

HTTP1.0和HTTP1.1的一些区别

- 缓存处理：http1.0使用Expires + Last-Modified/If-Modified-Since; http1.1使用Cache-Control、Etag/If-None-Match
- 默认长连接：Http1.1中默认开启Connection: keep-alive, 使得多个http请求可以复用同一个tcp连接, 节省去了频繁建立tcp连接的开销
- 允许范围请求：增加范围请求相关请求头Ranges, 允许只请求资源的一部分，助力大文件下载和断点续传
- Host请求头：默认每个请求都要带host头，不说的话会报400

HTTP1.X存在的问题

- 明文传输：http1.x的报文内容是明文的，有安全隐患
- header过大：http1.x的header携带的内容不怎么变化，但是每次请求都要带上且内容体积也很大，增加了用户流量

HTTP2.0的改进

- 二进制分帧和多路复用：http2.0的报文内容不再是明文，而是变成了一个二个的二进制帧，多个请求可以在一个tcp通道下并行传输(详见[浅析HTTP/2的多路复用](#))
- 头部压缩：对header用HPACK算法进行压缩，减小header体积
- 服务端推送：在 HTTP/2 当中，服务器已经不再是完全被动地接收请求，响应请求，它也能新建 stream 来给客户端发送消息，当 TCP 连接建立之后，比如浏览器请求一个 HTML 文件，服务器就可以在返回 HTML 的基础上，将 HTML 中引用到的其他资源文件一起返回给客户端，减少客户端的等待。

开发时针对Http的性能优化

- 浏览器针对http并发请求做了限制（chrome下一个域名最多允许6个并发请求），为了增加并行请求数，在使用http1.x时可以给一个服务端设置多个域名。但是在http2.0下，由于所有报文都是并行的，因此没必要做这个优化
- 为了避免多次请求重复建立连接造成的开销，http1.x下常做的优化是合并文件，减少请求数，在http2下，该优化也没有必要

Nginx开启http2.0

目前支持http2.0的浏览器要求必须开启ssl/tls, 因此开启http2.0的前提是配置https，然后再配置 `listen 443 ssl http2` 即可