报文流

HTTP报文在客户端、服务器和代理之间流动。“流入”、“流出”、“上游”、“下游”这些术语用来描述报文方向。

报文流入源端服务器

流入：流向服务器
流出：流向用户Agent代理

报文向下游流动

所有报文都会向下游流动。对请求报文来说，客户端在服务器的上游；而对于响应报文来说，服务器在客户端的上游。

报文的组成部分

三个部分：对报文进行描述的起始行、包含属性的首部块、包含数据的主体（可选）。
起始行和首部是由行分隔的ASCII文本。每行以一个由两个字符组成的行终止符序列作为结束，包括一个回车符和一个换行符。这个行终止序列可以写作CRLF。

报文的语法

请求报文的格式：

```
<method> <request-URL> <version>
<headers>

<entity-body>
```

响应报文的格式：

```
<version> <status> <reason-phrase>
<headers>

<entity-body>
```

PS：一组HTTP首部总是应该以一个空行（仅有CRLF）结束，甚至没有首部和实体的主体部分也应该如此。

起始行

请求报文：要做些什么；响应报文：发生了什么。

方法

请求的起始行以方法作为开始

状态码

状态码告诉客户端发生了什么事情，位于响应行的起始行中。

原因短语

响应起始行最后。为状态码提供了文本形式的解释。

版本号

说明了应用程序支持的最高HTTP版本。

PS：版本号不会当做小数来处理，每个数字当作一个单独的数字来处理。如HTTP/2.22比HTTP/2.3的版本高。

首部

名/值对的列表（后面具体介绍）

实体的主体部分

