WEB SOCKET:

Partiti dal WEB (Guardare Pagine da BROWSER) Basato su HTTP

→ Può Contenere:
- CSS (Cascade Style Sheet)
- Javascript
-

Lavorare Sull' Aspetto Grafico ← e NON Sulla STRUTTURA

può Arrivare :
- EMBEDDED
- file Separato ?

Codice che Viene Eseguito FRONT-END che lavora Sui DATI della Pagina WEB vista COME DOM

EVENTI SINCRONI/ASINCRONI

↳ Senza Tempo, NON si Sa Quando Arrivano

↳ Possono Arrivare:
- Dall' UTENTE
- dalla RETE

ho finito di Scaricare un file ←

JAVASCRIPT per Eseguire CODICE Lato FRONT-END

Entrambe Eseguite sul SERVER:

CGI vs WEB SERVICES

→ Rivolto al WEB Tradizionale dove Voglio una PAGINA DINAMICA

È Solo BACK-END ⇐ Utilizzo del WEB Come Protocollo di Livello Trasporto per Chiamate a Funzione

⇓

- Load Balancing
- MicroServizi (PER)
- SoA

IMPORTANTE CAPIRE COSA STA SUL BACK-END e Cosa FRONT

Cosa COLLEGA Tutti Questi PEZZI? Protocollo HTTP

Sdoppiare Processo SERVER per Troppo SOVRACCARICO può essere fatto se è Stato Ben Modularizzato Altrimenti NON è dividibile e Quindi dovrei CLONARE La funzione.

Mio OBIETTIVO è Avere TANTI MicroServizi (NON ci devono essere Troppa DIPENDENZA Tra i DATI).

WEB SOCKET:

Non Centra Nulla con HTTP o WEB SERVICES...

Siamo Nel mondo WEB, Quindi BROWSER che guarda una pagina generata da un SERVER HTTP.

HTTP NON può Aggiornare la Pagina Se tu NON lo Richiedi perchè si Basa su RICHIESTA e RISPOSTA.

Per Questo c'è la NECESSITÀ/VOLONTÀ di Cambiare il lvl APPLICAZIONE e Quindi Si SOSTITUISCE HTTP per WEB SOCKET.
↳protocollo lvl Applicazione

Sotto HTTP, c'è TCP (SYN, SYN ACK, ACK)... e dopo Aver instaurata una CONNESSIONE NON c'è Nessun Vincolo riguardo a chi può "PARLARE" per PRIMO o meno

TCP è BIDIREZIONALE (da NON confondere con Quello che Abbiamo deciso NOI di Costruire SOPRA, ovvero CLIENT-SERVER)

Connessione TCP Rimane Sempre Attiva durante il Cambiamento da HTTP a WEB SOCKET
↳peer-To-Peer

Se ci si parla in WEB SOCKET non c'è BISOGNO di FARE POLLING ma il BACK-END che fa il "PUSH" su Browser per Aggiornarlo con la Pagina Più RECENTE.

TCP è una CONNESSIONE che Rimane Sempre Attiva e NON viene Chiusa e Ri-Aperta

NAT ⇒ Network Address Translation ⇒ Salva un Secco di IP e c'è una Sicurezza Implicita con gli INDIRIZZO IP.

È però una SPORCIZIA... ma ci Sono più Vantaggi che Svantaggi

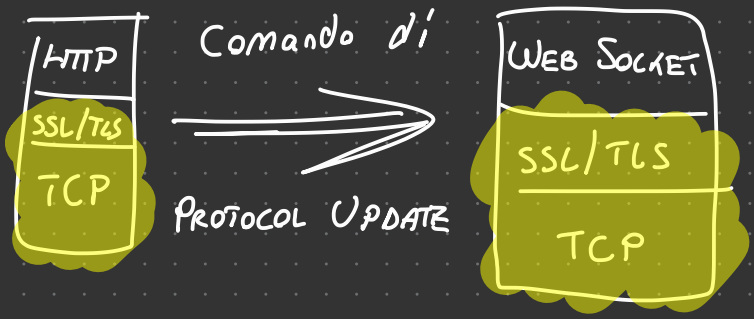
Se però NON ci fosse una CONNESSIONE TCP NON Saprei a chi Mandarlo.

Sfrutta le PORTE ed è Qui che Viene Mescolato con Lvl 3

NAT NON può Tenersi TUTTO in Cache e Quando Vede che NON Passa più Traffico Allora se NON passa più Traffico allora Viene Killata.

Messaggio KEEP ALIVE, Periodicamente mandati per TENERE Viva la RIGA Nella CACHE del NAT

WEB SOCKET SICURO ⇒ Tutto Uguale ma ci Si Appoggia



Rimane TUTTO Invariato, Cambia Solo il Modo in cui si Parlano ma è TUTTO Uguale il RESTO

NODEJS ⇒ Back-End che si Programma con JAVASCRIPT

↳ frameWork lato BACK-END

↳