

Domande Tec Web 7-02-24
(GDM)

- HTTP 1.1 w pipelining
- Long polling
 - ho il problema del timeout?
- Com'è fatta una HTTP request?
 - che fa il PUT?
- Che son gli ipertesti? node
- che protocolli a livello di trasporto nel WWW?
- Tag a: anchor. Che succede sotto?
- HTTP - HTML - URI: uri?
 - url vs urn
 - é opzionale il protocollo nell'url?
- Che si fa col Nome logico? Come si trasforma? (non in questa materia) DNS
- Cosa offre una Web Socket rispetto ai suoi predecessori?
 - Complica il meccanismo di comunicazione? (Pros) & (Cons) é un meccanismo async
 - Come si attiva una richiesta di web socket? (protocollo) su http come specifico che voglio l'upgrade? (header)
- Ragionamento per sviluppo in react
- http response message, cosa ci troviamo?
 - content/type
- autenticazione in http
- TLS cosa garantisce? come?
 - come viene garantita l'autenticità? certificati
- Come scalare un'applicazione web? (struttura verticale - orizzontale / n-tier)
 - struttura web app?
 - 3 tier: Web Server / Application Server / DB. Come si replica?
- js
 - a. quanti clienti e quanti server
 - b. Tipi di variabili : tipi valore e tipi riferimento
- Cosa si intende per ereditarietà in CSS → quali proprietà non vengono ereditate? (box model)
- Cache: freshness
- Cache lato client vs lato server (proxy cache)
- Proxy e gateway
- Se ho già cache, a cosa mi serve avere anche il proxy?
- Cos'è html e come si usa, differenze con SGML
- Come gestisco stato in app web (cookie o stato salvato sul server)
 - campi nei cookie
 - cosa succede in un caso e nell'altro se c'è crash del server
- Perché usiamo ajax?
- Content-type ? (Non ho sentito esattamente cosa ha chiesto)
- Architettura a tre livelli, problemi possibili con la replicazione (clustering, problema di replicare App server per stato di sessione e database per dati)

- varie versioni http → superiamo mai half duplex in http? (No) come lo superiamo (websocket), esempio di app che usa websocket (multi chat)
- Https (autenticità, integrità, Confidenzialità e come vengono implementate)
- Ereditarietà react
- Come funziona node.js, event loop
- Servlet single threaded vs multi threaded

(... non si sente ma si sta segnando il ragazzo le domande)

- Include vs forward
- Servlet e jsp possono accedere ai beans creati dalle altre?
- Cosa c'è in head html
- Caching
- Tipizzazione è forte in java? E in js?
- Json, xml (formati per trasmissione dati)
- Adattare pagine a dispositivi diversi (nell'header richiesta ho Dispositivo che fa richiesta: user-agent) Cosa c'è in header e body request http

(PB)

- EJB
 - a. come è gestita la concorrenza nei session bean stateless? (Resource pooling) per quanto permangono?
 - b. E nei session bean stateful? (Activation/passivation)
- Distribuzione servizi (orizzontale vs verticale e problemi)
- Servlet
 - a. Ciclo di vita (modello normale e deprecato)
 - b. Nel modello normale cosa succede per variabili di istanza? Ho problemi di concorrenza (si)
 - c. Se invece si tratta di variabili di classe? (No)
 - d. E nel modello deprecato? (viceversa)
- Validation della cache: descrizione generale, come capisco se ho versione aggiornata di pagina? (Nell'header http ho last-modified)
 - MVC
 - a. Processamento dei dati chi lo fa?
 - b. in MVC? (servlet dispatcher) (dispatch iniziale del container - poi controller)
 - c. è la dispatcher servlet sui dati?
 - JSP -> JSF: regole di navigazione (regole di navigazione tra pagine)
 - Dependency Injection: inversion of control - .
 - a. Quando il container invoca metodi setter / getter
 - b. error handling
 - HTTPS vs HTTP
 - a. Integrity Confidenzialità Autenticità
 - b. TLS
 - c. Header di richieste HTTPS?
 - WebSocket protocollo
 - a. HTTP protocollo (GET/POST). cosa fare per aprire una connessione ws? Header per upgrading
 - Cookie, che informazioni ci possiamo tenere?
 - Javascript / JSON

- sintassi JSON
 - a. object / array
 - b. che librerie lato server?
 - c. Gson .fromJson come fa?
- EJB - Message Driven Beans: a cosa servono? Cosa danno più?

(...)

- HTTP evoluzione overview - protocollo evoluzione
- Di cosa si parla in letteratura per Java Model 2?
 - a. Session Bean / Entity Beans / EJB container
 - b. Con cosa fai la View?
- JSF - che operazioni svolge il (server) all'arrivo di una richiesta?
 - a. navigation rule
 - b. quando avviene la validazione dei dati in input nel ciclo di vita?
- Cosa si intende per interazione "Forever Response" come schema di interazione in HTTP?
- una JSP può includere dei Java Bean che possono usare diversi scope parlane:
 - a. *Risposta: (application / sessione / richiesta / page)*
 - b. perché conviene, questa suddivisione? come funziona: (ex una istanza per richiesta)
- Dependency Injection in : a livello di programmazione.
 - a. *Risposta: file di configurazione, come fare injection (set/costruttore), Bean factory (get bean)*
 - b. Se la risorsa che viene iniettata ha a sua volta dipendenze? Cosa avviene?
 - c. Get Bean cosa avviene? Perché è conveniente?
- Qual è il ciclo di vita di una JSP - cosa avviene dopo il deploy?
 - a. Per mandare una nuova versione sul server? Bisogna rifare il .war?
 - b. (in esecuzione è come se fosse una servlet) ...
- AJAX, cosa succede se (per errore / o meno) faccio consecutivamente due send?
- Cookie - che elementi di informazione vi sono contenuti?
- Com'è fatta la gestione della concorrenza di richieste ad un session bean con stato (EJB)? (*Container: Activation & Passivation*)
 - Quando scatta la passivation?
- Protocollo HTTP, una richiesta di tipo HEAD? E a cosa può servire? (*se pagina c'è / non c'è / data di modifica*)
- Che altro tipo di richiesta posso fare per vedere se il server è connesso alla rete e funzionante? (TRACE)
 - Cosa fa TRACE? (*ping di rete: manda una richiesta e servitore deve rispondere con lo stesso messaggio*)
- Componente come viene gestita la concorrenza?
- Nel Web possono esistere intermedi (Web Caching) come si mantengono aggiornate
 - ci son 3 categorie di modalità. Quali?
 - Esempi delle 3.
 - Una nuova request invalida la cache?