

HTTP请求头头们！

HTTP报文首部

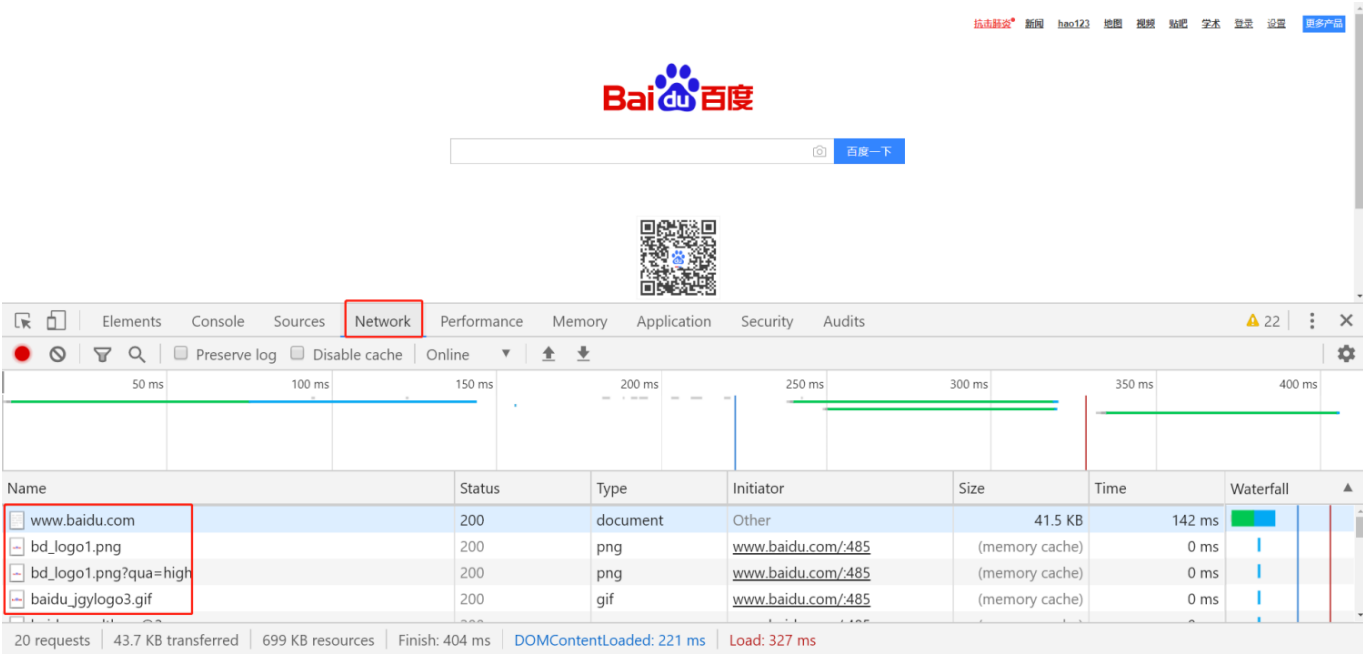
HTTP首部字段

HTTP首部字段的格式

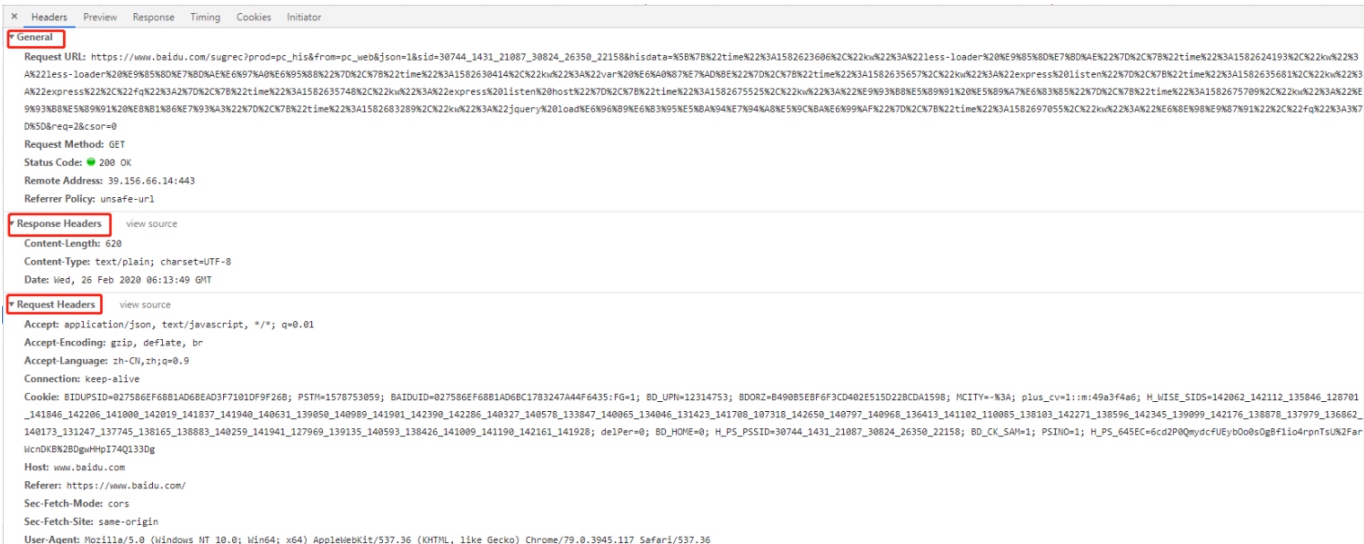
HTTP/1.1 首部字段一览

常用首部字段

在浏览器中不难发现各种请求，可能是数据接口，也可能是静态源。无论那个，我们都可以通过浏览器看到我们访问的内容。以谷歌为例：在控制台network选项中能开到该网站发送的任何HTTP请求。这是百度首页的若干请求



我们拿一个具体的请求来看一下都有什么？

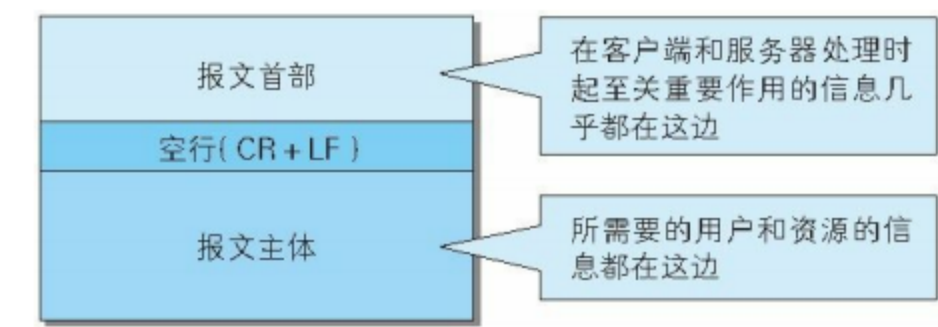


我们看到了是这样的一张图！！。

里面圈了三部分：分别是 General, Request, Response 这三部分，翻译过来，分别为：**通用首部，请求首部，相应首部**。

HTTP 协议的请求和响应报文中必定包含 HTTP 首部，只是我们平时在使用 Web 的过程中感受不到它。

HTTP报文首部



HTTP 协议的请求和响应报文中必定包含 HTTP 首部。首部内容为客 户端和服务端分别处理请求和响应提供所需要的信息。对于客户端用户来说，这些信息中的大部分内容都无须亲自查看。

HTTP首部字段

HTTP 首部字段传递重要信息。HTTP 首部字段是构成 HTTP 报文的要素之一。在客户端与服务器之间以 HTTP 协议进行通信的过程中，无论是请求还是响应都会使用首部字段，**它能起到传递额外重要作用的作用。使用首部字段是为了给浏览器和服务端提供报文主体大小、所使用的语言、认证信息等内容。**

