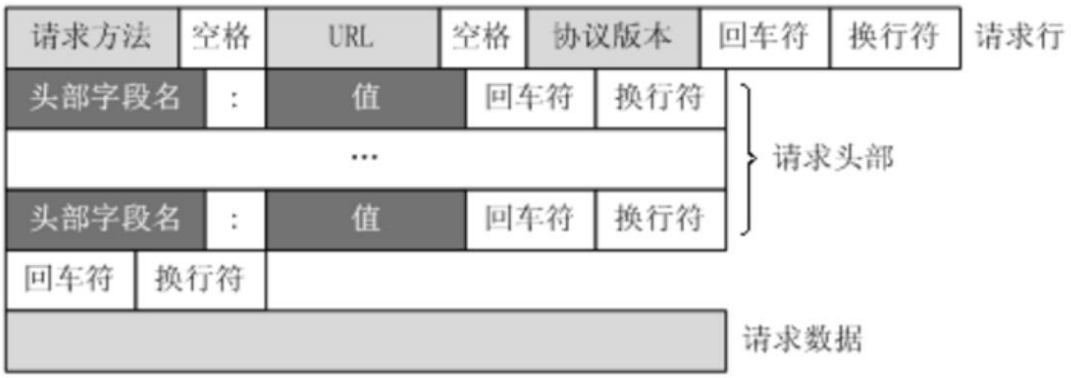
HTTP 消息结构

1、HTTP 请求消息

一个 HTTP 请求报文由请求行（request line）、请求头（header）、空行和请求数据（请求体）4 个部分组成，下图给出了请求报文的一般格式。



请求行：请求行由请求方法字段、URL 字段和 HTTP 协议版本字段 3 个字段组成，它们用空格分隔。例如，GET /index.html HTTP/1.1。
根据 HTTP 标准，HTTP 请求可以使用多种请求方法。
HTTP1.0 定义了三种请求方法： GET, POST 和 HEAD 方法。
HTTP1.1 新增了五种请求方法：OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。

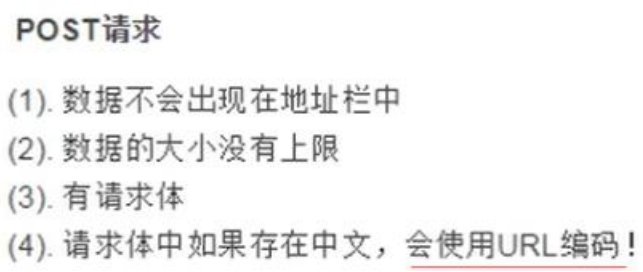
请求头：请求头部由关键字/值对组成，每行一对，关键字和值用英文冒号“:”分隔。请求头部通知服务器有关于客户端请求的信息，典型的请求头有：
User-Agent：产生请求的浏览器类型。

Accept：客户端可识别的内容类型列表。
Host：请求的主机名，允许多个域名同处一个 IP 地址，即虚拟主机。

空行：最后一个请求头之后是一个空行，发送回车符和换行符，通知服务器以下不再有请求头。
请求数据（请求体）：请求数据不在 GET 方法中使用，而是在 POST 方法中使用。POST 方法适用于需要客户填写表单的场合。与请求数据相关的最常使用的请求头是 Content-Type 和 Content-Length。

这里要特别说明一下 GET 和 POST 的区别
（1）GET
当客户端要从服务器中读取文档时，当点击网页上的链接或者通过在浏览器的地址栏输入网址来浏览网页的，使用的都是 GET 方式。GET 方法要求服务器将 URL 定位的资源放在响应报文的数据部分，回送给客户端。使用 GET 方法时，请求参数和对应的值附加在 URL 后面，利用一个问号（“?”）代表 URL 的结尾与请求参数的开始，传递参数长度受限制。例如，/index.jsp?id=100&op=bind,这样通过 GET 方式传递的数据直接表示在地址中，所以我们可以把请求结果以链接的形式发送给好友。
所以 GET 方式的请求消息一般不包含”请求数据”部分，请求数据以地址的形式表现在请求行里了。地址中”?”之后的部分就是通过 GET 发送的请求数据，我们可以在地址栏中清楚的看到，各个数据之间用”&”符号隔开。显然，这种方式不适合传送私密数据。另外，由于不同的浏览器对地址的字符限制也有所不同，一般最多只能识别 1024 个字符，所以如果需要传送大量数据的时候，也不适合使用 GET 方式。（APP 默认是 GET 方式）

（2）POST
使用 POST 方法可以允许客户端给服务器提供信息较多。POST 方法将请求参数封装在 HTTP 请求数据中，以名称/值的形式出现，可以传输大量数据，这样 POST 方式对传送的数据大小没有限制，而且也不会显示在 URL 中。POST 方式请求行中不包含数据字符串，这些数据保存在”请求数据”部分，各数据之间也是使用”&”符号隔开。POST 方式大多用于页面的表单中。因为 POST 也能完成 GET 的功能，因此多数人在设计表单的时候一律都使用 POST 方式。



