

HTTP

Protocollo (**STATELESS**) livello **applicativo** basato su **TCP**, utilizzato per trasferire le risorse Web, gestendo sia le **richieste** (URL) inviate al server, sia le **risposte** (pagine) inviate al client.

Il client è un programma applicativo che stabilisce una connessione al fine di inviare richieste

il server è un programma applicativo che accetta connessioni al fine di ricevere richieste ed inviare le rispettive risposte

il messaggio, è l'unità base di HTTP ed è definita come una sequenza di byte divisa in :

- REQUEST
- RESPONSE

L'URI (da non confondere con URL) è l'identificatore unico per una determinata risorsa

Un http message è definito tra due strutture :

- MESSAGE HEADER, contenente informazioni necessarie per l'identificazione del messaggio
- MESSAGE BODY, contiene materialmente i dati trasportati dal messaggio.

COMANDI DELLA RICHIESTA

- GET, serve per richiedere una risorsa specifica a un server, come ad esempio cliccando un link di un documento *HTML* o specificando un URL nell'apposito campo del browser, previsto con il passaggio di 2 parametri; l'informazione è passata appunto sopra URL e quindi è adatto per richieste non sensibili e non influenti sullo stato del server.
- POST, progettato come il messaggio per richiedere una risorsa, non contiene i dettagli della risorsa nell'URL ma bensì nel BODY del messaggio, per questo motivo adatto a comunicazioni di dati sensibili o che **ALTERANO** lo stato del server. (**FORM DI AUTENTICAZIONE**)

- PUT memorizza sul server una risorsa, come il POST serve per trasmettere informazioni dal client al server, a sua differenza però crea la risorsa, o la sovrascrive.
- DELETE eliminare una risorsa avendone l'URL
- HEAD funziona come il GET ma solo con header, utilizzato per verificare funzionamento di un URL.
- OPTION richiede informazioni sulle opzioni disponibili per la comunicazione
- TRACE consente al client di vedere cosa è stato ricevuto dal server, utile per diagnostica.

Proxy: Programma applicativo in grado di agire sia come Client che come Server al fine di effettuare richieste per conto di altri Clienti. Le Request vengono processate internamente oppure vengono ridirezionate al Server. Un proxy deve interpretare e, se necessario, riscrivere le Request prima di inoltrarle.

Cookie : struttura dati mobile come un token, dal client al server e viceversa, generati da entrambi vengono sempre passati ad ogni trasmissione ed hanno lo scopo di fornire un supporto per il mantenimento di stato in un protocollo come HTTP.

Web Cache : creata per memorizzare copie temporanee di documenti web, come pagine HTML, una web cache memorizza i documenti che la attraversano al fine di utilizzare i documenti in cache per le successive richieste, in modo da diminuire l'uso della banda ed il carico del server. Un esempio è lo **User Agent Cache**, che mantiene una cache delle pagine visitate dall'utente, evitando latenze dovute al caricamento di elementi statici come icone e sfondi.