
Il protocollo HTTP (cont.)

Corso di ***Applicazioni Telematiche***

A.A. 2008-09 – Lezione n.2

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II

Facoltà di Ingegneria

HTTP: Response

- La risposta HTTP è un messaggio testuale formato da una riga iniziale, da header facoltativi ed eventualmente un body (corpo)

Version status-code reason-phrase CRLF

[Header]

CRLF

Body

dove:

[...] indica un elemento opzionale

CRLF indica la sequenza di caratteri di codice ASCII

13 (base 16) = 19 (base 10) → CR = Carriage Return

10 (base 16) = 16 (base 10) → LF = Line Feed

HTTP: Response (2)

Esempio:

HTTP/1.1 200 OK

Date: Thu, 10 Apr 2003 11:46:53 GMT

Server: Apache/1.3.26 (Unix) PHP/4.0.3pl1

Last-Modified: Wed, 18 Dec 2002 12:55:37 GMT

Accept-Ranges: bytes

Content-Length: 7394

Content-Type: text/html

<HTML> ... </HTML>

Status code

- Lo status code è un numero di tre cifre, di cui la prima indica la classe della risposta, e le altre due la risposta specifica
- Esistono le seguenti classi:
 - **1xx: Informational**
 - Una risposta temporanea alla richiesta, durante il suo svolgimento
 - **2xx: Successful**
 - Il server ha ricevuto, capito e accettato la richiesta
 - **3xx: Redirection**
 - Il server ha ricevuto e capito la richiesta, ma sono necessarie altre azioni da parte del client per portare a termine la richiesta
 - **4xx: Client error**
 - La richiesta del client non può essere soddisfatta per un errore da parte del client (errore sintattico o richiesta non autorizzata)
 - **5xx: Server error**
 - La richiesta può anche essere corretta, ma il server non è in grado di soddisfare la richiesta per un problema interno

Status code: esempi

- **100 Continue**
 - se il client non ha ancora mandato il body
- **200 Ok**
 - GET con successo
- **201 Created**
 - PUT con successo
- **301 Moved permanently**
 - URL non valida, il server conosce la nuova posizione
- **400 Bad request**
 - errore sintattico nella richiesta
- **401 Unauthorized**
 - manca l'autorizzazione
- **403 Forbidden**
 - richiesta non autorizzabile
- **404 Not found**
 - URL errato
- **500 Internal server error**
 - tipicamente un programma in esecuzione sul server ha generato errore
- **501 Not implemented**
 - metodo non conosciuto dal server

Gli header di risposta

- Gli header della risposta sono posti dal server per specificare informazioni sulla risposta e su se stesso al client
 - **Server**: una stringa che descrive il server: tipo, sistema operativo e versione
 - **Accept-ranges**: specifica che tipo di range può accettare (valori previsti: byte e none)
 - **WWW-Authenticate**: l'header di WWW-Authenticate include una challenge (codice di partenza) con cui il meccanismo di autenticazione deve fare match in caso di una risposta 401, (unauthorized). Il client genererà con questo valore un valore di autorizzazione posto nell'header *Authorization* della prossima richiesta.
-

Gli header generali

- Gli header generali si applicano solo al messaggio trasmesso e si applicano sia ad una richiesta che ad una risposta, ma non necessariamente alla risorsa trasmessa
- **Date**: data ed ora della trasmissione
- **MIME-Version**: la versione MIME usata per la trasmissione (sempre 1.0)
- **Transfer-Encoding**: il tipo di formato di codifica usato per la trasmissione
- **Cache-Control**: il tipo di meccanismo di caching richiesto o suggerito per la risorsa
- **Connection**: il tipo di connessione da usare
 - Connection: Keep-Alive → tenere attiva dopo la risposta
 - Connection: Close → chiudere dopo la risposta
- **Via**: usato da proxy e gateway