图片、视频、HTML文档等等等等。

版本0.9的报文

请求中只有方法和请求URL，响应中只包含实体。

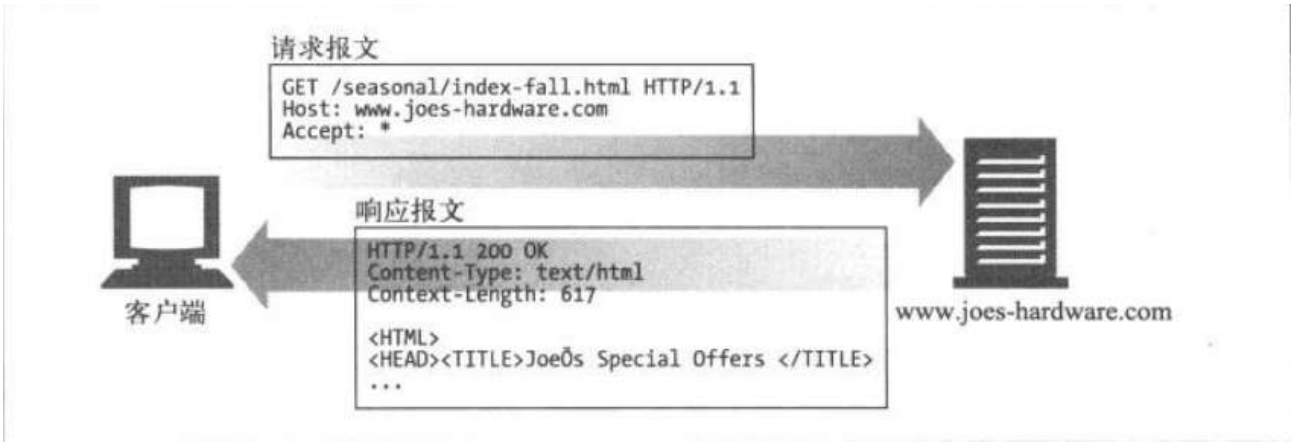
方法

安全方法

GET和HEAD都认为是安全方法，不会产生动作，意味着HTTP请求不会再服务器上产生什么结果。

GET方法

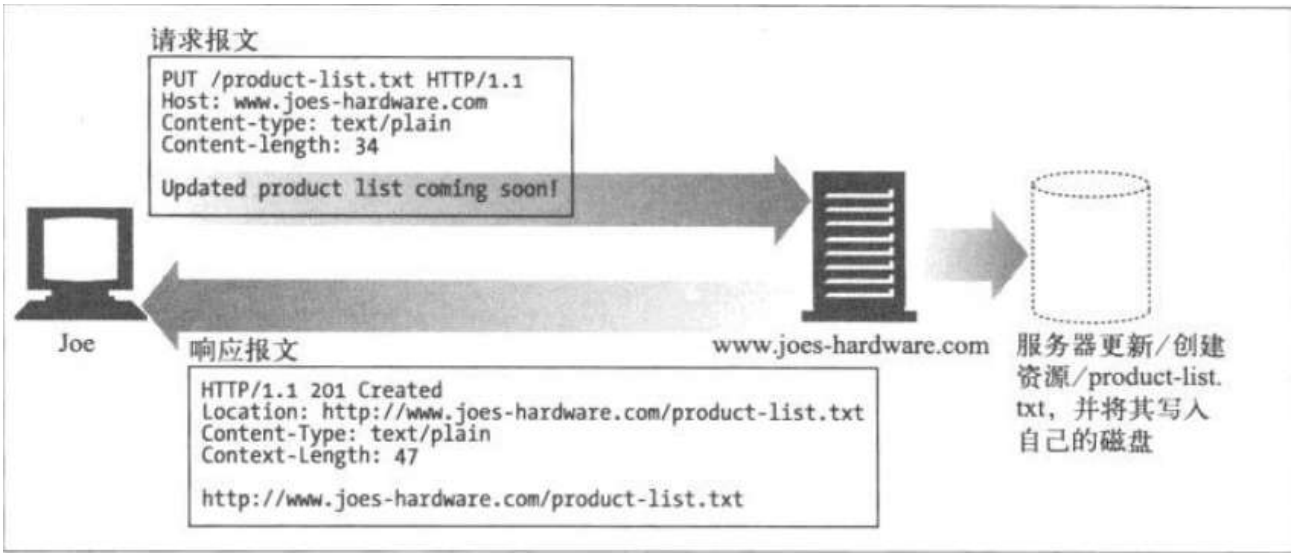
用于请求服务器发送某个资源。



HEAD

与GET方法类似，但服务器在响应中只返回首部，不会返回实体的主体部分。

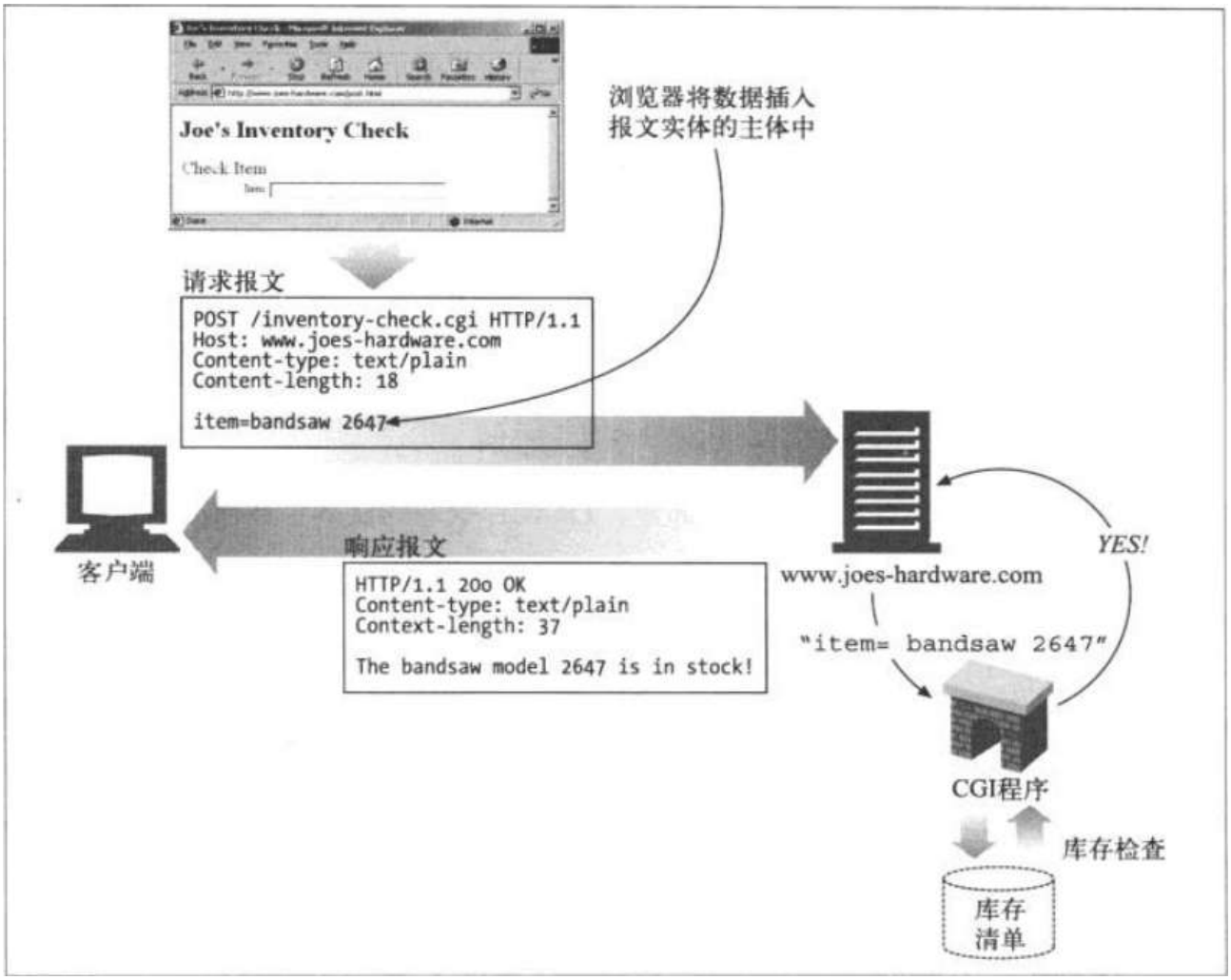
- 在不获取资源的情况下了解资源的情况
- 查看响应中的状态码看看某个对象是否存在
- 测试资源是否被修改了



