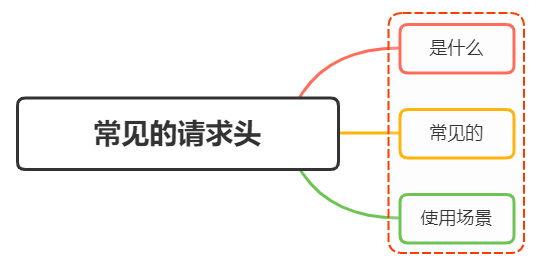
# 面试官：说说 HTTP 常见的请求头有哪些? 作用？



## 一、是什么

HTTP头字段（HTTP header fields）,是指在超文本传输协议（HTTP）的请求和响应消息中的消息头部分

它们定义了一个超文本传输协议事务中的操作参数

HTTP头部字段可以自己根据需要定义，因此可能在 Web服务器和浏览器上发现非标准的头字段

下面是一个HTTP请求的请求头：

GET /home.html HTTP/1.1  
Host: developer.mozilla.org  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0) Gecko/20100101 Firefox/50.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://developer.mozilla.org/testpage.html  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
If-Modified-Since: Mon, 18 Jul 2016 02:36:04 GMT  
If-None-Match: "c561c68d0ba92bbeb8b0fff2a9199f722e3a621a"  
Cache-Control: max-age=0

## 二、分类

常见的请求字段如下表所示：

|  |  |  |
| --- | --- | --- |
| 字段名 | 说明 | 示例 |
| Accept | 能够接受的回应内容类型（Content-Types） | Accept: text/plain |
| Accept-Charset | 能够接受的字符集 | Accept-Charset: utf-8 |
| Accept-Encoding | 能够接受的编码方式列表 | Accept-Encoding: gzip, deflate |
| Accept-Language | 能够接受的回应内容的自然语言列表 | Accept-Language: en-US |
| Authorization | 用于超文本传输协议的认证的认证信息 | Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ== |
| Cache-Control | 用来指定在这次的请求/响应链中的所有缓存机制 都必须 遵守的指令 | Cache-Control: no-cache |
| Connection | 该浏览器想要优先使用的连接类型 | Connection: keep-alive Connection: Upgrade |
| Cookie | 服务器通过 Set- Cookie （下文详述）发送的一个 超文本传输协议Cookie | Cookie: $Version=1; Skin=new; |
| Content-Length | 以 八位字节数组 （8位的字节）表示的请求体的长度 | Content-Length: 348 |
| Content-Type | 请求体的 多媒体类型 | Content-Type: application/x-www-form-urlencoded |
| Date | 发送该消息的日期和时间 | Date: Tue, 15 Nov 1994 08:12:31 GMT |
| Expect | 表明客户端要求服务器做出特定的行为 | Expect: 100-continue |
| Host | 服务器的域名(用于虚拟主机 )，以及服务器所监听的传输控制协议端口号 | Host: en.wikipedia.org:80 Host: en.wikipedia.org |
| If-Match | 仅当客户端提供的实体与服务器上对应的实体相匹配时，才进行对应的操作。主要作用时，用作像 PUT 这样的方法中，仅当从用户上次更新某个资源以来，该资源未被修改的情况下，才更新该资源 | If-Match: "737060cd8c284d8af7ad3082f209582d" |
| If-Modified-Since | 允许在对应的内容未被修改的情况下返回304未修改 | If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT |
| If-None-Match | 允许在对应的内容未被修改的情况下返回304未修改 | If-None-Match: "737060cd8c284d8af7ad3082f209582d" |
| If-Range | 如果该实体未被修改过，则向我发送我所缺少的那一个或多个部分；否则，发送整个新的实体 | If-Range: "737060cd8c284d8af7ad3082f209582d" |
| Range | 仅请求某个实体的一部分 | Range: bytes=500-999 |
| User-Agent | 浏览器的浏览器身份标识字符串 | User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:12.0) Gecko/20100101 Firefox/21.0 |
| Origin | 发起一个针对 跨来源资源共享 的请求 | Origin: http://www.example-social-network.com |

## 三、使用场景

通过配合请求头和响应头，可以满足一些场景的功能实现：

### 协商缓存

协商缓存是利用的是【Last-Modified，If-Modified-Since】和【ETag、If-None-Match】这两对请求头响应头来管理的

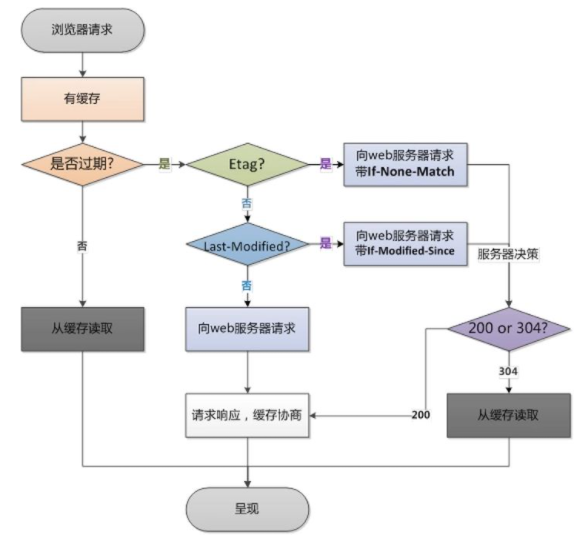
Last-Modified 表示本地文件最后修改日期，浏览器会在request header加上If-Modified-Since（上次返回的Last-Modified的值），询问服务器在该日期后资源是否有更新，有更新的话就会将新的资源发送回来

Etag就像一个指纹，资源变化都会导致ETag变化，跟最后修改时间没有关系，ETag可以保证每一个资源是唯一的

If-None-Match的header会将上次返回的Etag发送给服务器，询问该资源的Etag是否有更新，有变动就会发送新的资源回来

而强制缓存不需要发送请求到服务端，根据请求头expires和cache-control判断是否命中强缓存

强制缓存与协商缓存的流程图如下所示：



### 会话状态

cookie，类型为「小型文本文件」，指某些网站为了辨别用户身份而储存在用户本地终端上的数据，通过响应头set-cookie决定

作为一段一般不超过 4KB 的小型文本数据，它由一个名称（Name）、一个值（Value）和其它几个用于控制 Cookie有效期、安全性、使用范围的可选属性组成

Cookie 主要用于以下三个方面：

* 会话状态管理（如用户登录状态、购物车、游戏分数或其它需要记录的信息）
* 个性化设置（如用户自定义设置、主题等）
* 浏览器行为跟踪（如跟踪分析用户行为等

## 参考文献

* https://zh.wikipedia.org/wiki/HTTP头字段
* https://github.com/amandakelake/blog/issues/41