**Cosa si intende per URL (Uniform Resource Locator)?**

* Protocollo
* IP:porta
* path host e path risorsa

**Protocollo HTTP: definizione e caratteristiche**

Il protocollo HTTP è un protocollo del livello applicativo del modello TCP/IP delle reti di telecomunicazioni considerato stateless, dato che non mantiene informazioni sulle richieste precedenti del client. Tale protocollo ha le seguenti caratteristiche:

Il client inizia una connessione TCP (crea una socket) verso il server sulla porta 80 e il server accetta la connessione TCP dal client;

Le coppie nome-valore nell’header servono per il payload.

**Quali sono le due tipologie di messaggi HTTP e che struttura hanno?**

request e response, ma hanno la stessa struttura:

* start-line obbligatorio: protocollo e altre informazioni
* header lines opzionale: formato da coppie nome-valore
* payload opzionale: il contenuto vero e proprio che vogliamo mandare.

**Codici di stato response: cosa e quali sono?**

**Vengono usati perché non c’è memoria condivisa.**

* 1xx: ricevuto
* 2xx: conferma
* 3xx: azione aggiuntiva
* 4xx: richiesta non valida
* 5xx: errore, nonostante la richiesta valida

**Quali sono i tipi di connessione e sincronizzazione?**

Una connessione può essere:

* sincrona
* asincrona: no collegamento contemporaneo degli interlocutori

Una sincronizzazione può avvenire in 3 punti:

* request submission: il client invia un messaggio che viene preso in carico dal middleware;
* request delivery: viene poi trasferito al middleware ricevente;
* request processing: o passa al server che tratta il messaggio ricevuto.

Questa si divide in 3 ulteriori fasi: a) ricevuto; b) letto; c) processato.

Nelle varie fasi, i buffer utilizzati tracciano i messaggi che permettono la sincronizzazione. La sincronizzazione può essere:

* transiente: il destinatario non è connesso e i dati vengono scartati;
* persistente: il middleware conserva i dati fino alla consegna del messaggio al destinatario e non è necessario che i processi siano in esecuzione prima e dopo l’invio/ricezione dei messaggi.

**Quali sono le tipologie di comunicazione?**

* **comunicazione persistente asincrona**: il middleware mantiene il messaggio fin quando A non si connette e riceve il messaggio; B può anche non essere immediatamente attivo e ritorna il controllo;
* **comunicazione persistente sincrona**: A attende l’accettazione da parte del middleware di B, quindi A è bloccato
* **comunicazione transiente sincrona:**
* **comunicazione transiente asincrona**: A invia SSE B è pronto a ricevere. Se no i dati vengono scartati.
* **comunicazione transiente sincrona su ritorno**: A invia ma B non processa subito, manda un ack, e processa più avanti. Fa diventare persistente la comunicazione, se il sistema non lo fa.
* **comunicazione transiente sincrona basata su consegna**: uguale a prima, ma deve aspettare di ricevere un ACK quando B inizia a processare
* **comunicazione transiente sincrona basata su risposta**: uguale a prima ma A aspetta finchè non arriva l’ACK che B ha finito di processare.

**Cosa si intende per message – queueing model?**

Si dice message – queueing model un meccanismo che offre capacità di archiviazione senza richiedere che il mittente o destinatario siano attivi durante la trasmissione di messaggi. A seconda dello stato di sender e receiver, ovvero running o passive, le 4 primitive sono:

→ put: aggiunge un messaggio ad una coda;

→ get: blocca fino a quando la coda specificata non è vuota e rimuove il primo messaggio;

→ poll: controlla una coda specifica e rimuove il primo messaggio senza bloccare;

→ notify: installa un gestore da chiamare quando un messaggio viene inserito in una coda.

**Cosa sono il message broker e il PaS Pattern?**

Il message broker è un meccanismo che permette il pattern -> publish & subscribe. Ha il compito di ricevere dati da una sorgente e condividerli su un dato canale. Il PaS Pattern è un disaccoppiamento tra publisher e subscriber:

→ publisher: si registrano dichiarando di voler inviare messaggi su un determinato argomento;

→ subscriber: si registrano dichiarando di voler ricevere messaggi su un determinato argomento. Non conoscono l’indirizzo del publisher.

Tra i due fattori si ha un disaccoppiamento, quindi indipendenza e concorrenza, con scalabilità. Il broker può mantenere i dati e QUINDI LO STATO nel tempo, a seconda di diverse politiche.