SignalR hoạt động được **real-time** là nhờ vào việc **duy trì kết nối hai chiều (bi-directional connection)** giữa client và server. Sau đây là phần giải thích chi tiết:

**Bản chất của SignalR là gì?**

SignalR là một **thư viện real-time communication** của ASP.NET Core, cho phép **client và server gọi hàm của nhau** thông qua **kết nối liên tục**.

**Tại sao SignalR đáp ứng được real-time?**

SignalR **không dùng HTTP truyền thống** (request/response) theo kiểu "hỏi rồi trả lời", mà dùng **các kỹ thuật dưới đây để duy trì kết nối liên tục**:

**1. WebSockets (ưu tiên)**

* **Giao thức full-duplex** chạy trên TCP.
* Cho phép **gửi và nhận dữ liệu liên tục** giữa client và server mà không cần tạo lại kết nối.
* **Tốc độ cao, ít overhead.**
* SignalR sẽ **ưu tiên WebSocket** nếu trình duyệt và máy chủ hỗ trợ.

**2. Server-Sent Events (SSE) (chỉ từ server → client, không hỗ trợ IE)**

* Kết nối đơn chiều, server có thể **push dữ liệu** tới client.
* Dễ dùng nhưng **không hỗ trợ 2 chiều** và bị giới hạn bởi trình duyệt.

**3. Long Polling (fallback nếu không dùng được WebSocket/SSE)**

* Client liên tục **gửi request** đến server.
* Nếu server chưa có dữ liệu, nó giữ request lại một thời gian, sau đó trả về hoặc timeout.
* **Kém hiệu quả hơn**, nhưng là **giải pháp đảm bảo tương thích tối đa.**

📌 **SignalR tự động chọn giải pháp tốt nhất**: WebSocket > SSE > Long Polling.

**Real-time khác với HTTP như thế nào?**

| **Yếu tố** | **HTTP truyền thống** | **SignalR (WebSocket)** |
| --- | --- | --- |
| Kết nối | Tạo mới mỗi request | Kết nối giữ nguyên |
| Giao tiếp | Một chiều (Client → Server) | Hai chiều |
| Tốc độ | Chậm hơn (có header, handshake) | Nhanh, nhẹ |
| Thực hiện Push | Không | Có thể push từ server |
| Tình huống phù hợp | Giao tiếp tạm thời | Chat, thông báo, game, cập nhật dữ liệu realtime |