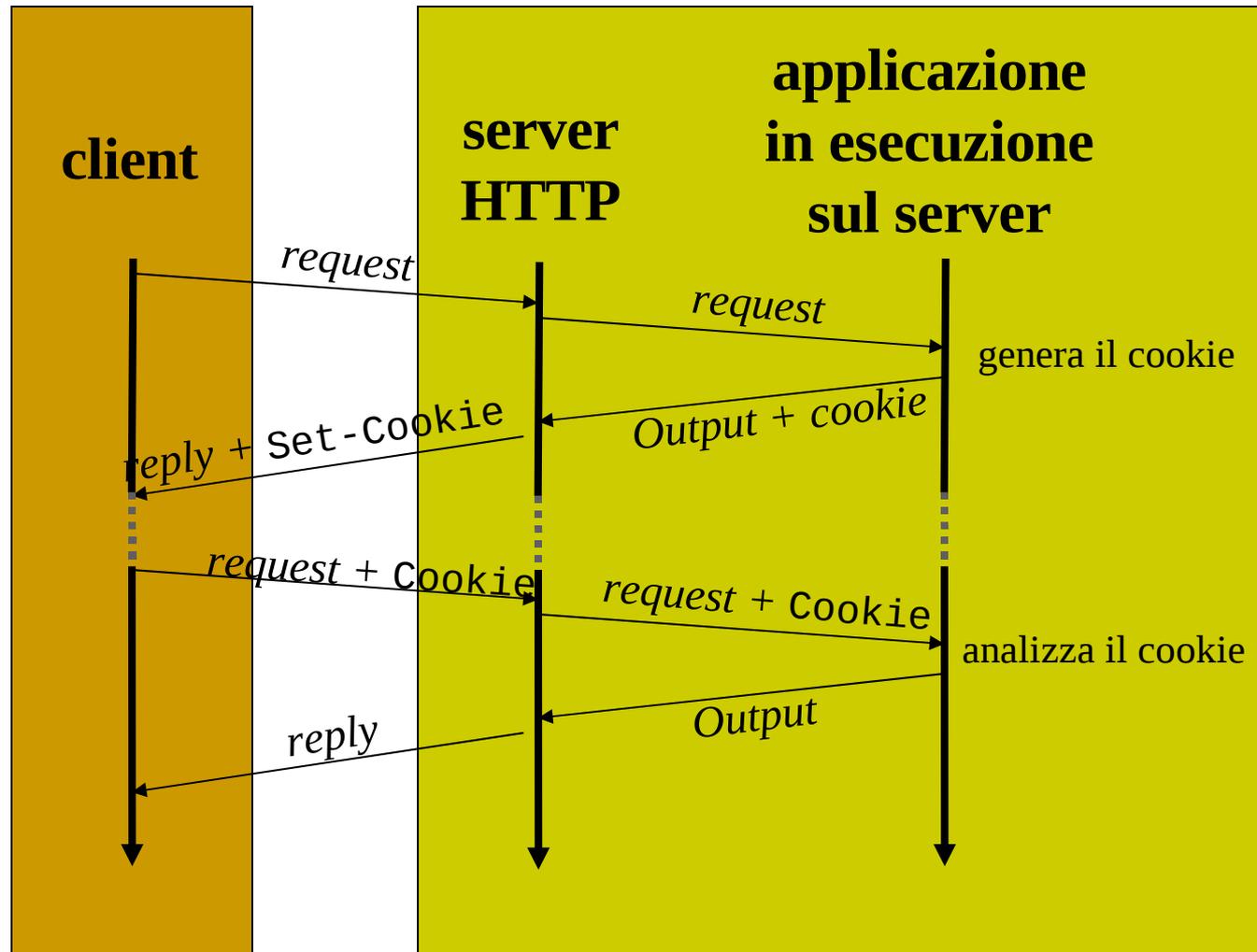
Gli header dell'entità

- Gli header dell'entità danno informazioni sul body del messaggio, o, se non vi è body, sulla risorsa specificata
 - **Content-Type:** il tipo MIME dell'entità acclusa
 - Specifica se è un testo, se un'immagine GIF, un'immagine JPG, un suono WAV, un filmato MPG, ecc...
 - Obbligatorio in ogni messaggio che abbia un body
 - **Content-Length:** la lunghezza in byte del body
 - Obbligatorio se la connessione è persistente
 - **Content-Base, Content-Encoding, Content-Language, Content-Location, Content-MD5, Content-Range:** l'URL di base, la codifica, il linguaggio, l'URL della risorsa specifica, il valore di digest MD5 e il range richiesto della risorsa
 - **Expires:** una data dopo la quale la risorsa è considerata non più valida (e quindi va richiesta o cancellata dalla cache)
 - **Last-Modified:** la data e l'ora dell'ultima modifica
 - Serve per decidere se la copia posseduta (es. in cache) è ancora valida o no
 - Obbligatorio se possibile
-

I cookies

- **HTTP è stateless**: il server non è tenuto a mantenere informazioni su connessioni precedenti
 - Un cookie è una breve informazione scambiata tra il server ed il client che serve a mantenere un'informazione di stato tra una richiesta e le successive
 - Tramite un cookie il client mantiene lo stato di precedenti connessioni, e lo manda al server di pertinenza ogni volta che richiede un documento
 - Esempio: tramite un cookie si viene rediretti sulla pagina in Italiano tutte le volte che ci si ricollega allo stesso server
 - I cookies furono inizialmente proposti da Netscape e successivamente definiti in RFC 2109 ed in RFC 2965
-

Cookies (2)



Cookies: header specifici

- I cookies dunque usano due header: uno per la risposta, ed uno per le richieste successive:
 - **Set-Cookie**: header della risposta
 - il client può memorizzarlo (se vuole) e rispedito alla prossima richiesta
 - **Cookie**: header della richiesta
 - il client decide se spedito sulla base del nome del documento, dell'indirizzo IP del server, e dell'età del cookie