PUT方法会向服务器写入文档。很多Web服务器要求在执行PUT之前用密码登录。

POST

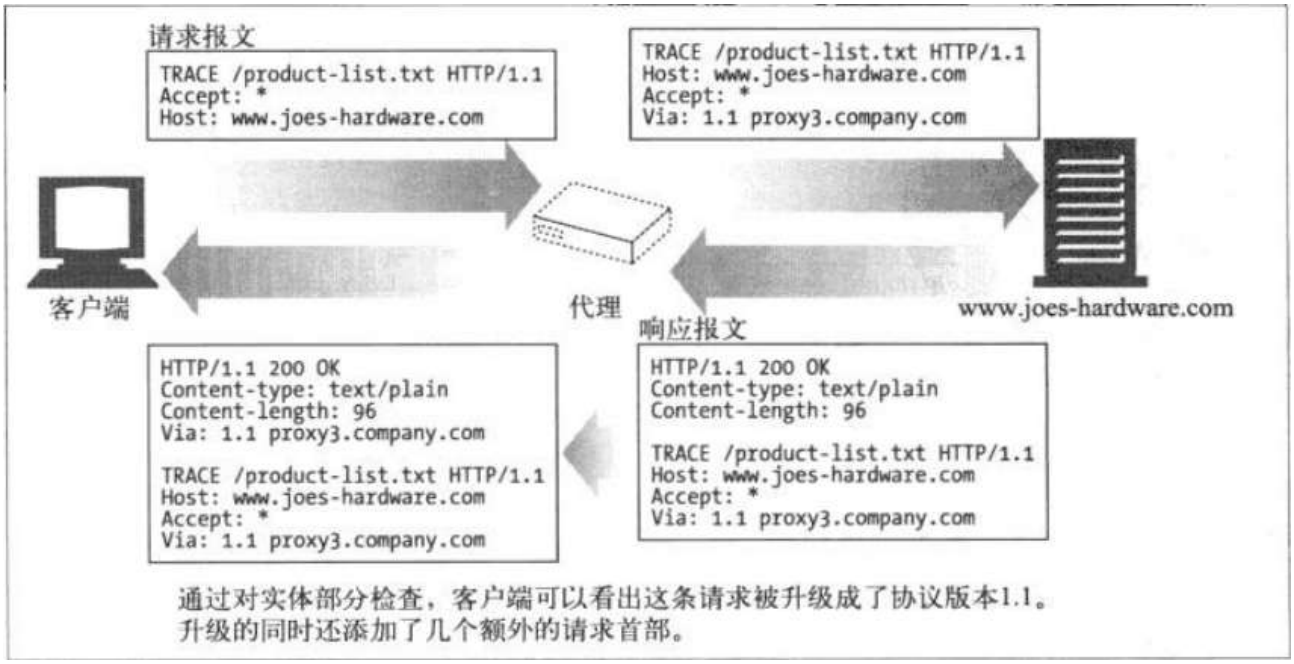
起初是用来向服务器输入数据的。通常会用它来支持HTML的表单。



TRACE

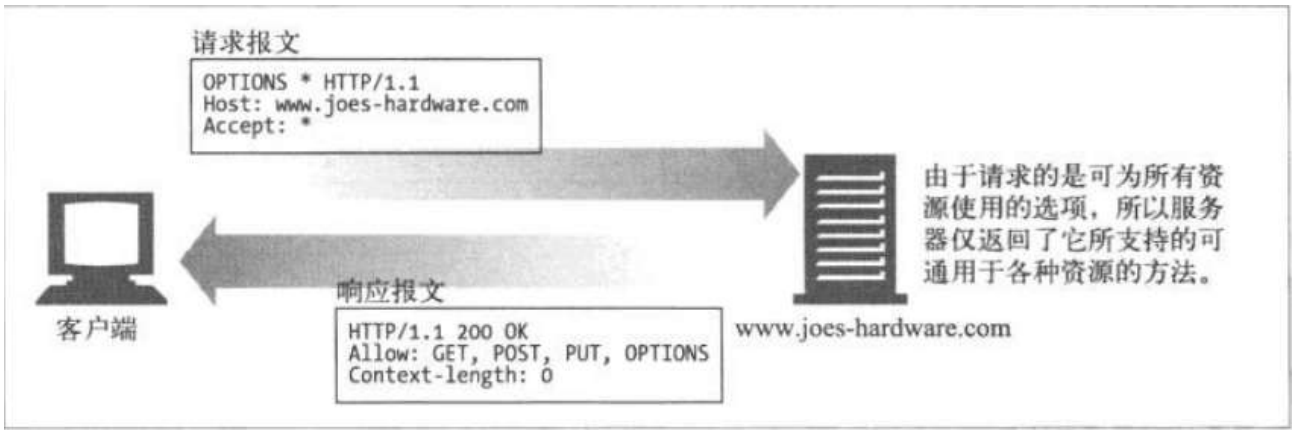
客户端在发送请求时，可能要穿过防火墙、代理、网关或其他一些应用程序，每个中间节点都可能会修改原始的HTTP请求，TRACE方法允许客户端在最终把请求发送给服务器时看看它变成了什么样子。

