Unit 2. Task 1: HTTP and HTTPS Protocols

Javier Aguilera Aguilera 2º CFGS DAW

Contents

1 HTTP

1.1 What is HTTP

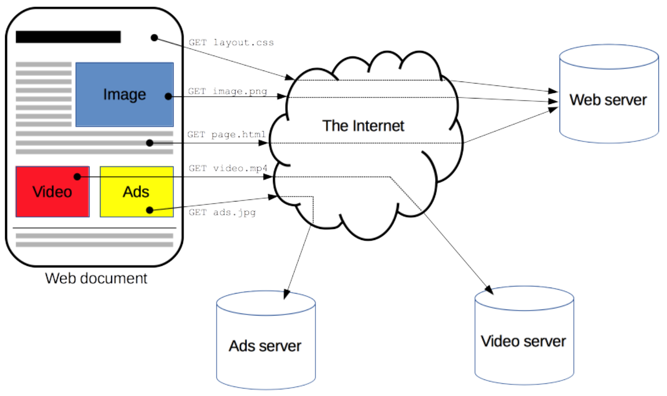


Figure 1: Fetching a page

HTTP, which stands “Hypertext Transfer Protocol”, is the name of a protocol that allows you to make a request for data and resources, for example HTML documents.

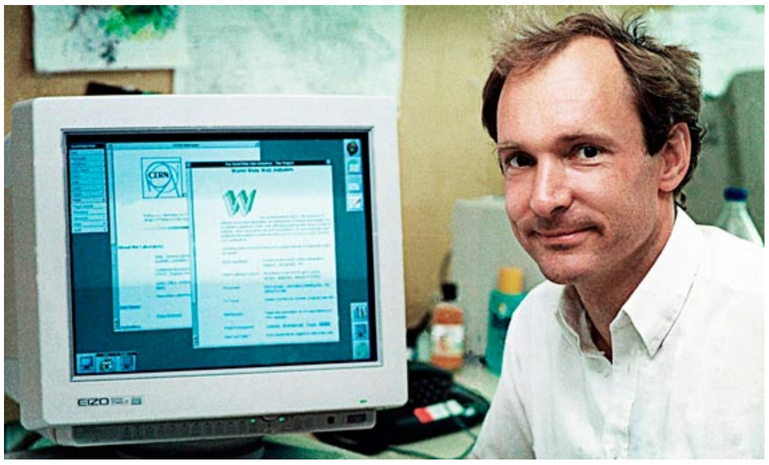


Figure 2: Tim Berners-Lee

It was created on the early 90’s by Tim Berners-Lee, although it has had several updates over time. It is the basis of any type of data exchange on the Web, and a client-server protocol structure. This means that a data request is inicialized by the element that is going to recive the data, usually a web browser.

Thus, a complete web page results from the join of several subdocuments received.

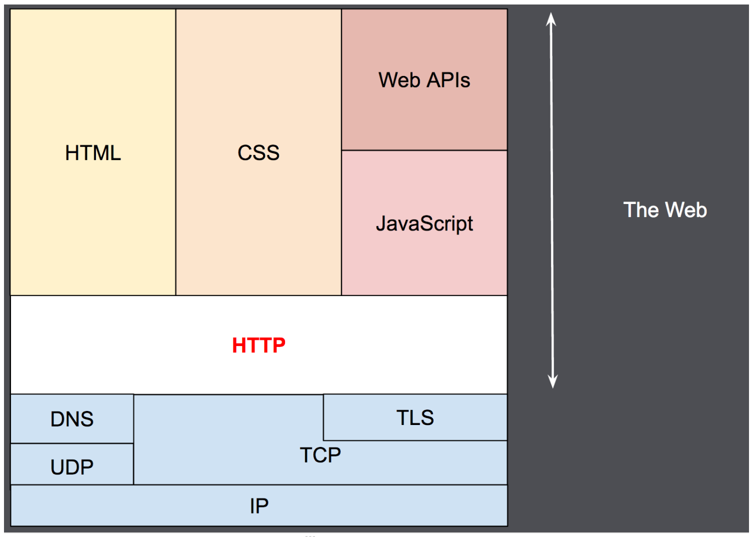


Figure 3: Web layers

1.1.1 HTTP key characteristics

- HTTP is symple: HTTP is designed and developed to be read and easy understood by people, which make debugging easier and reduces learning curve.
- HTTP is extensible: HTTP headers (introduced in HTTP/1.0) makes this protocol easier to extends and experiment.
- HTTP is a sessions protocol, but without states: This means that HTTP doesn't save any data between two request from the same sesion. This entails into some troubles, if a user wants to interact orderly and coherently with some determined web, for example using "shopping cart" . We can solve it using cookies: HTTP cookies allows to save up data from the comunication sesion.
- HTTP and conections: HTTP doesn't need the protocol that supports it to maintain a continuous connection between the participants in the communication, it only needs to be a reliable protocol or that it doesn't lose messages.

1.2 HTTP versions

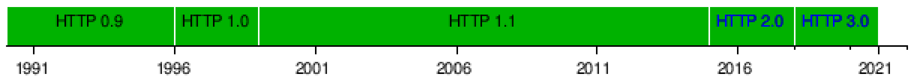


Figure 4: HTTP line time

HTTP has gone through multiples versions, most of them backward compatible. The table 1 has a quick review of the released date for each one.

Version	*Release date*
0.9	1991
HTTP/1.0	May 1996
HTTP/1.1	June 1999
HTTP/1.2	February 2000
HTTP/2	May 2015
HTTP/3	October 2018

Table 1: HTTP versions release date

1.3 HTTP basic architecture

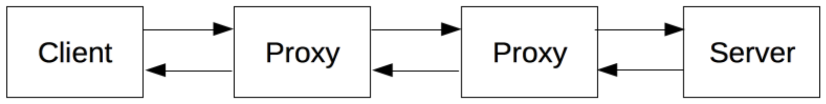


Figure 5: Client - server chain

HTTP is a protocol based on the principle client-server: Request are send by a entity: the user agent. Each individual request is sent to a server that manages and responds to it. Between every request and response there are several intermediaries, proxies, that makes differentes functions: gateways or caches, for example.

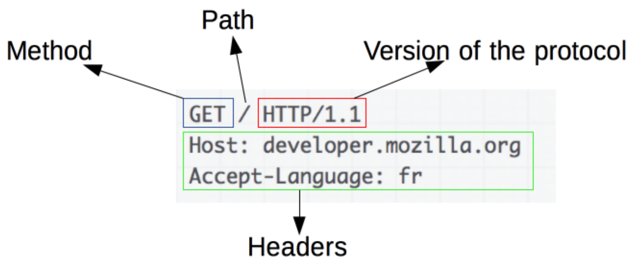
1.4 How it works

1.4.1 Messages

In HTTP/1.1 and earlier, the messages were in text format and were fully understandable directly by a person. In HTTP/2, the messages are structured in a new binary format and the frames allow the compression of the headers and their multiplexing. Thus, even if only part of the original message in HTTP is sent in this format, the sematics of each message is the same and the client can form the original message in HTTP/1.1. Then, it is possible to interpret HTTP/2 messages in the HTTP/1.1 format.

There are two types of HTTP messages: requests and responses, each following its own format.

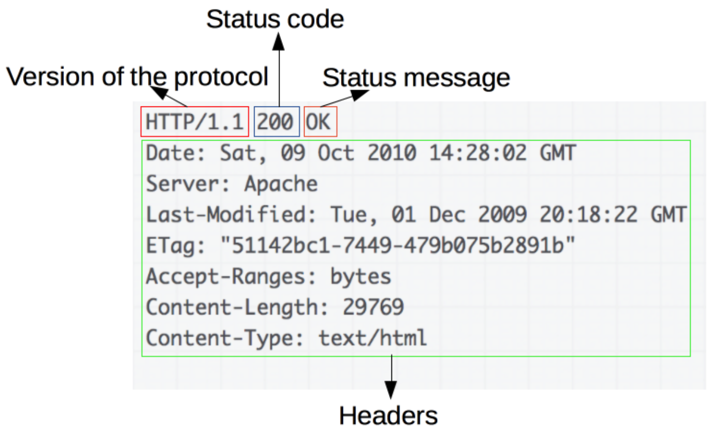
- Request



```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

Figure 6: HTTP request

- Response



```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
```

Figure 7: HTTP response

1.4.2 HTTP flow

When a client wants to communicate with the server, directly or using an intermediate proxy, it has to follow the next steps:

1. Open a TCP conection: It will be used to make a single or multiple request, and the response will be received. The client can open a new connection, reuse an existing one or open several at the same time.

2. Make a HTTP request (Figure 6)

3. Read the response sent by the server (Figure 7)

2 HTTPS

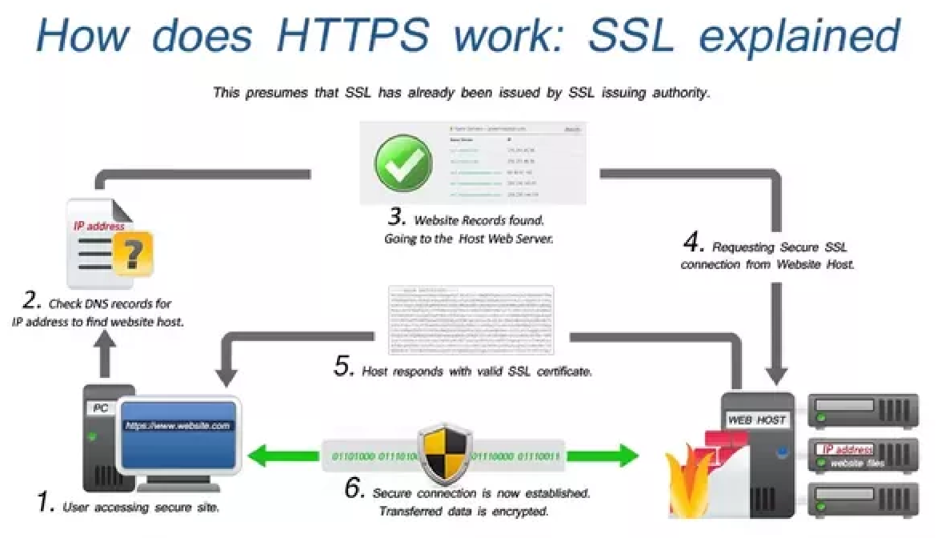

How does HTTPS work: SSL explained
This presumes that SSL has already been issued by SSL issuing authority.
3. Website Records found. Going to the Host Web Server.
IP address
2. Check DNS records for IP address to find website host.
4. Requesting Secure SSL connection from Website Host.
5. Host responds with valid SSL certificate.
PC
https://www.website.com
1. User accessing secure site.
6. Secure connection is now established. Transferred data is encrypted.
WEB HOST
IP address
website files

Figure 8: How HTTPS works?

HTTPS is an HTTP-based application protocol used for the secure transfer of data of hypertext.

Use TLS to create an encrypted channel more suitable for sensitive information traffic than the HTTP protocol. In this way, it is achieved that the sensitive information can't be used by an attacker who has managed to intercept the data transfer of the connection because it is an encrypted data stream that will be impossible to decrypt.

2.1 Certificates

To prepare a web server that accepts HTTPS connections, the administrator must create a public key certificate for the web server. This certificate must be signed by a certificate authority for the web browser to accept it. The authority certifies that the certificate holder is who he claims to be. Web browsers are generally distributed with root certificates signed by most certificate authorities so they can verify certificates signed by them.

2.2 Differences between HTTP and HTTPS

The following table has a sumary of the main differences between HTTP and HTTPS:

Characteristic	*HTTP*	*HTTPS*
URL begining	"http://"	"https://"
Port	80	443/4433
Uses TLS/SSL	No	Yes
Certificate	No	Yes

Table 2: Differences

HTTP is subject to man-in-the-middle and eavesdropping attacks that can allow the attacker to gain access to banks and website accounts and confidential information .

References

[1] https://developer.mozilla.org/es/docs/Web/HTTP/Overview

[2] https://www.bbvaopenmind.com/wp-content/uploads/2015/06/BBVA-OpenMind-tim-berners-lee-1-1.jpg

[3] https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto

[4] https://i.stack.imgur.com/oH2tP.png