 - Un browser può essere configurato per accettare o rifiutare i cookies
 - Alcuni siti web richiedono necessariamente la capacità del browser di accettare i cookies
-

HTTP: entità intermedie (1)

- Il protocollo HTTP prevede l'esistenza di entità intermedie nella catena di distribuzione server-client
 - **Proxy**
 - Sono entità in gradi di agire sia da server (nei confronti dei client) che da client (nei confronti dei server origine)
 - Possono rispondere direttamente ai client oppure attraverso una interazione con il server di origine
 - **Gateway**
 - Un gateway è un'entità che agisce nei confronti del client come se fosse il server di origine
 - Può essere utilizzato per accedere tramite HTTP a servizi di diversa natura (non veicolati tramite HTTP)
-

HTTP: entità intermedie (2)

■ Tunnel

- Entità che inoltrano in maniera trasparente (cioè senza interpretarli) messaggi del protocollo HTTP

■ Cache

- Entità intermedia che memorizza localmente i messaggi di risposta e successivamente li riutilizza per generare altri messaggi di risposta evitando ulteriori interazioni con il server di origine
 - Entità cache possono essere dislocate nel client (browser), nel server o in proxy cache
-

Tipi di proxy HTTP

- In generale un proxy si pone come intermediario tra client e server e decide se e come rispondere al client
- Proxy cache
 - Richieste multiple agli stessi URL possono essere salvate in una locazione intermedia per una maggiore efficienza nella gestione delle risposte
- Proxy di filtro
 - Esigenze di sicurezza o di controllo degli abusi di una rete possono richiedere l'effettiva esecuzione della richiesta solo in certi casi, e altrimenti la risposta con un generico messaggio di mancata autorizzazione.
- Intermediari
 - Un proxy trasparente esegue tutte le richieste e fornisce tutte le risposte, ma in certi casi può convertire o modificare la risposta.
 - Ad esempio fornire link a vocabolari, togliere i banner, convertire i formati ignoti, ecc.
 - Ad esempio, WBI di IBM (<http://www.almaden.ibm.com/cs/wbi/>)