缺点：它假定中间应用程序对不同类型请求的处理是相同的。然而很多应用程序会根据方法的不同做出不同的事情，TRACE不提供区分这些方法的机制。



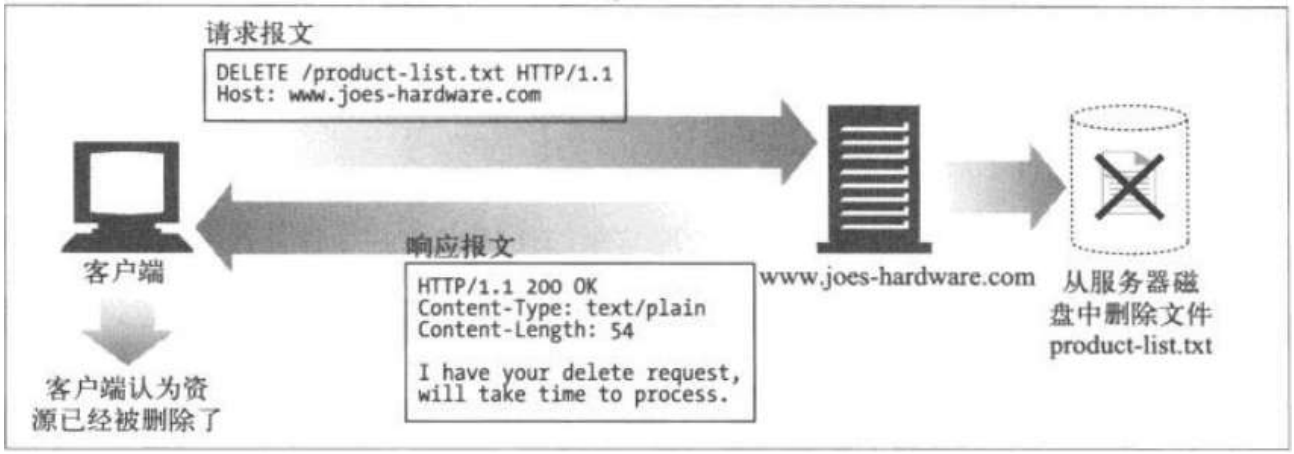
OPTIONS

请求Web服务器告知其支持的各种功能。



DELETE

请服务器删除URL指定的资源。但是客户端无法保证删除操作一定被执行。



扩展方法

没有在HTTP/1.1规范中定义的方法。“对所发送的内容要求严一点，对所接收的内容宽容一些”。

方　法	描　述
LOCK	允许用户“锁定”资源——比如，可以在编辑某个资源的时候将其锁定，以防别人同时对其进行修改
MKCOL	允许用户创建资源
COPY	便于在服务器上复制资源
MOVE	在服务器上移动资源

状态码

100~199——信息状态码

100：收到了请求的初始部分，晴客户端继续。

200~299——成功状态码

200：请求没问题
204：响应报文中包含若干首部和一个状态行，但没有实体的主体部分
206：成功执行了一个部分或范围请求

300~399——重定向状态码

304：Not Modified

400~499——客户端错误状态码

400：客户端发送了一个错误的请求
404：Not Found

500~599——服务器错误状态码

500：服务遇到一个妨碍它为请求提供服务的错误
502：作为代理或网关使用的服务器收到了一条伪响应

首部

- 通用首部
  客户端和服务器都可以使用的通用首部
- 请求首部
- 响应首部
- 实体首部
- 扩展首部

通用首部

首　部	描　述
Connection	允许客户端和服务器指定与请求 / 响应连接有关的选项
Date[5]	提供日期和时间标志，说明报文是什么时间创建的
MIME-Version	给出了发送端使用的 MIME 版本
Trailer	如果报文采用了分块传输编码（chunked transfer encoding）方式，就可以用这个首部列出位于报文拖挂（trailer）部分的首部集合[6]
Transfer-Encoding	告知接收端为了保证报文的可靠传输，对报文采用了什么编码方式
Update	给出了发送端可能想要“升级”使用的新版本或协议
Via	显示了报文经过的中间节点（代理、网关）

通用缓存首部

首　部	描　述
Cache-Control	用于随报文传送缓存指示
Pragma[7]	另一种随报文传送指示的方式，但并不专用于缓存

请求首部

首　部	描　述
Client-IP[8]	提供了运行客户端的机器的 IP 地址
From	提供了客户端用户的 E-mail 地址[9]
Host	给出了接收请求的服务器的主机名和端口号
Referer	提供了包含当前请求 URI 的文档的 URL
UA-Color	提供了与客户端显示器的显示颜色有关的信息
UA-CPU[10]	给出了客户端 CPU 的类型或制造商
UA-Disp	提供了与客户端显示器（屏幕）能力有关的信息
UA-OS	给出了运行在客户端机器上的操作系统名称及版本
UA-Pixels	提供了客户端显示器的像素信息
User-Agent	将发起请求的应用程序名称告知服务器

