Sono (quasi) come una telefonata permanente tra client e server.  
**WebSocket** è un protocollo che, dopo un breve “saluto” via HTTP (handshake), apre una **connessione unica, full-duplex e persistente**: da lì in poi **entrambi** possono inviarsi messaggi **in tempo reale**, senza dover fare una richiesta ad ogni aggiornamento (come avviene con HTTP classico, polling o long-polling).

**Perché usarli**

* **Realtime**: chat, notifiche live, trading, dashboard, giochi multiplayer, collaborazioni tipo Google Docs.
* **Efficienza**: meno overhead rispetto a tante richieste HTTP.
* **Bidirezionale**: se arrivano dei messaggi client e server ascoltano entrambi. Sia il fe può chiamare il be, che il be può mandare messaggi al fe.

**Come funziona (in breve)**

1. Il client fa una richiesta HTTP GET con header speciali (Upgrade: websocket).
2. Se il server accetta, la connessione si “promuove” a WebSocket.
3. Da quel momento si scambiano **frame** (testo o binario) su quella stessa TCP, finché una parte chiude.

**Esempio con Node.js**

**Server (Node.js)**

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

**Client (browser)**

Immagine che contiene testo, Carattere, linea, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.  
  
**Con Socket.IO (più alto livello)**

Socket.IO usa WebSocket **quando possibile** e fa fallback (es. polling – interrogare continuamente il server con una richiesta) se serve, aggiunge stanze, riconnessione, ACK, ecc.

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene testo, Carattere, schermata, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.