例子：
1、GET 方式

```
//请求首行
GET /hello/index.jsp HTTP/1.1
//请求头信息，因为GET请求没有正文
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101 Firefox/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Cookie: JSESSIONID=369766FDF6220F7803433C0B2DE36D98
//空行
//因为GET没有正文，所以下面为空
```

2、POST 方式

```
// 请求首行
POST /hello/index.jsp HTTP/1.1
//请求头信息
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101 Firefox/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer: http://localhost/hello/index.jsp
Cookie: JSESSIONID=369766FDF6220F7803433C0B2DE36D98
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
// 这里是空行
//POST有请求正文
username=hello
```

3、HTTP 响应消息

HTTP 响应也由 4 个部分组成，分别是：状态行、响应头、空行、响应正文（响应体）。

正如你所见，在响应中唯一真正的区别在于第一行中用状态信息代替了请求信息。状态行（status line）通过提供一个状态码来说明所请求的资源情况。

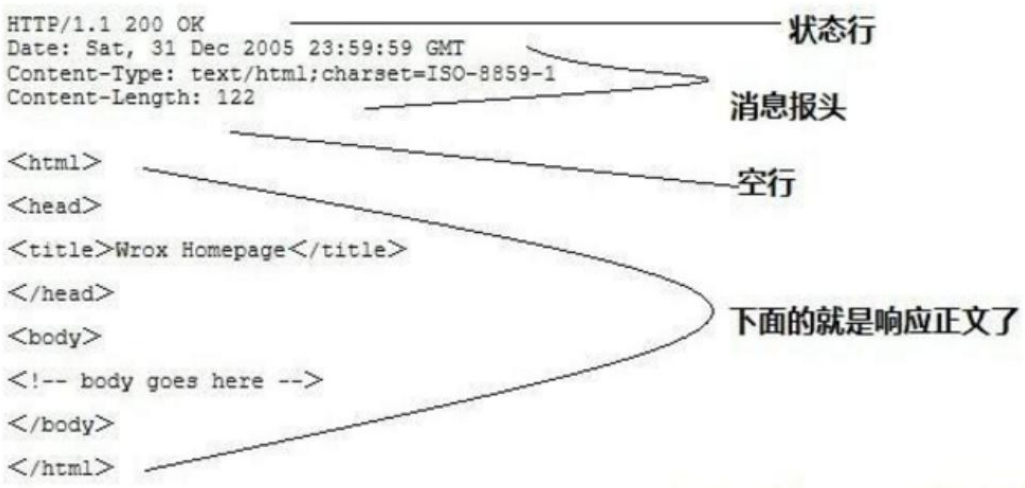
状态行格式如下：

HTTP-Version　　Status-Code　　Reason-Phrase

其中，HTTP-Version 表示服务器 HTTP 协议的版本；

Status-Code 表示服务器发回的响应状态代码；

Reason-Phrase 表示状态代码的原因短语。



第一部分：状态行，由 HTTP 协议版本号， 状态码， 状态消息 三部分组成。
第一行为状态行，（HTTP/1.1）表明 HTTP 版本为 1.1 版本，状态码为 200，状态消息为（ok）
第二部分：响应头，用来说明客户端要使用的一些附加信息
Date:生成响应的日期和时间；Content-Type:指定了 MIME 类型的 HTML(text/html)，编码类型是 UTF-8
第三部分：空行，消息报头后面的空行是必须的
第四部分：响应正文，服务器返回给客户端的文本信息。
空行后面的 html 部分为响应正文。

