

HTTP / 1.0

IN HTTP/1.0, IL CLIENT apre una connessione, manda una richiesta al server e riceve una risposta; dopodiché, la connessione viene chiusa.

QUESTO SIGNIFICA CHE LE CONNESSIONI SONO NON PERMANENTI, NON PERSISTENTI.

IL FORMATO DELLA RICHIESTA È:

1. **PRIMA RIGA:** CONTIENE IL METODO (AD ES. GET), LA RISORSA (RADICE / PER LA HOMEPAGE), E LA VERSIONE DEL PROTOCOLLO. OGNI RIGA TERMINA CON UN ACCAPO.
2. **HEADER:** DOPO LA PRIMA RIGA, SI SPECIFICANO GLI HEADER, L'INTESTAZIONE DEL PROTOCOLLO. UN HEADER CHE È OBBLIGATORIO DALLA DEFINIZIONE DEL PROTOCOLLO È L'HEADER HOST. L'IDEA DEL FUNZIONAMENTO È: HO APERTO UNA CONNESSIONE, PER APRIRE QUESTA CONNESSIONE HO DOVUTO INDICARE L'INDIRIZZO DEL SERVER (INFO GIÀ DATA) MA LO DEVO RIPETERE NEL GET. L'UNICO HEADER OBBLIGATORIO ALL'INTERNO DI UNA RICHIESTA È QUELLO DELL'HOST. ALCUNI HEADER OPZIONALI SONO:
 - USER-AGENT -> SERVE PER DIRE L'IDENTITÀ DEL BROWSER
 - ACCEPT-LANGUAGE -> IT, PREFERISCO LA VERSIONE ITALIANA
 - <CR><LF> -> SERVE PER ANDARE A CAPO (LA RICHIESTA FINISCE SEMPRE CON UNA RIGA VUOTA)

QUINDI AD ESEMPIO:

```
GET / HTTP/1.0
HOST: WWW.DIBRIS.UNIGE.IT
USER-AGENT:
ACCEPT-LANGUAGE: IT <CR><LF>
<CR><LF>
```

RIPOSTA STANDARD POSITIVA:

```
HTTP/1.0 200 OK <CR><LF>
```

SEGUE POI LA PARTE DEGLI HEADER, SEGUITA DA UNA RIGA VUOTA E INFINE DAL BODY.

GLI HEADER NELLA RISPOSTA:

- CONNECTION: CLOSE -> INDICA CHE LA CONNESSIONE SARÀ CHIUSA DOPO LA RISPOSTA (NON PERSISTENTE)

- Date -> Data di invio della risposta in formato standard GMT (per rendere indipendente dalla posizione del server e il suo fuso orario)
- Server -> Versione del server contattato
- Last-Modified -> Data dell'ultima modifica del file inviato, utile per gestire il caching
- Content-Length -> Lunghezza del file in byte
- Content-Type -> Tipo di contenuto, espresso in formato MIME (es. text/html)

Quindi ad esempio:

```
HTTP/1.0 200 OK <CR><LF>
Connection: Close
Date:
Server:
Last-Modified:
Content-Length:
Content-Type:
<CR><LF>
```

Questa struttura potrebbe essere complicata -> una complicazione potrebbe succedere quando l'interazione non avviene direttamente con il server di riferimento ma con un elemento intermedio, ossia un proxy.

Dal punto di vista del client non cambia nulla, il proxy invece usa un meccanismo di caching e mantiene una copia di alcuni file posti originariamente sul server -> velocizzare la risposta al client mandando come risposta la versione del file che si trova all'interno della cache, senza contattare il server

Un'altra cosa abbastanza pratica è quella di mantenere una cache all'interno del client: se il client accede più volte allo stesso file, la prima volta lo scarica dal server o dal proxy e le volte successive ha già la copia in locale

Come faccio a sapere se questi contenuti sono aggiornati? A volte le versioni nel server cambiano, quindi devono essere ricaricate dal server originario. Come si può risolvere? Quando mi viene dato il file mi viene detta la data dell'ultima volta che è stato modificato il file. Io so qual è l'ultima data della modifica del file che ho all'interno della mia cache; quindi, posso chiedere al server la sua e se sono uguali non devo ricaricare il

FILE, SE INVECE SONO DIVERSE SO CHE LA MIA COPIA È OBSOLETA E DEVO SOSTITUIRLA CON QUELLA AGGIORNATA.

ABBIAMO VISTO CHE OLTRE AL METODO GET ABBIAMO UN METODO CHIAMATO HEAD; LA DIFFERENZA È CHE GET CONTIENE TUTTA LA PARTE DI INTESTAZIONE + FILE MENTRE HEAD CONTIENE SOLO LA PARTE INTESTAZIONE E NON IL FILE. POSSIAMO CHIEDERE IL FILE TRAMITE IL METODO HEAD.

IN PRATICA PERÒ SI FA UNA COSA DIVERSA: SI AGGIUNGE UN HEADER AL GET. L'HEADER AGGIUNTO SI CHIAMA **IF MODIFIED SINCE** SEGUITO DA UNA DATA IN FORMATO GMT.