1.Accept首部

将客户端喜好和能力告知服务器

首　部	描　述
Accept	告诉服务器能够发送哪些媒体类型
Accept-Charset	告诉服务器能够发送哪些字符集
Accept-Encoding	告诉服务器能够发送哪些编码方式
Accept-Language	告诉服务器能够发送哪些语言
TE[11]	告诉服务器可以使用哪些扩展传输编码

2.条件请求首部

为请求加上某些限制

首　部	描　述
Expect	允许客户端列出某请求所要求的服务器行为
If-Match	如果实体标记与文档当前的实体标记相匹配，就获取这份文档[12]
If-Modified-Since	除非在某个指定的日期之后资源被修改过，否则就限制这个请求
If-None-Match	如果提供的实体标记与当前文档的实体标记不相符，就获取文档
If-Range	允许对文档的某个范围进行条件请求
If-Unmodified-Since	除非在某个指定日期之后资源没有被修改过，否则就限制这个请求
Range	如果服务器支持范围请求，就请求资源的指定范围[13]

3.安全请求首部

对请求进行质询/响应认证

首　部	描　述
Authorization	包含了客户端提供给服务器，以便对其自身进行认证的数据
Cookie	客户端用它向服务器传送一个令牌——它并不是真正的安全首部，但确实隐含了安全功能[14]
Cookie2	用来说明请求端支持的 cookie 版本，参见 11.6.7 节

4.代理请求首部

首　部	描　述
Max-Forward	在通往源端服务器的路径上，将请求转发给其他代理或网关的最大次数——与 TRACE 方法一同使用[15]
Proxy-Authorization	与 Authorization 首部相同，但这个首部是在与代理进行认证时使用的
Proxy-Connection	与 Connection 首部相同，但这个首部是在与代理建立连接时使用的

响应首部

首　部	描　述
Age	（从最初创建开始）响应持续时间[16]
Public[17]	服务器为其资源支持的请求方法列表
Retry-After	如果资源不可用的话，在此日期或时间重试
Server	服务器应用程序软件的名称和版本
Title[18]	对 HTML 文档来说，就是 HTML 文档的源端给出的标题
Warning	比原因短语中更详细一些的警告报文

1.协商首部

首　部	描　述
Accept-Ranges	对此资源来说，服务器可接受的范围类型
Vary	服务器查看的其他首部的列表，可能会使响应发生变化，也就是说，这是一个首部列表，服务器会根据这些首部的内容挑选出最适合的资源版本发送给客户端

2.安全响应首部

首　部	描　述
Proxy-Authenticate	来自代理的对客户端的质询列表
Set-Cookie	不是真正的安全首部，但隐含有安全功能，可以在客户端设置一个令牌，以便服务器对客户端进行标识[19]
Set-Cookie2	与 Set-Cookie 类似，RFC 2965 Cookie 定义，参见 11.6.7 节
WWW-Authenticate	来自服务器的对客户端的质询列表

实体首部

首　部	描　述
Allow	列出了可以对此实体执行的请求方法
Location	告知客户端实体实际上位于何处，用于将接收端定向到资源的（可能是新的）位置（URL）上去

1.内容首部

首　部	描　述
Content-Base[20]	解析主体中的相对 URL 时使用的基础 URL
Content-Encoding	对主体执行的任意编码方式
Content-Language	理解主体时最适宜使用的自然语言
Content-Length	主体的长度或尺寸
Content-Location	资源实际所处的位置
Content-MD5	主体的 MD5 校验和
Content-Range	在整个资源中此实体表示的字节范围
Content-Type	这个主体的对象类型

2.实体缓存首部

首　部	描　述
ETag	与此实体相关的实体标记[21]
Expires	实体不再有效，要从原始的源端再次获取此实体的日期和时间
Last-Modified	这个实体最后一次被修改的日期和时间