HTTP 首部字段根据实际用途被分为以下 **4 种类型**。

通用首部字段 (General Header Fields)

请求报文和响应报文两方都会使用的首部。

请求首部字段 (Request Header Fields)

从客户端向服务器端发送请求报文时使用的首部。补充了请求的附加内容、客户端信息、响应内容相关优先级等信息。

响应首部字段 (Response Header Fields)

从服务器端向客户端返回响应报文时使用的首部。补充了响应的附加内容，也会要求客户端附加额外的内容信息。

实体首部字段 (Entity Header Fields)

针对请求报文和响应报文的实体部分使用的首部。补充了资源内容更新时间等与实体有关的信息。

对于浏览器来说，实体首部直接放到响应或者请求首部当中所以没有做额外的分类。

HTTP首部字段的格式

▼ General

Request URL: `https://www.baidu.com/`

Request Method: `GET`

Status Code:  `200 OK`

Remote Address: `39.156.66.18:443`

Referrer Policy: `unsafe-url`

▼ Response Headers

[view source](#)

Bdpagetype: `1`

Bdqid: `0xa61f7c0900150c04`

Cache-Control: `private`

Connection: `keep-alive`

根据这部分，不难发现首部字段的格式，类似于js当中的key Value形式。也就是所谓的键值对形式。

HTTP 首部字段是由首部字段名和字段值构成的，中间用冒号“:”分隔。

首部字段名: 字段值

例如，在 HTTP 首部中以 Content-Type 这个字段来表示报文主体的 对象类型。

Content-Type: `text/html`

就以上述示例来看，首部字段名为 Content-Type，字符串 `text/html` 是字段值。

另外，字段值对应单个 HTTP 首部字段可以有多个值，如下所示。

Keep-Alive: `timeout=15, max=100`

HTTP/1.1 首部字段一览

1. 通用首部字段

首部字段名	说明
Cache-Control	控制缓存的行为
Connection	逐跳首部、连接的管理
Date	创建报文的日期时间
Pragma	报文指令
Trailer	报文末端的首部一览
Transfer-Encoding	指定报文主体的传输编码方式
Upgrade	升级为其他协议
Via	代理服务器的相关信息
Warning	错误通知

2. 请求首部字段

首部字段名	说明
Accept	用户代理可处理的媒体类型
Accept-Charset	优先的字符集
Accept-Encoding	优先的内容编码
Accept-Language	优先的语言（自然语言）
Authorization	Web认证信息
Expect	期待服务器的特定行为
From	用户的电子邮箱地址
Host	请求资源所在服务器
If-Match	比较实体标记（ETag）
If-Modified-Since	比较资源的更新时间
If-None-Match	比较实体标记（与 If-Match 相反）
If-Range	资源未更新时发送实体 Byte 的范围请求
If-Unmodified-Since	比较资源的更新时间（与If-Modified-Since相反）
Max-Forwards	最大传输逐跳数
Proxy-Authorization	代理服务器要求客户端的认证信息
Range	实体的字节范围请求
Referer	对请求中 URI 的原始获取方
TE	传输编码的优先级
User-Agent	HTTP 客户端程序的信息

3. 响应首部字段

首部字段名	说明
Accept-Ranges	是否接受字节范围请求
Age	推算资源创建经过时间
ETag	资源的匹配信息
Location	令客户端重定向至指定URI
Proxy-Authenticate	代理服务器对客户端的认证信息
Retry-After	对再次发起请求的时机要求
Server	HTTP服务器的安装信息
Vary	代理服务器缓存的管理信息
WWW-Authenticate	服务器对客户端的认证信息

