吗？

监听的问题，有没有办法增加一条监听通道，认证过身份的客户端就可以接收服务器推送，这个认证需要定时刷新确保客户端alive

作者回复:

1.正确，早期的互联网是田园时期，民风淳朴。

2.good。

3.这个就需要由http/2来解决了，不过你的思路很不错。



飞翔的葱油饼

2021-04-06

有个问题啊。老师在上节课中有提到HTTP有可靠传输的特点，这是由于HTTP基于TCP协议，而TCP提供了可靠传输，而这节中提到HTTP的灵活性，其下层使用的传输层协议是可以变的，甚至可以使用UDP，那这样的情况下，HTTP是否就不具有可靠传输性了呢？

作者回复: http不能直接用udp，而是在udp之上的quic，它的下层协议要求必须是可靠的传输。

领资料



共 3 条评论 >





Geek_Maggie
2021-03-15

自己看书总觉得像是学会了但是浮于表面。听了老师课程，结合实际去理解，感觉加深了不少印象

作者回复: 对，多动手做实验，实践出真知。



刘杰

2021-03-11

- 1、我最喜欢HTTP简单、灵活易于扩展的优点，易于扩展就意味着有无限的可能。最不喜欢不安全的缺点，不安全意味着无法知道真实身份是我想要连接的，容易被有心人利用
- 2、HTTP的另外一个优点应用广泛，可以使我们轻易的将开发的应用部署在服务器上，让大家可以轻易访问
- 3、关于不安全的，可以使用HTTPS。关于明文问题，可以考虑客户端和服务端自定义一套自己能看懂的编码。

作者回复: 说的很好。

明文和不安全都可以用https来解决。



王子凡kerry

2021-02-16

老师，HTTP1.1和HTTP2分别适合什么场景呢？不可能1.1跟2相比，一点优点都没有吧

作者回复: http/1的优点是简单、应用广泛、支持的工具多，在大多数要求不高的地方用http/1就足够了。

而http/2是专门为了解决传输性能和安全问题而设计的，有tls1.2和多路复用，适合高流量高并发的场合。

可以看后面的飞翔篇，进一步了解http/2的特点。



领资料



我的朋友叫垃圾呆

2021-01-03

老师，我查了资料之后还是不太理解 文本协议和二进制协议，我看之前的文章有说，从http2

开始，变成了二进制协议，这篇文章说 tcp是二进制协议，而http是文本协议，不太能理解两者的区别，在传输信息的时候不都是转换成二进制的字节流进行传输的吗

作者回复: 这种说法主要是以协议是否human readable为判断标准的，就是说传输的数据是否让人直接不借助工具就能够阅读。

对比一下，http协议的状态行、头字段都用的是ASCII码，可以直接看，而tcp的头用的是二进制位，不能直接看。

共 2 条评论 >