LA RISPOSTA DA PARTE DEL SERVER PUÒ ESSERE DI 2 TIPI DIVERSI:

- 200 OK CON CONTENUTO DEL FILE (OSSIA SE IL FILE È STATO MODIFICATO MANDA QUELLO NUOVO)
- LA RISPOSTA 304 NOT MODIFIED (E NON INVIA IL CONTENUTO DEL FILE)

IN QUESTA VERSIONE ABBIAMO UN METODO DI TIPO GET CHE PERÒ È UN GET CONDIZIONALE, OVERO CONDIZIONATO ALLA DATA LAST MODIFIED.

IL PROTOCOLLO HTTP È UN PROTOCOLLO SENZA STATO; OSSIA DOPO AVER MANDATO LA RISPOSTA, SI DIMENTICA DI CIÒ CHE HA FATTO

DA PARTE DEL CLIENT PERÒ, SE VOLESSI FARE UNA SERIE DI RICHIESTE CONSECUTIVE, SAREBBE MOLTO UTILE SAPERE COSA HA FATTO IL SERVER PRIMA.

E QUINDI DOPO AVER DEFINITO IL PROTOCOLLO SENZA STATO SI È CREATO QUESTO MECCANISMO DI DARE AL CLIENT UN **MECCANISMO CHIAMATO COOKIE**, CHE PERMETTE AL CLIENT DI TENERSI LA STORIA CON IL SERVER. COME FUNZIONA IL MECCANISMO DEI COOKIE?

IN LINEA DI PRINCIPIO FUNZIONA COSÌ:

SET-COOKIE: IL SERVER AD UN CERTO PUNTO DECIDE DI MANDARE L'HEADER SET-COOKIE, SEGUITO DA UNA STRINGA DI CARATTERI. QUELLO CHE VIENE SPECIFICATO DOPO QUESTO HEADER VIENE LETTO DAL CLIENT, IL CLIENT PRENDE NOTA DI QUESTO COOKIE: SE NON DEVE PIÙ CONTATTARE QUEL SERVER ALLORA QUEL COOKIE È STATO INUTILE, SE INVECE IL CLIENT DEVE EFFETTUARE ALTRE OPERAZIONI SUL SERVER, PER ESEMPIO MANDARE RICHIESTE CON ALTRI METODI GET, NEL MOMENTO IN CUI MANDA RICHIESTE AGGIUNGE L'HEADER SET-COOKIE. PRENDE LA STRINGA DI CARATTERI SUCCESSIVA E LA RIMANDA AL SERVER SOTTO FORMA DI COOKIE. LA VOLTA DOPO IL SERVER SI VEDE RITORNARE LA SEQUENZA CHE AVEVA DEFINITO LUI IN PRECEDENZA. È UN MODO PER RICORDARE AL SERVER QUELLO CHE HA FATTO PRIMA. IL CLIENT COSÌ TIENE TRACCIA DI QUELLO CHE IL SERVER ESEGUE PRIMA; IL CLIENT SE LO RICORDA E LO MANDA INDIETRO AL SERVER. **IL PROTOCOLLO È QUINDI SENZA MEMORIA, MA CON L'UTILIZZO DEI COOKIE CI PERMETTE DI TENERE MEMORIA DI QUELLO CHE FA IL SERVER**

IL COOKIE VIENE DEFINITO DA: UN NOME (STRINGA DI CARATTERI QUALSIASI) = VALORE (STRINGA DI CARATTERI QUALSIASI)

SET-COOKIE: NOME = VALORE

IL FATTO CHE I COOKIE ABBIANO UN NOME PERMETTE DI SETTARE PIÙ COOKIE.

OGNI SERVER GESTISCE I SUOI COOKIE IN MODO TOTALMENTE INDIPENDENTE DAGLI ALTRI SERVER: SE IL CLIENT SI CONNETTE CON VARI SERVER, LI RICEVE IN MANIERA AUTONOMA PER OGNI SERVER.

I COOKIE POSSONO SCADERE NEL TEMPO, POSSONO AVERE UN PERIODO DI VALIDITÀ E DOPODICHÉ SCADERE. SUCCEDDE CHE IL CLIENT MANDERÀ TUTTI I COOKIE CHE SI POSSONO USARE AL MOMENTO MENTRE NON INVIA I COOKIE SCADUTI.

QUESTA SCADENZA PUÒ ESSERE DEFINITA IN DIVERSI MODI:

- SOTTO FORMA DI DATA CON AD ESEMPIO IL COMANDO **COOKIE-EXPIRED**
- COOKIE VALIDO PER TOT MINUTI / TOT GIORNI, QUINDI **COOKIE A BREVE TERMINE**
- COOKIE VALIDO SOLO DURANTE QUESTA SESSIONE DI LAVORO

I COOKIE DI SESSIONE POSSONO ESSERE MEMORIZZATI IN UNA STRUTTURA DATI DEL BROWSER, QUINDI ALLOCATI IN MEMORIA RAM.

I COOKIE PERMANENTI DEVONO ESSERE MEMORIZZATI ALL'INTERNO DEL FILE SYSTEM DEL CLIENT.

COME LI DISTINGUO? GUARDANDO LA DATA DI SCADENZA, NEL SET COOKIE POSSONO ESSERCI PARAMETRI AGGIUNTIVI, TRA CUI LA DATA DI SCADENZA.

TRACCIAMENTO -> CIRCOSTANZA IN CUI IL SERVER HA INTERESSE DI IDENTIFICARE IL CLIENT

COOKIE DI TRACCIAMENTO: DO UN IDENTIFICATORE PER ESEMPIO UTD = 5