4. 实体首部字段

首部字段名	说明
Allow	资源可支持的HTTP方法
Content-Encoding	实体主体适用的编码方式
Content-Language	实体主体的自然语言
Content-Length	实体主体的大小（单位：字节）
Content-Location	替代对应资源的URI
Content-MD5	实体主体的报文摘要
Content-Range	实体主体的位置范围
Content-Type	实体主体的媒体类型
Expires	实体主体过期的日期时间
Last-Modified	资源的最后修改日期时间

常用首部字段

1.AcceptAccept: text/html 浏览器可以接受服务器回发的类型为 text/html。Accept: */* 代表浏览器可以处理所有类型,(一般浏览器发给服务器都是发这个)。

2.Accept-EncodingAccept-Encoding: gzip, deflate 浏览器申明自己接收的编码方法，通常指定压缩方法，是否支持压缩，支持什么压缩方法（gzip, deflate），（注意：这不是只字符编码）。

3.Accept-LanguageAccept-Language:zh-CN,zh;q=0.9 浏览器申明自己接收的语言。

4.ConnectionConnection: keep-alive 当一个网页打开完成后，客户端和服务器之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接。Connection: close 代表一个Request完成后，客户端和服务器之间用于传输HTTP数据的TCP连接会关闭，当客户端再次发送Request，需要重新建立TCP连接。

5.Host（发送请求时，该报头域是必需的）Host:www.baidu.com 请求报头域主要用于指定被请求资源的Internet主机和端口号，它通常从HTTP URL中提取出来的。

6.RefererReferer:https://www.baidu.com/?tn=62095104_8_oem_dg 当浏览器向web服务器发送请求的时候，一般会带上Referer，告诉服务器我是从哪个页面链接过来的，服务器籍此可以获得一些信息用于处理。

7.User-AgentUser-Agent:Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36 告诉HTTP服务器，客户端使用的操作系统和浏览器的名称和版本。

8.Cache-ControlCache-Control:private 默认为private 响应只能够作为私有的缓存，不能再用户间共享Cache-Control:public 响应会被缓存，并且在多用户间共享。正常情况，如果要求HTTP认证,响应会自动设置为 private.Cache-Control:must-revalidate 响应在特定条件下会被重用，以满足接下来的请求，但是它必须到服务器端去验证它是不是仍然是最新的。Cache-Control:no-cache 响应不会被缓存,而是实时向服务器端请求资源。Cache-Control:max-age=10 设置缓存最大的有效时间，但是这个参数定义的是时间大小（比如：60）而不是确定的时间点。单位是[秒 seconds]。Cache-Control:no-

store 在任何条件下，响应都不会被缓存，并且不会被写入到客户端的磁盘里，这也是基于安全考虑的某些敏感的响应才会使用这个。

9.Cookie Cookie是用来存储一些用户信息以便让服务器辨别用户身份的（大多数需要登录的网站上面会比较常见），比如cookie会存储一些用户的用户名和密码，当用户登录后就会在客户端产生一个cookie来存储相关信息，这样浏览器通过读取cookie的信息去服务器上验证并通过后会判定你是合法用户，从而允许查看相应网页。当然cookie里面的数据不仅仅是上述范围，还有很多信息可以存储在cookie里面，比如sessionid等。

10.Range（用于断点续传）Range:bytes=0-5 指定第一个字节的位置和最后一个字节的位置。用于告诉服务器自己想取对象的哪部分。

11.Cache-Control（对应请求中的Cache-Control）Cache-Control:private 默认为private 响应只能够作为私有的缓存，不能再用户间共享Cache-Control:public 浏览器和缓存服务器都可以缓存页面信息。Cache-Control:must-revalidate 对于客户机的每次请求，代理服务器必须想服务器验证缓存是否过时。Cache-Control:no-cache 浏览器和缓存服务器都不应该缓存页面信息。Cache-Control:max-age=10 是通知浏览器10秒之内不要烦我，自己从缓冲区中刷新。Cache-Control:no-store 请求和响应的信息都不应该被存储在对方的磁盘系统中。

