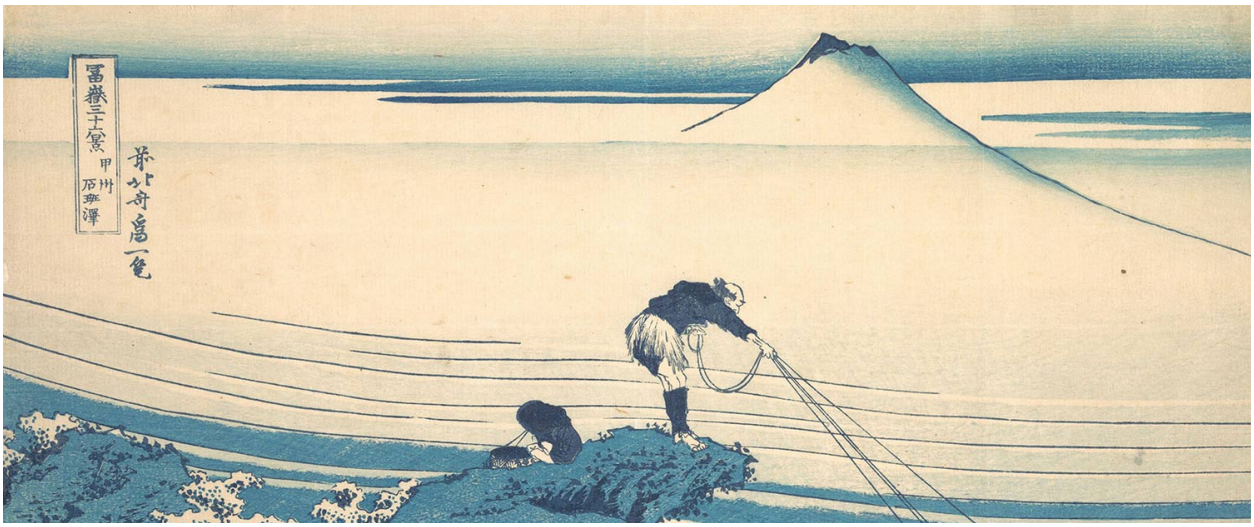


Esercizio HTTP

Tags HTTP

09102024_HTTP_laboratorio_rev2.pdf

IMPORTANTE

Esercizio 1

Rispondi alle seguenti domande:

1. Qual è il metodo HTTP utilizzato per la richiesta principale?
2. Quale status code è stato restituito dal server? Cosa indica questo codice?
3. Quali header di richiesta e di risposta sono presenti?
4. Il browser ha richiesto risorse aggiuntive (CSS, immagini, script)? Quali?

5. Cosa succede se clicchi su un link o ricarichi la pagina? Noti cambiamenti nelle richieste?

6. C'è qualche differenza tra le richieste GET e POST che riesci a identificare?

Url completo della risorsa alla quale sto cercando di accedere:
https://me.stecca.dev/

1.

Il metodo usato per accedere alla risorsa principale è il metodo GET

2.

Il codice che mi è stato restituito dal server è → 304 (Not Verified), questo codice viene usato dal server nel momento in cui disponiamo già della risorsa richiesta all'interno della cache del browser.

3.

Response Headers:

- **Age:** Indica quanto tempo è passato da quando la risposta è stata generata dal server di origine.
- **Cache-Control:** Specifica le direttive per i meccanismi di caching in entrambe le direzioni della richiesta/risposta.
- **Date:** La data e l'ora in cui il messaggio è stato originato.
- **Etag:** Un identificatore univoco per una specifica versione di una risorsa.
- **Expires:** La data/ora dopo la quale la risposta è considerata obsoleta.
- **Vary:** Indica come determinare se una risposta in cache corrisponde a una richiesta.
- **Via:** Informa sul proxy attraverso cui è passata la risposta.
- **X-Cache:** Indica se la risposta è stata servita da una cache.
- **X-Cache-Hits:** Il numero di volte che una risposta è stata servita da una cache.

- **X-Fastly-Request-ID:** Un identificatore univoco per la richiesta, specifico di Fastly CDN.
- **X-Served-By:** Identifica il server che ha servito la risposta.
- **X-Timer:** Fornisce informazioni sui tempi di elaborazione della richiesta.

