HTTP

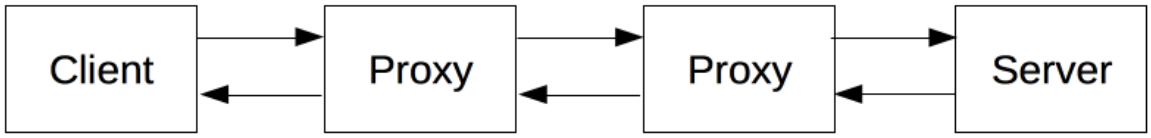
HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files. such as text, images, sound, video and other multimedia files -- over the web. As soon as a user opens their web browser, they are indirectly using HTTP. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols, which forms the foundation of the internet. The latest version of HTTP is HTTP/3.

- HTTP stands for Hypertext Transfer Protocol.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- HTTP is an application-level protocol. The communication usually takes place through TCP/IP sockets, but any reliable transport can also be used.
- The standard (default) port for HTTP connection is 80, but other port can also be used.
- The first version of HTTP was HTTP/0.9, which was introduced in 1991.
- The latest version of HTTP is HTTP/3, which was published in September 2019. It is an alternative to its processor HTTP/2.
- HTTP is used to make communication between a variety of hosts and clients. It supports a mixture of network configuration.
- HTTP is a protocol that is used to transfer the hypertext from the client end to the server end, but HTTP does not have any security.
- Whenever a user opens their Web Browser, that means the user indirectly uses HTTP.

Components of Http based system

HTTP is a client-server protocol: requests are sent by one entity, the user-agent. Most of the time the user-agent is a Web browser.Each individual request is sent to a server, which handles it and provides an answer called the *response*. Between the

client and the server there are numerous entities, collectively called proxies, which perform different operations and act as gateways or caches.



Client user agent

The *user-agent* is any tool that acts on behalf of the user. This role is primarily performed by the Web browser,The browser is always the entity initiating the request. It is never the server.

To display a Web page, the browser sends an original request to fetch the HTML document that represents the page. It then parses this file, making additional requests corresponding to execution scripts, layout information (CSS) to display, and sub-resources contained within the page (usually images and videos). The Web browser then combines these resources to present the complete document, the Web page. Scripts executed by the browser can fetch more resources in later phases and the browser updates the Web page accordingly.

A Web page is a hypertext document. The browser translates these directions into HTTP requests, and further interprets the HTTP responses to present the user with a clear response.

The web server

On the opposite side of the communication channel is the server, which *serves* the document as requested by the client. A server appears as only a single machine virtually; but it may actually be a collection of servers sharing the load (load balancing), or a complex piece of software interrogating other computers (like cache, a DB server, or e-commerce servers), totally or partially generating the document on demand. A server is not necessarily a single machine, but several server software instances can be hosted on the same machine.

Basic Features

There are three basic features that make HTTP a simple but powerful protocol:

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

The importance of Http

HTTP is a simple protocol, but more than that, it's a striking feature that is also accessible, since it was designed to have messages that can be read and understood by any user. In addition, its stateless nature simplifies the performance of the server and makes it faster, since there is no need to store or clear data for the next requests.

Disadvantage

- HTTP requires high power to establish communication and transfer data.
- HTTP is less secure, because it does not uses any encryption method like https use TLS to encrypt normal http requests and response.
- HTTP is not optimized for cellular phone and it is too gabby.
- HTTP does not offer genuine exchange of data because it is less secure.
- Client does not close connection until it receives complete data from server and hence server needs to wait for data completion and cannot be available for other clients during this time.

Messages

HTTP messages are of two types: request and response. Both the message types follow the same message format.

Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.

Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.

HTTP methods

HTTP methods help to distinguish requests made to the server. This is so we know what actions have been performed on the resource. Here are the most common methods Create, Read, Update, Delete actions. Actually, there are more methods in the HTTP protocol,

HTTP method	Description
GET	the GET method is a basic request. Responsible for displaying the current representation of the resource. For example, the HTTP GET method is used to display web pages or forms. Parameters are passed in a URL address. So everybody can see them. That's why GET should only be used to display, not to save sensitive data.
POST	the POST method is used when you want to add new resources. For example, creating a new account or sending files. Parameters are passed in the body. The POST method is safer than GET and should be used to send sensitive data.
PUT	the PUT method is used when you want to update existing resource. For example you want to change an email in your existing account.
DELETE	the DELETE method is responsible for deleting resources. For example, you can delete your account.

HTTP Status Code

In response to HTTP requests, servers often issue response codes, indicating the request is being processed, there was an error in the request or that the request is being redirected. Common response codes include:

- **200 OK.** This means that the request, such as GET or POST, worked and is being acted upon.
- **300 Moved Permanently.** This response code means that the URL of the requested resource has been changed permanently.
- **401 Unauthorized.** The client, or user making the request of the server, has not been authenticated.
- **403 Forbidden**. The client's identity is known but has not been given access authorization.
- **404 Not Found.** This is the most frequent error code. It means that the URL is not recognized or the resource at the location does not exist.
- **500 Internal Server Error.** The server has encountered a situation it doesn't know how to handle.