Basics of HTTP

ABHIJEET THORAT (GIST)
Abhijeet.kadance.96@gmail.com

25 November 2021

Client/Server Messaging Basics

HTTP is a stateless request/response protocol that operates by exchanging messages across a reliable transport- or session-layer "connection". An HTTP "client" is a program that establishes a connection to a server for the purpose of sending one or more HTTP requests. An HTTP "server" is a program that accepts connections in order to service HTTP requests by sending HTTP responses.

- USER AGENT : refers to any of the various client programs that initiate a request, including (but not limited to) browsers, spiders (web-based robots), command-line tools, custom applications, and mobile apps.
- ORIGIN SERVER : The term "origin server" refers to the program that can originate authoritative responses for a given target resource.

URI

HTTP relies upon the Uniform Resource Identifier (URI) standard to indicate the target resource and relationships between resources. Messages are passed in a format similar to that used by Internet mail and the Multipurpose Internet Mail Extensions (MIME). Most HTTP communication consists of a retrieval request (GET) for a representation of some resource identified by a URI.

A client sends an HTTP request to a server in the form of a request message, beginning with a request-line that includes a method, URI,and protocol version, followed by header fields containing request modifiers, client information, and representation metadata, an empty line to indicate the end of the header section, and finally a message body containing the payload body.

A server responds to a client's request by sending one or more HTTP response messages, each beginning with a status line that includes the

- protocol version,
- a success or error code,
- and textual reason phrase

, possibly followed by header fields containing **server information**, **resource metadata**, and **representation metadata**, an **empty line to indicate the end of the header section**, and **finally a message body containing the payload body** (if any) .

The following example illustrates a typical message exchange for a
GET request on the URI

```
Client request:

  GET /hello.txt HTTP/1.1
  User-Agent: curl/7.16.3 libcurl/7.16.3 OpenSSL/0.9.7l zlib/1.2.3
  Host: www.example.com
  Accept-Language: en, mi


Server response:

  HTTP/1.1 200 OK
  Date: Mon, 27 Jul 2009 12:28:53 GMT
  Server: Apache
  Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
  ETag: "34aa387-d-1568eb00"
  Accept-Ranges: bytes
  Content-Length: 51
  Vary: Accept-Encoding
  Content-Type: text/plain

  Hello World! My payload includes a trailing CRLF.
```

Mentioned above can be seen through Browser developer options and some 3rd party tools too..

Cache

A "cache" is a local store of previous response messages and the subsystem that controls its message storage, retrieval, and deletion.A cache stores cacheable responses in order to reduce the response time and network bandwidth consumption on future, equivalent requests.

This is the reason that you can see websites even if there is no Internet connection. The same Url mapping request would get the server cache entry of the same request previously done.

HTTP headers | Host

The **HTTP Host** represents the **domain name** of the server. It may also represent the **Transmission Control Protocol (TCP)** port number which the server uses. Defining the port number is optional, the default value is considered. For example, “80” is assigned as the port number for an HTTP URL when there is no port number specified. The HTTP Host header is a request type header. The host header field must be sent in all HTTP/1.1 request messages. If a request message does not have any header field or more than one header field, a 400 Bad Request is sent.

Syntax :

```
Host: <host>:<port>
```

Directives: The HTTP header Host accepts two directives mentioned above and described below:

<host>: This directive represents the domain name of the server.
<port>: This directive is an **optional** one. It represents the TCP port number in which the server is working.

```
Example:
```

```
GET / HTTP/1.1
User-Agent: WebSniffer/1.0 (+http://websniffer.cc/)
Host: google.com
Accept: */*
Referer: https://websniffer.cc/
Connection: Close
```

HTTP Response Header

Name	Value
HTTP/1.1 301 Moved Permanently	
Location:	**http://www.google.com/**
Content-Type:	text/html; charset=UTF-8
Date:	Mon, 29 Nov 2021 08:34:53 GMT
Expires:	Wed, 29 Dec 2021 08:34:53 GMT
Cache-Control:	public, max-age=2592000
Server:	gws
Content-Length:	219
X-XSS-Protection:	0
X-Frame-Options:	SAMEORIGIN
Connection:	close

Content

```
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

Here the host header has all the information regarding the requested URI, like what type of request is done, Host name, content type and much more.