Request Header:

- **authority:** Specifica il dominio del server a cui si sta facendo la richiesta (in questo caso, me.stecca.dev).
- **method:** Indica il metodo HTTP utilizzato per la richiesta (GET in questo caso).
- **path:** Specifica il percorso della risorsa richiesta sul server (/ indica la root).
- **scheme:** Indica il protocollo utilizzato (https in questo caso).
- **accept:** Specifica i tipi di contenuto che il client è in grado di comprendere.
- **accept-encoding:** Indica i metodi di compressione supportati dal client.
- **accept-language:** Specifica le preferenze linguistiche del client.
- **cache-control:** Definisce le direttive per il caching della richiesta.
- **if-modified-since:** Chiede al server di inviare la risorsa solo se è stata modificata dopo la data specificata.
- **if-none-match:** Utilizzato per la validazione della cache, confronta l'ETag fornito con quello della risorsa sul server.
- **priority:** Indica la priorità della richiesta.
- **sec-ch-ua, sec-ch-ua-mobile, sec-ch-ua-platform:** Forniscono informazioni sul browser e sul dispositivo del client.
- **sec-fetch-*:** Una serie di header che forniscono informazioni sul contesto della richiesta.
- **upgrade-insecure-requests:** Indica al server che il client preferisce connessioni sicure.
- **user-agent:** Identifica il software del client che sta effettuando la richiesta.

4.

Il browser ha richiesto molte risorse aggiuntive, essendo un sito costruito come portfolio contiene molte immagini e contenuti i quali vengono mandati dal server come risorse aggiuntive nel momento in cui le vogliamo visionare singolarmente come per esempio script javascript con il fine di rendere il sito dinamico e tutte le foto rappresentate con formato webp che è il formato usato per rappresentare foto tramite web

5.

Nel momento in cui andiamo a ricaricare la pagina le richieste che andremo a fare e ricevere saranno le stessa dato che stiamo andando ad accedere alla stessa risorsa mentre nel momento in cui clicchiamo su un link accederemo ad una risorsa diversa da quella precedente e di conseguenza cambieranno anche le richieste ad essa connesse.

6.

Come da definizione nel momento in cui andiamo ad effettuare una richiesta GET stiamo solamente richiedendo una risorsa per visualizzarla mentre nel momento in cui effettuiamo una POST dobbiamo per forza inviare dei dati annessi ad essa e di conseguenza come possiamo visualizzare anche dal campo payload mandiamo dei dati

ESERCIZIO 2

Due risposte plausibili del server HTTP (solo la status line):

1. **Risorsa non modificata:**

HTTP/1.1 304 Not Modified

2. **Risorsa trovata e mostrata**

HTTP/1.1 200 OK

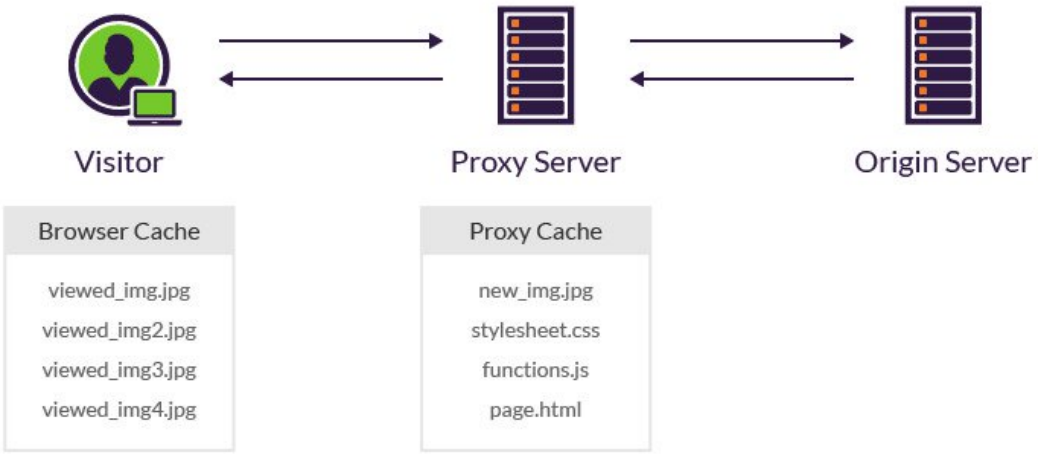
3. Risorsa non trovata

HTTP/1.1 404 NOT FOUND

4. Risorsa Proibita

   HTTP/1.1 403 FORBIDDEN

Comportamento del Proxy:



Quando una richiesta come quella sopra viene inviata attraverso un proxy, il comportamento del proxy varia a seconda della configurazione e delle policy di caching. Ecco due scenari possibili:

1. **Proxy con caching abilitato:**

Il proxy potrebbe aver memorizzato nella sua cache una copia della risorsa richiesta in una precedente richiesta → Cache Hit, mentre nel caso nella cache

non fosse presente la risorsa della quale abbiamo bisogno → Cache Miss.

