

# HTTP

# HTTP e il modello client-server

Il web è basato sul modello client-server che ha due componenti:

- un client che fa richieste
- un server che risponde

I client sono elementi attivi che utilizzano HTTP per fare richieste ad un server. Usano un URL per identificare la risorsa che vogliono ottenere. Richiedono pagine web, immagini, video, ecc.

I server sono elementi passivi che rispondono alle richieste dei client. Rimangono in ascolto di eventuali connessioni su una determinata porta TCP. Utilizzano HTTP per rispondere alle richieste dei client. Forniscono pagine web, immagini, video, ecc.

# HTTP e il modello client-server

HTTP è un protocollo usato per trasmettere risorse, non solo file. Generalmente le risorse sono file, ma possono essere anche risultati di un calcolo, un'immagine generata al volo, ecc.

Il W3C definisce un URI (Uniform Resource Identifier) come una sequenza di caratteri che identifica una risorsa astratta o fisica, mentre un URL (Uniform Resource Locator) è un termine informale associato a protocolli quali HTTP, FTP, ecc.

Un esempio di URL è

```
https://ospite:123@stecca.dev:443/uploads/query.php?name=stecca&surname=dev#anchor
```

dove:

- `https` è il protocollo
- `ospite:123` è l'utente e la password
- `stecca.dev` è il nome del server (host)
- `443` è la porta
- `/uploads/query.php` è il percorso della risorsa
- `name=stecca&surname=dev` è la query string
- `anchor` è un fragment

# Il protocollo HTTP

HTTP (HyperText Transfer Protocol) è un protocollo di livello applicativo che fornisce il livello di trasporto a tutti i protocolli di livello superiore. Possiamo dire che il web è nato grazie all'insieme di HTML, HTTP e URL.

Il protocollo HTTP utilizza le sessioni per mantenere lo stato tra le richieste. Ogni sessione inizia con una connessione TCP, utilizzando la porta 80 per HTTP e la porta 443 per HTTPS. Una volta stabilita la connessione, il client e il server possono scambiarsi messaggi.

# Connessione permanente

Nella connessione permanente, il client e il server mantengono la connessione TCP aperta dopo aver ricevuto la risposta. Questo permette di inviare più richieste e risposte senza dover aprire una nuova connessione TCP per ogni richiesta.

Esistono due tipi di connessione permanente:

- Connessione permanente **incanalata**: il client e il server possono inviare richieste e risposte in qualsiasi momento. Il server invia le risposte in ordine.
- Connessione permanente **non incanalata**: il client invia una richiesta e attende la risposta prima di inviare la richiesta successiva.

# HTTP Request

Una richiesta HTTP è composta da:

- una riga di richiesta (request line)
- un'intestazione (header)
- un corpo (body)

La sintassi della riga di richiesta è:

```
<metodo> <URI> <versione HTTP>
```

```
[Header]
```

```
[Body]
```

# HTTP Request

Dove:

- `<metodo>` è il metodo HTTP utilizzato
- `<URI>` è l'URI della risorsa richiesta
- `<versione HTTP>` è la versione del protocollo HTTP utilizzata
- `[Header]` è l'intestazione della richiesta, e contiene informazioni aggiuntive
- `[Body]` è il corpo della richiesta, e contiene i dati inviati al server



# HTTP Response

Una risposta HTTP è composta da:

- una riga di stato (status line)
- un'intestazione (header)
- un corpo (body)

La sintassi della riga di stato è:

```
<versione HTTP> <codice di stato>
```

```
[Header]
```

```
[Body]
```

# Metodi HTTP

I metodi HTTP definiscono le azioni che possono essere eseguite su una risorsa. I metodi HTTP più comuni sono:

- **GET** : richiede una rappresentazione della risorsa specificata. Le richieste GET non devono avere effetti collaterali sul server.
- **HEAD** : richiede una risposta identica a quella di una richiesta GET, ma senza il corpo della risposta.
- **POST** : invia dati al server per creare una nuova risorsa.
- **PUT** : invia dati al server per creare o aggiornare una risorsa.
- **DELETE** : elimina la risorsa specificata.

# Codifica dei caratteri

La codifica URL è un meccanismo per trasformare i caratteri non ASCII in una forma che possa essere trasmessa su Internet. La codifica URL sostituisce ogni carattere non ASCII con una sequenza di tre caratteri: un segno di percentuale seguito da due cifre esadecimali che rappresentano il valore del carattere.

Ad esempio, il carattere spazio viene codificato come `%20`.

# Codici di stato HTTP

I codici di stato HTTP sono un insieme di codici numerici che indicano lo stato della risposta HTTP. I codici di stato HTTP sono divisi in cinque categorie:

- 1xx: informazioni
- 2xx: successo
- 3xx: reindirizzamento
- 4xx: errore del client
- 5xx: errore del server

# Codici di stato HTTP

I codici di stato HTTP più comuni sono:

- `100 Continue` : il server ha ricevuto la richiesta e il client può continuare a inviare la richiesta.
- `200 OK` : la richiesta è stata completata con successo.
- `201 Created` : la richiesta è stata completata con successo e una nuova risorsa è stata creata.
- `204 No Content` : la richiesta è stata completata con successo, ma non c'è contenuto da restituire.
- `301 Moved Permanently` : la risorsa richiesta è stata spostata in un nuovo URI.
- `400 Bad Request` : la richiesta non può essere elaborata a causa di un errore di sintassi.

- `401 Unauthorized` : la richiesta non è stata eseguita perché non è stata fornita l'autorizzazione.
- `403 Forbidden` : il server ha rifiutato di elaborare la richiesta.
- `404 Not Found` : la risorsa richiesta non è stata trovata.
- `500 Internal Server Error` : il server ha riscontrato un errore imprevisto.