

Http版本区别

HTTP 1.0

- 请求与响应支持头域
- 响应对象以一个响应状态行开始
- 响应对象不只限于超文本
- 开始支持客户端通过POST方法向Web服务器提交数据，支持GET、HEAD、POST方法
- 支持长连接（但默认还是使用短连接），缓存机制，以及身份认证

HTTP 1.1

- Persistent Connection（keepalive连接）
- chunked编码传输
- 字节范围请求
- Pipelining（请求流水线）
- 请求消息和响应消息都应支持Host头域
- 新增了一批Request method

HTTP1.1增加了OPTIONS,PUT,DELETE,TRACE,CONNECT方法

- 缓存处理

HTTP/1.1在1.0的基础上加入了一些cache的新特性，引入了实体标签，一般被称为e-tags，新增更为强大的Cache-Control头。

HTTP 2.0

- 多路复用（二进制分帧）

HTTP 2.0最大的特点: 不会改动HTTP 的语义，HTTP 方法、状态码、URI 及首部字段，等等这些核心概念上一如往常，却能致力于突破上一代标准的性能限制，改进传输性能，实现低延迟和高吞吐量。而之所以叫2.0，是在于新增的二进制分帧层。在二进制分帧层上，HTTP 2.0 会将所有传输的信息分割为更小的消息和帧,并对它们采用二进制格式的编码，其中HTTP1.x的首部信息会被封装到Headers帧，而我们的request body则封装到Data帧里面。

- 头部压缩

当一个客户端向相同服务器请求许多资源时，像来自同一个网页的图像，将会有大量的请求看上去几乎同样的，这就需要压缩技术对付这种几乎相同的信息。

- 随时复位

HTTP1.1一个缺点是当HTTP信息有一定长度大小数据传输时，你不能方便地随时停止它，中断TCP连接的代价是昂贵的。使用HTTP2的RST_STREAM将能方便停止一个信息传输，启动新的信息，在不中断连接的情况下提高带宽利用效率。

- 服务器端推流: **Server Push**

客户端请求一个资源X，服务器端判断也许客户端还需要资源Z，在无需事先询问客户端情况下将资源Z推送到客户端，客户端接受到后，可以缓存起来以备后用。

- 优先权和依赖

每个流都有自己的优先级别，会表明哪个流是最重要的，客户端会指定哪个流是最重要的，有一些依赖参数，这样一个流可以依赖另外一个流。优先级别可以在运行时动态改变，当用户滚动页面时，可以告诉浏览器哪个图像是最重要的，你也可以在一组流中进行优先筛选，能够突然抓住重点流。