目录

HTTP 数据包详解

1. HTTP 报文格式

HTTP 由请求和响应两部分组成，所以对应的也有两种报文格式。下面分别介绍 HTTP 请求报文格式和 HTTP 响应报文格式。

HTTP 请求报文格式

以上表格中，第 1 行为“请求行”，第 2、3、4 行为“请求头部”，第 5 行为空行，第 6 行为“请求正文”。分别介绍这 4 部分：

1.请求行：由 3 部分组成，分别为：请求方法、URL（见备注 1）以及协议版本，之间由空格分隔，请求方法包括 GET、POST 等。协议版本的格式为：HTTP/主版本号.次版本号，常用的有 HTTP/1.0 和 HTTP/1.1。

2.请求头部包含很多客户端环境以及请求正文的有用信息。请求头部由“关键字：值”对组成，每行一堆，关键字和值之间使用英文“：”分隔。

3.空行，这一行非常重要，必不可少。表示请求头部结束，下面就是请求正文。

4.请求正文：可选部分，比如 GET 请求就没有请求正文；POST 比如以提交表单数据方式为请求正文。

HTTP 响应报文格式：

以上表格中，第 1 行为“状态行”，第 2、3、4 行为“响应头部”，第 5 行为空行，第 6 行为“响应正文”。下面分别介绍这 4 部分：

(1)状态行由由 3 部分组成，分别为：协议版本，状态码，状态码描述，之间由空格分隔。状态代码为 3 位数字，200~299 的状态码表示成功，300~399 的状态码指资源重定向，

400~499 的状态码指客户端请求出错，500~599 的状态码指服务端出错（HTTP/1.1 向协议中引入了信息性状态码，范围为 100~199）。这里列举几个常见的:

状态码	说明
200	响应成功
400	客户端请求有语法错误，不能被服务器识别
404	请求资源不存在
500	服务器内部错误

(2)响应头部与请求头部类似，也包含了很多有用的信息。
(3)空行，这一行非常重要，必不可少。表示响应头部结束
(4)响应正文，服务器返回的文档，最常见的为 HTML 网页。

2. HTTP 的头域

在 HTTP 的请求消息和应答消息中，都包含头域。头域分为 4 种，其中请求头域和应答头域分别只在请求消息和应答消息中出现，通用头域和实体头域在两种消息中都可以出现，但实体头域只有当消息中包含了实体数据时才会出现。下面分别介绍这 4 种头域中的域名城和功能。

HTTP 请求头域

Header	解释
Accept	指定客户端能够接收的内容类型
Accept-Charset	浏览器可以接受的字符编码集。
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-Language	浏览器可接受的语言
Accept-Ranges	可以请求网页实体的一个或者多个子范围字段
Authorization	HTTP 授权的授权证书
Cache-Control	指定请求和响应遵循的缓存机制
Connection	表示是否需要持久连接。（HTTP 1.1 默认进行持久连接）
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。
Content-Length	请求的内容长度
Content-Type	请求的与实体对应的 MIME 信息
Date	请求发送的日期和时间
Expect	请求的特定的服务器行为
From	发出请求的用户的 Email
Host	指定请求的服务器的域名和端口号
If-Match	只有请求内容与实体相匹配才有效
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-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-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-Unmodified-Since	只在实体在指定时间之后未被修改才请求成功

Max-Forwards	限制信息通过代理和网关传送的时间
Pragma	用来包含实现特定的指令
Proxy-Authorization	连接到代理的授权证书
Range	只请求实体的一部分，指定范围
Referer	先前网页的地址，当前请求网页紧随其后,即来路
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
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）
User-Agent	User-Agent 的内容包含发出请求的用户信息
Via	通知中间网关或代理服务器地址，通信协议
Warning	关于消息实体的警告信息
Accept	指定客户端能够接收的内容类型