12.Content-Type Content-Type: text/html; charset=UTF-8 告诉客户端，资源文件的类型，还有字符编码，客户端通过utf-8对资源进行解码，然后对资源进行html解析。通常我们会看到有些网站是乱码的，往往就是服务器端没有返回正确的编码。

13.Content-Encoding Content-Encoding: gzip 告诉客户端，服务端发送的资源是采用gzip编码的，客户端看到这个信息后，应该采用gzip对资源进行解码。

14.Date Date: Tue, 03 Apr 2018 03:52:28 GMT 这个是服务端发送资源时的服务器时间，GMT是格林尼治所在地的标准时间。http协议中发送的时间都是GMT的，这主要是解决在互联网上，不同时区在相互请求资源的时候，时间混乱问题。

15.Server Server: Tengine/1.4.6 这个是服务器和相对应的版本，只是告诉客户端服务器信息。

16.Transfer-Encoding Transfer-Encoding: chunked 这个响应头告诉客户端，服务器发送资源的方式是分块发送的。一般分块发送的资源都是服务器动态生成的，在发送时还不知道发送资源的大小，所以采用分块发送，每一块都是独立的，独立的块都能标示自己的长度，最后一块是0长度的，当客户端读到这个0长度的块时，就可以确定资源已经传输完了。

17.Expires Expires: Sun, 1 Jan 2000 01:00:00 GMT 这个响应头也是跟缓存有关的，告诉客户端在这个时间前，可以直接访问缓存副本，很显然这个值会存在问题，因为客户端和服务器的时间不一定会都是相同的，如果时间不同就会导致问题。所以这个响应头是没有Cache-Control: max-age=*这个响应头准确的，因为max-age=date中的date是个相对时间，不仅更好理解，也更准确。1

18.Last-Modified Last-Modified: Dec, 26 Dec 2015 17:30:00 GMT 所请求的对象的最后修改日期(按照 RFC 7231 中定义的“超文本传输协议日期”格式来表示)

19.Connection Connection: keep-alive 这个字段作为回应客户端的Connection: keep-alive，告诉客户端服务器的tcp连接也是一个长连接，客户端可以继续使用这个tcp连接发送http请求。

20.Etag ETag: "737060cd8c284d8af7ad3082f209582d" 就是一个对象（比如URL）的标志值，就一个对象而言，比如一个html文件，如果被修改了，其Etag也会别修改，所以，ETag的作用跟Last-Modified的作用差不多，主要供WEB服务器判断一个对象是否改变了。比如前一次请求某个html文件时，

获得了其 ETag，当这次又请求这个文件时，浏览器就会把先前获得ETag值发送给WEB服务器，然后WEB服务器会把这个ETag跟该文件的当前ETag进行对比，然后就知道这个文件有没有改变了。

21.RefreshRefresh: 5; url=http://baidu.com 用于重定向，或者当一个新的资源被创建时。默认会在5秒后刷新重定向。

22.Access-Control-Allow-Origin Access-Control-Allow-Origin: * *号代表所有网站可以跨域资源共享，如果当前字段为*那么Access-Control-Allow-Credentials就不能为trueAccess-Control-Allow-Origin: www.baidu.com 指定哪些网站可以跨域资源共享

23.Access-Control-Allow-MethodsAccess-Control-Allow-Methods: GET,POST,PUT,DELETE 允许哪些方法来访问

24.Access-Control-Allow-CredentialsAccess-Control-Allow-Credentials: true 是否允许发送cookie。默认情况下，Cookie不包括在CORS请求之中。设为true，即表示服务器明确许可，Cookie可以包含在请求中，一起发给服务器。这个值也只能设为true，如果服务器不要浏览器发送Cookie，删除该字段即可。如果access-control-allow-origin为*，当前字段就不能为true

25.Content-RangeContent-Range: bytes 0-5/7877 指定整个实体中的一部分的插入位置，他也指示了整个实体的长度。在服务器向客户返回一个部分响应，它必须描述响应覆盖的范围和整个实体长度。