请求头属性：

Header	解释	示例
Accept	指定客户端能够接收的内容类型	Accept: text/plain, text/html
Accept-Charset	浏览器可以接受的字符编码集。	Accept-Charset: iso-8859-5
Accept-Encoding	指定浏览器可以支持的web服务器返回内容压缩编码类型。	Accept-Encoding: compress, gzip
Accept-Language	浏览器可接受的语言	Accept-Language: en,zh
Accept-Ranges	可以请求网页实体的一个或者多个子范围字段	Accept-Ranges: bytes
Authorization	HTTP授权的授权证书	Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Cache-Control	指定请求和响应遵循的缓存机制	Cache-Control: no-cache
Connection	表示是否需要持久连接。（HTTP 1.1默认进行持久连接）	Connection: close
Cookie	HTTP请求发送时，会把保存在该请求域名下的所有cookie值一起发送给web服务器。	Cookie: $Version=1; Skin=new;
Content-Length	请求的内容长度	Content-Length: 348
Content-Type	请求的与实体对应的MIME信息	Content-Type: application/x-www-form-urlencoded
Date	请求发送的日期和时间	Date: Tue, 15 Nov 2010 08:12:31 GMT
Expect	请求的特定的服务器行为	Expect: 100-continue
From	发出请求的用户的Email	From: user@email.com
Host	指定请求的服务器的域名和端口号	Host: www.zcmhi.com
If-Match	只有请求内容与实体相匹配才有效	If-Match: “737060cd8c284d8af7ad3082f209582d”
If-Modified-Since	如果请求的部分在指定时间之后被修改则请求成功，未被修改则返回304代码	If-Modified-Since: Sat, 29 Oct 2010 19:43:31 GMT
If-None-Match	如果内容未改变返回304代码，参数为服务器先前发送的Etag，与服务器回应的Etag比较判断是否改变	If-None-Match: “737060cd8c284d8af7ad3082f209582d”
If-Range	如果实体未改变，服务器发送客户端丢失的部分，否则发送整个实体。参数也为Etag	If-Range: “737060cd8c284d8af7ad3082f209582d”
If-Unmodified-Since	只在实体在指定时间之后未被修改才请求成功	If-Unmodified-Since: Sat, 29 Oct 2010 19:43:31 GMT
Max-Forwards	限制信息通过代理和网关传送的时间	Max-Forwards: 10
Pragma	用来包含实现特定的指令	Pragma: no-cache
Proxy-Authorization	连接到代理的授权证书	Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Range	只请求实体的一部分，指定范围	Range: bytes=500-999
Referer	先前网页的地址，当前请求网页紧随其后,即来路	Referer: http://www.zcmhi.com/archives/71.html
TE	客户端愿意接受的传输编码，并通知服务器接受接受尾加头信息	TE: trailers,deflate;q=0.5
Upgrade	向服务器指定某种传输协议以便服务器进行转换（如果支持）	Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11
User-Agent	User-Agent的内容包含发出请求的用户信息	User-Agent: Mozilla/5.0 (Linux; X11)
Via	通知中间网关或代理服务器地址，通信协议	Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)
Warning	关于消息实体的警告信息	Warn: 199 Miscellaneous warning

响应头属性：

Header	解释	示例
Accept-Ranges	表明服务器是否支持指定范围请求及哪种类型的分段请求	Accept-Ranges: bytes
Age	从原始服务器到代理缓存形成的估算时间（以秒计，非负）	Age: 12
Allow	对某网络资源的有效的请求行为，不允许则返回405	Allow: GET, HEAD
Cache-Control	告诉所有的缓存机制是否可以缓存及哪种类型	Cache-Control: no-cache
Content-Encoding	web服务器支持的返回内容压缩编码类型。	Content-Encoding: gzip
Content-Language	响应体的语言	Content-Language: en,zh
Content-Length	响应体的长度	Content-Length: 348
Content-Location	请求资源可替代的备用的另一地址	Content-Location: /index.htm
Content-MD5	返回资源的MD5校验值	Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ==
Content-Range	在整个返回体中本部分的字节位置	Content-Range: bytes 21010-47021/47022
Content-Type	返回内容的MIME类型	Content-Type: text/html; charset=utf-8
Date	原始服务器消息发出的时间	Date: Tue, 15 Nov 2010 08:12:31 GMT
ETag	请求变量的实体标签的当前值	ETag: “737060cd8c284d8af7ad3082f209582d”
Expires	响应过期的日期和时间	Expires: Thu, 01 Dec 2010 16:00:00 GMT
Last-Modified	请求资源的最后修改时间	Last-Modified: Tue, 15 Nov 2010 12:45:26 GMT
Location	用来重定向接收方到非请求URL的位置来完成请求或标识新的资源	Location: http://www.zcmhi.com/archives/94.html
Pragma	包括实现特定的指令，它可应用到响应链上的任何接收方	Pragma: no-cache
Proxy-Authenticate	它指出认证方案和可应用到代理的该URL上的参数	Proxy-Authenticate: Basic
refresh	应用于重定向或一个新的资源被创造，在5秒之后重定向（由网景提出，被大部分浏览器支持）	Refresh: 5; url=http://www.atool.org/httptest.php
Retry-After	如果实体暂时不可取，通知客户端在指定时间之后再次尝试	Retry-After: 120
Server	web服务器软件名称	Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Set-Cookie	设置Http Cookie	Set-Cookie: UserID=JohnDoe; Max-Age=3600; Version=1
Trailer	指出头域在分块传输编码的尾部存在	Trailer: Max-Forwards
Transfer-Encoding	文件传输编码	Transfer-Encoding:chunked
Vary	告诉下游代理是使用缓存响应还是从原始服务器请求	Vary: *
Via	告知代理客户端响应是通过哪里发送的	Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)
Warning	警告实体可能存在的问题	Warning: 199 Miscellaneous warning
WWW-Authenticate	表明客户端请求实体应该使用的授权方案	WWW-Authenticate: Basic

参考链接：

https://blog.csdn.net/xiaochengyihe/article/details/80910913

https://www.cnblogs.com/lauhp/p/8979393.html

https://www.cnblogs.com/weibanggang/p/9454581.html