应答头域只在应答消息中出现，是 Web 服务器向浏览器提供的一些状态和要求。如下

HTTP 应答头域

Header	解释
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
Age	从原始服务器到代理缓存形成的估算时间（以秒计，非负）
Allow	对某网络资源的有效的请求行为，不允许则返回 405
Cache-Control	告诉所有的缓存机制是否可以缓存及哪种类型
Content-Encoding	web 服务器支持的返回内容压缩编码类型。
Content-Language	响应体的语言
Content-Length	响应体的长度
Content-Location	请求资源可替代的备用的另一地址
Content-MD5	返回资源的 MD5 校验值
Content-Range	在整个返回体中本部分的字节位置
Content-Type	返回内容的 MIME 类型
Date	原始服务器消息发出的时间
ETag	请求变量的实体标签的当前值
Expires	响应过期的日期和时间
Last-Modified	请求资源的最后修改时间
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
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
Proxy-Authenticate	它指出认证方案和可应用到代理的该 URL 上的参数
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
Server	web 服务器软件名称
Set-Cookie	设置 Http Cookie

Trailer	指出头域在分块传输编码的尾部存在
Transfer-Encoding	文件传输编码
Vary	告诉下游代理是使用缓存响应还是从原始服务器请求
Via	告知代理客户端响应是通过哪里发送的
Warning	警告实体可能存在的问题
WWW-Authenticate	表明客户端请求实体应该使用的授权方案
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）

通用头域既可以用在请求消息中，也可以用在应答消息。

HTTP 通用头域

Header	解释
Cache-Control	Cache-Control 指定请求和响应遵循的缓存机制，可以附带很多的规定值。
Connection	表示是否需要持久连接
Date	表示消息发送的时间
Pragma	Pragma 头域用来包含实现特定的指令，最常用的是 Pragma:no-cache，用于定义页面缓存
Trailer	表示以 Chunked 编码传输的实体数据的尾部存在哪些头域
Transfer-Encoding	WEB 服务器表明自己对本响应消息体（不是消息体里面的对象）作了怎样的编码，比如是否分块（chunked），例如：Transfer-Encoding: chunked
Upgrade	它可以指定另一种可能完全不同的协议，如 HTTP/1.1 客户端可以向服务器发送一条 HTTP/1.0 请求，其中包含值为"HTTP/1.1"的 Update 头部，这样客户端就可以测试一下服务器是否也使用 HTTP/1.1 了。
Via	列出从客户端到 OCS 或者相反方向的响应经过了哪些代理服务器，他们用什么协议（和版本）发送的请求。
Warning	用于警告应用到实体数据上的缓存操作或转换可能缺少语义透明度。

只有在请求和应答消息中包含实体数据时，才需要实体头域。请求消息中的实体数据是一些由浏览器向 web 服务器提交的数据，如在浏览器中采用 POST 方式提交表单时，浏览器就要把表单中的数据封装在请求消息的实体数据部分。应答消息中的实体数据是 web 服务器发给浏览器的媒体数据，如网页，图片和文档等。实体头域说明了实体数据的一些属性。如下表

HTTP 实体头域

Header	解释
Allow	列出由请求 URI 标识的资源所支持的方法集
Content-Encoding	说明实体数据是如何编码的
Content-Language	说明实体数据所采用的自然语言
Content-Length	说明实体数据的长度
Content-Location	说明实体数据的资源位置
Content-MD5	给出实体数据的 MD5 值，用于保证实体数据的完整性
Content-Range	用于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
Content-Type	用于向接收方指示实体的介质类型，指定 HEAD 方法送到接收方的实体介质类型，或 GET 方法发送的请求介质类型
Expires	指定实体数据的有效期
Last-Modified	指定服务器上保存内容的最后修订时间。

3.Http 请求数据包

（1）请求行

包含三个内容 method + request-URI + http-version。

method 包含有 post , get, head,delete, put, connect, options, patch, propfind, propatch, mkcol, copy, move, lock, unlock, trace, head。

get	通过请求URI获得资源
post	用于添加新的资源，用于表单提交
put	用于修改某个内容
delete	删除某个内容
connect	用于代理进行传输例如SSL
options	询问可以执行那些方法
patch	部分文档更该
propfind	查看属性
proppatch	设置属性
mkcol	创建集合
copy	拷贝
move	移动
lock	加锁
unlock	解锁
trace	用于远程诊断服务器
head	类似于get，用于检查对象是否存在用于得到元数据

