

HTTP or HyperText Transport Protocol

Foreword¹

In order for information systems to communicate with each other, languages have been created, which are called protocols. We differentiate between low-level protocols such as TCP or UDP, which govern a large range of communications on the Internet, and high-level protocols such as HTTP(s), SMTP, FTP, which rely on TCP or UDP to exist, it is an overlay.

Creation²

It was invented in 1990 by Tim Berners-Lee, who also created the tag "language" called HTML and web addresses like "http://www.example.com".

Definition, need and purpose³

It is literally the hypertext transfer protocol, it is a communication protocol between a client and a server, that is to say a client, so the browser in the vast majority of cases, and a server, a computer which is constantly on, and which serves and therefore responds to the client's requests or resource requests. This is called a client-server model. An HTTP request will therefore imply a response, in concrete terms a visitor wants to access a WEB page, he will therefore type the URL in his browser, and this will build and send this request to the server, which will send a response containing the page to be visited. The HTTP protocol is part of the application layer of the OSI model (Open Systems Interconnection).

Methods and structure of an HTTP request/response⁴

Both a request and a response have a similar structure:

1. A start line or command line (Command, URL, Version, Method, Status)
2. Headers (optional)
3. An empty line
4. A content or message body (optional)

¹ <https://openclassrooms.com/fr/courses/1946386-comprendre-le-web/6874807-decouvrez-les-protocoles>

² https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Impl%C3%A9mentation

³ <https://www.pierre-giraud.com/http-reseau-securite-cours/introduction-protocole-transfert-hypertexte/>

⁴ https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol#HTTP_1.0

There are several types of HTTP requests, which are differentiated from each other by method calls. They allow to carry out a specific task on the server side:

- GET :
 - permet d'obtenir des informations (document, graphique, le résultat d'une requête dans une base de données, ...);
 - peut envoyer des informations dans l'URL limités en taille à 240 caractères sur certains serveurs (données d'un formulaire simple);
- POST permet de poster des informations (une données graphique, des données pour une base de données, ...) en quantité illimité ;
- HEAD pour demander les en-têtes de la réponse seulement pour récupérer la taille des documents, l'heure de modification, ...
- PUT et DELETE sont faites pour agir sur les documents du serveur (elles ne sont pas largement supportées à cause des problèmes de sécurité ;
- TRACE retourne le contenu exact de la demande (utilisée pour le débogage) ;
- OPTIONS pour demander au serveur les méthodes et les options qu'il supporte.

5

The characteristics of each method are as follows:

HTTP method ↕	RFC ↕	Request has Body ↕	Response has Body ↕	Safe ↕	Idempotent ↕	Cacheable ↕
GET	RFC 7231	Optional	Yes	Yes	Yes	Yes
HEAD	RFC 7231	Optional	No	Yes	Yes	Yes
POST	RFC 7231	Yes	Yes	No	No	Yes
PUT	RFC 7231	Yes	Yes	No	Yes	No
DELETE	RFC 7231	Optional	Yes	No	Yes	No
CONNECT	RFC 7231	Optional	Yes	No	No	No
OPTIONS	RFC 7231	Optional	Yes	Yes	Yes	No
TRACE	RFC 7231	No	Yes	Yes	Yes	No
PATCH	RFC 5789	Yes	Yes	No	No	No

6

⁵ http://freetique.free.fr/Cours/Cours%20Efrei/Cours%20de%20Java/Servlets/servlet_http.pdf

⁶ <https://developer.mozilla.org>

Evolution⁷

HTTP has had the capacity since its creation to adapt to the continuous needs of the evolution of the WEB. At the beginning, the pages were only made of HTML, a "language" tag, and with the progression and diversification of the WEB, other resources besides HTML have been implemented to embellish the websites, for example CSS for styling, Javascript for interactivity between the user and the page, and media, i.e., images and videos. This means that several resources are requested for a single HTTP request. Several resources, therefore, requests that follow one another. In concrete terms, the first request will retrieve the HTML document, and the browser will read this document and send a new request for each new resource.

The major differences between the versions of HTTP are the additions of new methods seen above. HTTP-2 introduced a system of priority between the different resources to be retrieved, which reduces the time taken to download a page, and introduced a multiplexing system which allows asynchronous downloading, i.e., parallel requests for the same connection to the server.

Through practice

There are several powerful tools for simulating or making real HTTP requests. Here are some of them:

1. cURL, stands for "Client URL Request Library" used on the command line to create requests and receive the response in raw form, several arguments/options accompany this tool in CLI.
2. Classes specific to different languages such as Javascript allow a request to be made without having to refresh the web page, this class is called "Fetch".
3. Postman, one of the most advanced tools for making requests of various types, performing tests and learning to communicate with URLs and APIs.
4. Wireshark allows you to retrieve, analyse, visualise, filter and format the various packets that pass through a network.

⁷ <https://www.pierre-giraud.com/http-reseau-securite-cours/introduction-protocole-transfert-hypertexte/#:~:text=HTTP%2C%20a%20protocol%20for%20transfer%20client%20%E2%80%93%20server&text=For%20example%2C%20when%20a%20visitor,is%20called%C3%A9%20a%20request%C3%AAt%20HTTP.>

Here is a screenshot of a query and its raw response from the IntelliJ integrated development environment of the JetBrains suite:

