Protocollo HTTP

**HTTP** (Hypertext Transfer Protocol, letteralmente protocollo di trasferimento di un ipertesto) è un **protocollo informatico** che consente la comunicazione tra client e server attraverso internet.

Un protocollo **informatico**  è un insieme di regole definite al fine di gestire la comunicazione tra dispositivi.

**Come funziona HTTP: request e response**

Il protocollo HTTP prevede che il Client faccia una richiesta verso il server e il server risponda.

**Request: il client invia una richiesta**

Ogni conversazione tra client e server sul web inizia con una richiesta, un in un formato speciale, noto appunto come HTTP.

Una richiesta HTTP è composta dalle seguenti parti:

[method] [URI] [version]

[headers]

[body]

Un esempio di richiesta HTTP sarebbe quindi qualcosa del genere:

GET /blog/protocollo-http-cosa-e-come-funziona HTTP/1.1

Host: www.dominioesempio.it

Accept: text/html

User-agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64)

La prima riga di una richiesta HTTP contiene due parti fondamentali:

il metodo http: GET

l'URI: /blog/protocollo-http-cosa-e-come-funziona

L'URI è l'indirizzo o il percorso univoco che identifica la risorsa che il client desidera. Il metodo HTTP definisce ciò che il client vuole fare con la risorsa.

**Request methods: metodi di una richiesta HTTP**

I metodi HTTP definiscono i modi in cui il client può interagire con la risorsa. La tabella seguente mostra alcuni dei metodi utilizzati durante una richiesta HTTP.

Metodi HTTP

GET: Recupera una risorsa dal server (ad es. visitando una pagina)

POST: Invia una risorsa al server (ad. es compilando un modulo)

DELETE: Cancella una risorsa dal server (ad es. eliminando un file)

PUT: Memorizza una risorsa sul server (as es. caricando un file)

HEAD: Recupera solo l'header della risposta senza la risorsa

Esistono altri metodi definiti dalle specifiche del protocollo HTTP, ma molti di essi sono poco utilizzati o non supportati.

**Request headers**: intestazioni di richiesta

Oltre alla prima riga, una richiesta HTTP contiene altre informazioni chiamate headers: intestazioni di richiesta.

Le intestazioni forniscono molte informazioni supplementari come l'host dove si trova la risorsa (Host), i formati di risposta accettati dal client (Accept) e l'applicazione utilizzata dal client per effettuare la richiesta (User-Agent).

L’intestazione può contenere:

Host: Specifica il server dove è ospitata la risorsa richiesta

Accept: Specifica quali formati di file (i MIME Type) sono accettati dal client

Accept-Language: Indica quale lingua supporta il client

Accept-Charset: Indica il set di caratteri accettato dal client

User-Agent: Specifica quale applicazione sta effettuando la richiesta

Per approfondire leggi HTTP request headers

**Response**: il server restituisce una risposta

Il client invia la richiesta al server, quindi attende la risposta (response). L'obiettivo del server è quindi di interpretare la richiesta del client e restituire una risposta.

Una volta che il server ha ricevuto la richiesta, conosce esattamente la risorsa di cui il client ha bisogno (tramite l'URI) e ciò che il client vuole fare con quella risorsa (tramite il metodo).

Ad esempio, nel caso di una richiesta GET, il server prepara la risorsa e la restituisce in una risposta HTTP.

Come per la richiesta, una risposta HTTP è composta dalle seguenti parti:

[version] [status code] [reason]

[headers]

[body]

La risposta inviata al browser sarà simile a questa

HTTP/1.1 200 OK

Date: Wed, 08 Aug 2018 21:23:53 GMT

Connection: keep-alive, Keep-Alive

Content-Encoding: br

Server: Apache

Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>

<html lang="it">

 <head>

 <title>Protocollo HTTP: cos'è e come funziona passo passo</title>

 </head>

 <body>

 <p>Protocollo HTTP</p>

 </body>

</html>

La risposta HTTP contiene la risorsa richiesta (il contenuto HTML in questo caso), più altre importanti informazioni come il codice di stato della risposta.

**Status Codes**: codici di stato HTTP

Il codice di stato comunica il risultato della richiesta al client.

Esistono diversi codici di stato che indicano il successo, un errore o che il client deve fare qualcosa (ad esempio reindirizzare l'utente a un'altra pagina).

Il codice di stato è composto da 3 cifre, la prima indica la “famiglia” dello stato, ad esempio se la prima cifra è 2 significa successo, se è 5 errore interno al server ecc. Le ultime due cifre, invece, specificano meglio la tipologia di situazione (successo, errore ecc) all’interno della famiglia individuata dalla prima cifra.

Di seguito alcuni dei codici di stato più frequenti:

|  |  |
| --- | --- |
| **Codice** | **Descrizione** |
| 200 | OK - La richiesta è andata a buon fine |
| 301 | Moved Permanently -La risorsa richiesta è stata spostata definitivamente ad un nuovo URI |
| 404 | Not Found -La risorsa richiesta non è stata trovata |
| 500 | Internal Server Error - Errore generico causato di solito da una configurazione errata del server |

Response headers: intestazioni di risposta

Come la richiesta, anche una risposta HTTP contiene informazioni aggiuntive note come intestazioni di risposta HTTP (response headers). Il corpo della stessa risorsa può essere restituito in più formati diversi come HTML o JSON e l'intestazione Content-Type utilizza tipi di MIME, come ad esempio text/html, per dire al client quale formato viene restituito.

L’intestazione può contenere:

Content-Encoding: Tipo di codifica utilizzata sui dati dal server

Content-Type: Indica il MIME Type dei dati (Es. text/html)

Date: La data e l'orario in cui i dati sono stati inviati

Serve: Nome del server web utilizzato