主要介绍 get 方法和 post 方法

get 方法：

是在 url 中说明情请求的资源，

比如 https://www.baidu.com/con?from=self?_t=1466609839126 其中？后的数据就是请求的数据，并且连接用&，get 方法也可以提交表单数据，但是提交的数据在 url 中，其他人可以通过查看历史记录中的 url 来获取你提交的数据，这样很不安全。

post 方法：

传输数据不在 url 中,而在数据段中出现,并且请求头多了 Content-Type 和 Content-Length,post 提交表单数据的时候比 get 方法更安全。

post 方法提交表单和 get 方法提交表单相比较：

get 明文传输，信息附加在 url 上面，get 明文传输，post 更加安全

get 传输有大小限制,应该是 3k，post 需要制定传输类型

get 多用于获取数据，根据 get 变量的不同调用不同的数据，post 多用于提交数据，提交用户输入的数据

get 方法和 post 方法的区别：

1>Get 是向服务器发索取数据的一种请求，而 Post 是向服务器提交数据的一种请求。

2>Get 是获取信息，而不是修改信息，类似数据库查询功能一样，数据不会被修改。

3>Get 请求的参数会跟在 url 后进行传递，请求的数据会附在 URL 之后，以?分割 URL 和传输数据，参数之间以&相连,%XX 中的 XX 为该符号以 16 进制表示的 ASCII，如果数据是英文字母/数字，原样发送，如果是空格，转换为+，如果是中文/其他字符，则直接把字符串用 BASE64 加密。

4>Get 传输的数据有大小限制，因为 GET 是通过 URL 提交数据，那么 GET 可提交的数据量就跟 URL 的长度有直接关系了，不同的浏览器对 URL 的长度的限制是不同的。

5>GET 请求的数据会被浏览器缓存起来，用户名和密码将明文出现在 URL 上，其他人可以查到历史浏览记录，数据不太安全。在服务器端，用 Request.QueryString 来获取 Get 方式提交来的数据。

6>Post 请求则作为 http 消息的实际内容发送给 web 服务器，数据放置在请求体中，Post 没有限制提交的数据。Post 比 Get 安全，当数据是中文或者不敏感的数据，则用 get，因为使用 get，参数会显示在地址，对于敏感数据和不是中文字符的数据，则用 post。

7>POST 表示可能修改变服务器上的资源的请求，在服务器端，用 Post 方式提交的数据只能用 Request.Form 来获取。

（2）请求头

Accept：指浏览器或其他客户可以接爱的 MIME 文件格式。Servlet 可以根据它判断并返回适当的文件格式。

User-Agent：是客户浏览器名称。

Host：对应网址 URL 中的 Web 名称和端口号。

Accept-Langeuage：指出浏览器可以接受的语言种类，如 en 或 en-us，指英语。

connection：用来告诉服务器是否可以维持固定的 HTTP 连接。http 是无连接的，HTTP/1.1 使用 Keep-Alive 为默认值，这样，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)，不需要每次都建立连接

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。

Referer：表明产生请求的网页 URL。如比从网页/icconcept/index.jsp 中点击一个链接到网页/icwork/search，在向服务器发送的 GET/icwork/search 中的请求中，Referer 是 http://hostname:8080/icconcept/index.jsp。这个属性可以用来跟踪 Web 请求是从什么网站来的。

User-Agent：是客户浏览器名称。

Content-Type：用来表名 request 的内容类型。可以用 HttpServletRequest 的 getContentType()方法取得。

Accept-Charset：指出浏览器可以接受的字符编码。英文浏览器的默认值是 ISO-8859-1.