Quando riceve la richiesta dal client, il proxy verifica prima nella propria cache.

Se il proxy ha la risorsa nella cache ma è incerto sulla sua validità (a causa dell'intestazione "If-Modified-Since"), inoltra la richiesta al server originario.

Se il server risponde con **304 Not Modified**, il proxy invia al client la risorsa dalla sua cache.

1. Se il server risponde con **200 OK**, il proxy aggiorna la sua cache con la nuova versione della risorsa e la invia al client.

1. **Proxy senza caching o con caching disabilitato:**

In questo caso, il proxy inoltra la richiesta direttamente al server, comportandosi come un semplice intermediario.

Il server risponderà direttamente al proxy con uno dei due codici di stato (**304** o **200**), e il proxy trasmetterà la risposta al client senza fare ulteriori elaborazioni.

Esercizio 3: Calcolo del tempo di download

Dati:

- Numero di oggetti: N :11 (1 file HTML + 10 oggetti)
- Dimensione di ogni oggetto: L = 200 kbit
- Capacità del collegamento: C = 100 kbit/s
- Dimensione messaggi di controllo: m = 100 bit
- Ritardo di propagazione: τ = 100 ms = 0.1 s

1. Connessioni TCP parallele non persistenti:

Velocità di trasmissione per connessione: r = C / N = 100 / 11 ≈ 9.09 kbit/s

Tempo di trasferimento per oggetto: Toggetto = L / r = 200 / 9.09 ≈ 22 s

Tempo totale: Ttotale = Toggetto + RTT = 22 + 0.1 = 22.1 s

2. Connessioni TCP seriali non persistenti:

Tempo di trasferimento per oggetto: toggetto = L / C = 200 / 100 = 2 s

Tempo totale: Ttotale = N *(Toggetto + RTT) = 11* (2 + 0.1) = 24.1 s

Esercizio 4: Condivisione del collegamento e connessioni TCP

Dati:

- Numero di oggetti: 12 (1 file HTML + 11 oggetti)
- Dimensione di ogni oggetto: L = 50 kB = 400 kbit
- Capacità del collegamento: C = 1 Mbit/s = 1000 kbit/s
- RTT: 150 ms = 0.15 s
- Numero totale di flussi: n = 10 → Numero flussi utili → 9

Velocità effettiva per il flusso HTTP: r = C / n = 1000 / 10 = 100 kbit/s

1. Singola connessione TCP persistente:

Tempo di trasferimento per oggetto: Toggetto = L / r = 400 / 100 = 4 s

Tempo totale: ttotale = (12 *toggetto) + RTT = (12* 4) + 0.15 = 48.15 s

2. Connessioni TCP parallele non persistenti:

Tempo di trasferimento per oggetto: Toggetto = L / r = 400 / 100 = 4 s

Tempo totale: Ttotale = Toggetto + RTT = 4 + 0.15 = 4.15 s

Esercizio 5: HTTP Caching

Dati:

- Cache hit rate: P = 0.4
- Cache miss rate: Q = 0.6
- Dimensione pagina web: L = 100 kB = 800 kbit

- Capacità del collegamento client-proxy: C = 1 Gb/s = 1,000,000 kbit/s
- Capacità del collegamento proxy-server: c = 100 Mb/s = 100,000 kbit/s

Tempo di risposta per cache hit: Thit = L / C = 800 / 1,000,000 = 0.0008 s

Tempo di risposta per cache miss: Tmiss = (L / c) + (L / C) = (800 / 100,000) + (800 / 1,000,000) = 0.0088 s

Ritardo medio: Tmedio = P *Thit + Q* tmiss = 0.4 *0.0008 + 0.6* 0.0088 = 0.00560 s = 5.60 ms

Quindi, il ritardo medio sperimentato dal generico client è di 5.60 millisecondi.