**请求报头**

**1. Host (主机和端口号)**

Host：对应网址URL中的Web名称和端口号，用于指定被请求资源的Internet主机和端口号，通常属于URL的一部分。

**2. Connection (链接类型)**

Connection：表示客户端与服务连接类型

1. Client 发起一个包含 Connection:keep-alive 的请求，HTTP/1.1使用 keep-alive 为默认值。
2. Server收到请求后：
   * 如果 Server 支持 keep-alive，回复一个包含 Connection:keep-alive 的响应，不关闭连接；
   * 如果 Server 不支持 keep-alive，回复一个包含 Connection:close 的响应，关闭连接。
3. 如果client收到包含 Connection:keep-alive 的响应，向同一个连接发送下一个请求，直到一方主动关闭连接。

**keep-alive在很多情况下能够重用连接，减少资源消耗，缩短响应时间，比如当浏览器需要多个文件时(比如一个HTML文件和相关的图形文件)，不需要每次都去请求建立连接。**

**3. Upgrade-Insecure-Requests (升级为HTTPS请求)**

Upgrade-Insecure-Requests：升级不安全的请求，意思是会在加载 http 资源时自动替换成 https 请求，让浏览器不再显示https页面中的http请求警报。

***HTTPS 是以安全为目标的 HTTP 通道，所以在 HTTPS 承载的页面上不允许出现 HTTP 请求，一旦出现就是提示或报错。***

**4. User-Agent (浏览器名称)**

User-Agent：是客户浏览器的名称，以后会详细讲。

**5. Accept (传输文件类型)**

Accept：指浏览器或其他客户端可以接受的MIME（Multipurpose Internet Mail Extensions（多用途互联网邮件扩展））文件类型，服务器可以根据它判断并返回适当的文件格式。

**举例：**

Accept: \*/\*：表示什么都可以接收。

Accept：image/gif：表明客户端希望接受GIF图像格式的资源；

Accept：text/html：表明客户端希望接受html文本。

Accept: text/html, application/xhtml+xml;q=0.9, image/\*;q=0.8：表示浏览器支持的 MIME 类型分别是 html文本、xhtml和xml文档、所有的图像格式资源。

***q是权重系数，范围 0 =< q <= 1，q 值越大，请求越倾向于获得其“;”之前的类型表示的内容。若没有指定q值，则默认为1，按从左到右排序顺序；若被赋值为0，则用于表示浏览器不接受此内容类型。***

***Text：用于标准化地表示的文本信息，文本消息可以是多种字符集和或者多种格式的；Application：用于传输应用程序数据或者二进制数据。[详细请点击](http://blog.sina.com.cn/s/blog_866e403f010179f1.html" \t "_blank)***

**6. Referer (页面跳转处)**

Referer：表明产生请求的网页来自于哪个URL，用户是从该 Referer页面访问到当前请求的页面。这个属性可以用来跟踪Web请求来自哪个页面，是从什么网站来的等。

有时候遇到下载某网站图片，需要对应的referer，否则无法下载图片，那是因为人家做了防盗链，原理就是根据referer去判断是否是本网站的地址，如果不是，则拒绝，如果是，就可以下载；

**7. Accept-Encoding（文件编解码格式）**

Accept-Encoding：指出浏览器可以接受的编码方式。编码方式不同于文件格式，它是为了压缩文件并加速文件传递速度。浏览器在接收到Web响应之后先解码，然后再检查文件格式，许多情形下这可以减少大量的下载时间。

**举例：Accept-Encoding:gzip;q=1.0, identity; q=0.5, \*;q=0**

如果有多个Encoding同时匹配, 按照q值顺序排列，本例中按顺序支持 gzip, identity压缩编码，支持gzip的浏览器会返回经过gzip编码的HTML页面。 **如果请求消息中没有设置这个域服务器假定客户端对各种内容编码都可以接受。**

**8. Accept-Language（语言种类）**

Accept-Langeuage：指出浏览器可以接受的语言种类，如en或en-us指英语，zh或者zh-cn指中文，当服务器能够提供一种以上的语言版本时要用到。

**9. Accept-Charset（字符编码）**

Accept-Charset：指出浏览器可以接受的字符编码。

**举例：Accept-Charset:iso-8859-1,gb2312,utf-8**

* ISO8859-1：通常叫做Latin-1。Latin-1包括了书写所有西方欧洲语言不可缺少的附加字符，英文浏览器的默认值是ISO-8859-1.
* gb2312：标准简体中文字符集;
* utf-8：UNICODE 的一种变长字符编码，可以解决多种语言文本显示问题，从而实现应用国际化和本地化。

**如果在请求消息中没有设置这个域，缺省是任何字符集都可以接受。**

**10. Cookie （Cookie）**

Cookie：浏览器用这个属性向服务器发送Cookie。Cookie是在浏览器中寄存的小型数据体，它可以记载和服务器相关的用户信息，也可以用来实现会话功能，以后会详细讲。

**11. Content-Type (POST数据类型)**

Content-Type：POST请求里用来表示的内容类型。

**举例：Content-Type = Text/XML; charset=gb2312：**

指明该请求的消息体中包含的是纯文本的XML类型的数据，字符编码采用“gb2312”。

**常用的响应报头(了解)**

理论上所有的响应头信息都应该是回应请求头的。但是服务端为了效率，安全，还有其他方面的考虑，会添加相对应的响应头信息，从上图可以看到：

**1. Cache-Control：must-revalidate, no-cache, private。**

这个值告诉客户端，服务端不希望客户端缓存资源，在下次请求资源时，必须要从新请求服务器，不能从缓存副本中获取资源。

* Cache-Control是响应头中很重要的信息，当客户端请求头中包含Cache-Control:max-age=0请求，明确表示不会缓存服务器资源时,Cache-Control作为作为回应信息，通常会返回no-cache，意思就是说，"那就不缓存呗"。
* 当客户端在请求头中没有包含Cache-Control时，服务端往往会定,不同的资源不同的缓存策略，比如说oschina在缓存图片资源的策略就是Cache-Control：max-age=86400,这个意思是，从当前时间开始，在86400秒的时间内，客户端可以直接从缓存副本中读取资源，而不需要向服务器请求。

**2. 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。

**3. 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。

**4. 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。

**5. Date：Sun, 21 Sep 2016 06:18:21 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。

**6. 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。

**7. Pragma:no-cache**

这个含义与Cache-Control等同。

**8.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的信息。

**9. Transfer-Encoding：chunked**

这个响应头告诉客户端，服务器发送的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。

**10. Vary: Accept-Encoding**

告诉缓存服务器，缓存压缩文件和非压缩文件两个版本，现在这个字段用处并不大，因为现在的浏览器都是支持压缩的。