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。

get 方法请求头例如：

Accept:image/webp,image/*,*/*;q=0.8

Accept-Encoding:gzip, deflate, sdch, br

Accept-Language:en-US,en;q=0.8

Connection:keep-alive

Cookie:PSTM=1466499789; BAIDUID=D3A617EE01FFA9DB9B7E3E5F0D3A01EE:FG=1; BIDUPSID=4AA34EC11075CB66B8BC9792DD422B6F; BDUSS=VCc1M0cVQtYnFGfmxTUW5kVTUydnBZUmhiWFRXbnRlMnpIdWV2ODVxNHZ1WkZYQVFBQUFBJCQAAAAAAAAAAAEAAADkEA1ZtPO3rMfRt6zH0cfRAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAC8salcvLGpXdz; BD_HOME=1; BD_UPN=123353; BDRCVFR[feWj1Vr5u3D]=I67x6TjHwwYf0; BD_CK_SAM=1; H_PS_PSSID=19292_18286_1458_20318_18241_20369_17942_20388_19690_20417_18560_17001_15560_12277_20253; BDSVRTM=0

Host:www.baidu.com

Referer:https://www.baidu.com/s?wd=http%20%E8%AF%B7%E6%B1%82%E6%95%B0%E6%8D%AE%E7%9A%84%E6%95%B0%E6%8D%AE%E5%8C%85%E6%A0%BC%E5%BC%8F&rsv_spt=1&rsv_iqid=0x9b746a8000022af9&issp=1&f=8&rsv_bp=1&rsv_idx=2&ie=utf-8&rqlang=cn&tn=baiduhome_pg&rsv_enter=1&oq=http%20%E8%AF%B7%E6%B1%82%E6%96%B9%E5%BC%8Fpost%20url%E6%A0%BC%E5%BC%8F&rsv_t=59fb7cEn5xgK8JFpqQ7F7coy6k6dn5sGpEMj1cDM4oMoy0TGArJ2l3fxOqy6F9lXoqoi&inputT=7936&rsv_pq=ca5859d100027005&rsv_sug3=73&rsv_sug1=12&rsv_sug7=100&rsv_sug2=0&rsv_sug4=32020

User-Agent:Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36

post 方法的请求头

Accept:*/*

Accept-Encoding:gzip, deflate, br

Accept-Language:en-US,en;q=0.8

Authorization:Basic WkEtMTE0MjcyNjAyMDY=

Connection:keep-alive

Content-Length:666

Content-Type:application/json

Host:zhihu-web-analytics.zhihu.com

Origin:http://www.zhihu.com

Referer:http://www.zhihu.com/question/41690822

User-Agent:Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36

Request Payload

view source

相比之下多了 content-Type 和 Content-Length

1. Content-Type:表示的是请求报文体的 MIME 类型 ，注：GET 的请求消息体是空的 所以不需要指定消息体的 MIME 类型
2. Content-Length:表示的是 post 的数据的长度

例如请求数据：

GET/sample.jspHTTP/1.1

Accept:image/gif.image/jpeg,*/*

Accept-Language:zh-cn

Connection:Keep-Alive

Host:localhost

User-Agent:Mozila/4.0(compatible;MSIE5.01;Window NT5.0)

Accept-Encoding:gzip,deflate

第一行为 http 请求行，包含方法，URI 和 http 版本

1-7 为请求头，包含浏览器，主机，接受的编码方式和压缩方式；第 8 行表示一个空行 表示请求头结束 这个空行是必须的；

数据体

第 9 行是数据体，比如是需要查询的信息。

4.http 响应数据包

状态行 + 响应头 + 响应正文

（1）状态行是由：HTTP-Version+Status-Code+Reason-Phrase

比如：HTTP/1.1 200 ok

分别表示 http 版本 + 状态码 + 状态代码的文本描述

状态码：

1xx	指示信息–表示请求已接收，继续处理
2xx	成功–表示请求已被成功接收、理解、接受
3xx	重定向–要完成请求必须进行更进一步的操作。
4xx	客户端错误–请求有语法错误或请求无法实现。
5xx	服务器端错误–服务器未能实现合法的请求。

（2）响应头：包含服务器类型，日期，长度，内容类型等

Server:Apache Tomcat/5.0.12

Date:Mon,6Oct2003 13:13:33 GMT

Content-Type:text/html

Last-Moified:Mon,6 Oct 2003 13:23:42 GMT

Content-Length:112

（3）响应正文响应正文就是服务器返回的 HTML 页面