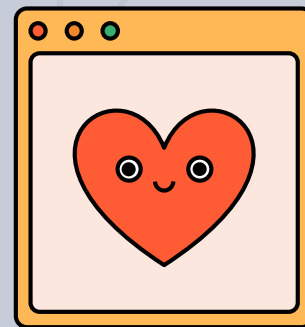


SignalR in Asp.net core



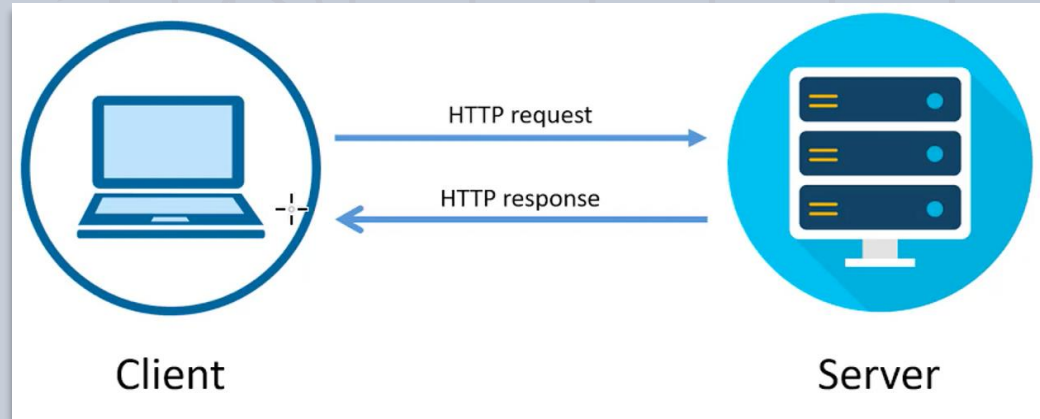
Traditional Server-Client Communication

- **Request-Response Model:**

Clients send a request, servers respond with data.

- **Challenges:**

- High latency due to repeated requests.
- Inefficient for real-time applications like chats or live notifications.



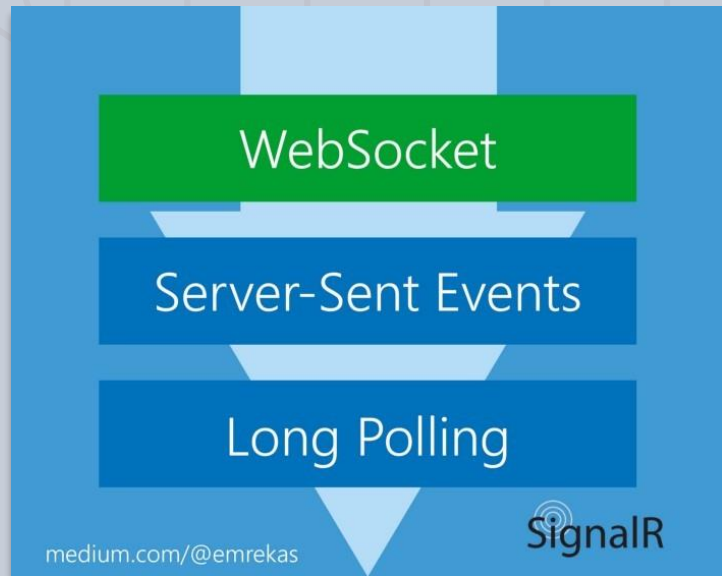


Real-Time Communication Methods

Long Polling: Client holds connection until data is available, then reconnects.

Server-Sent Events: Server pushes updates to the client but only supports one-way communication.

WebSocket: Full-duplex communication between client and server.



What is SignalR?

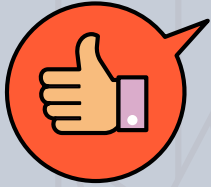


•**Definition:** A powerful library for building real-time web applications in ASP.NET Core.

•**Key Features:**

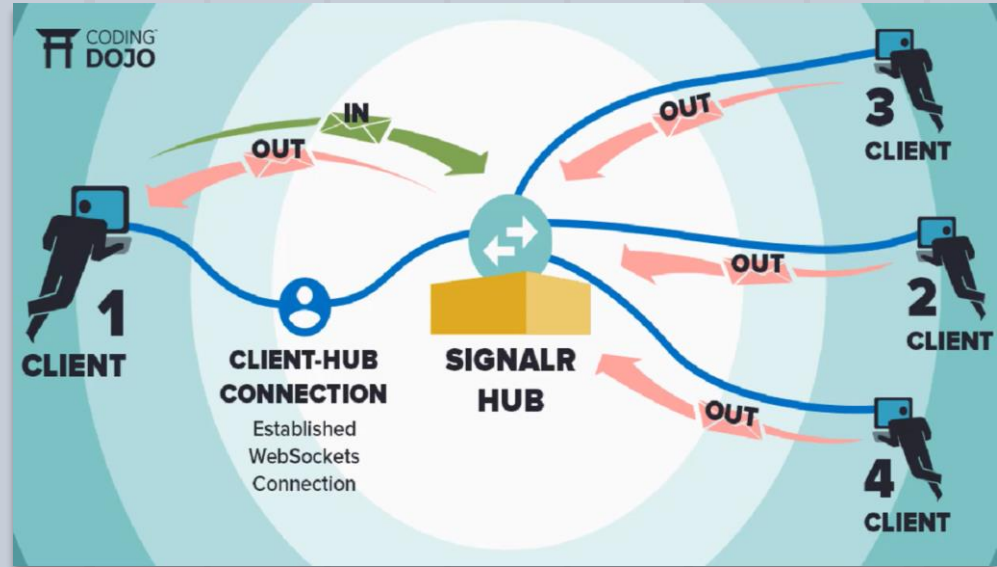
- Real-time, bi-directional communication between server and client.
- Supports various transport mechanisms (WebSockets, Server-Sent Events, Long Polling).
- Simple API for easy integration into your applications.
- Scalable architecture for handling large numbers of clients.





How SignalR works ?

- **Hubs:** The core component of SignalR, allowing clients to invoke methods on the server and vice versa.
- **Clients:** Represent connected clients and can receive messages from the server.
- **Servers:** Handle incoming requests, process data, and broadcast messages to clients.
- **Transport Mechanisms:** Explain the different ways SignalR can establish connections and transmit data.





Use cases & benefits of signalR

Use cases	benefits
<ul style="list-style-type: none">• Real-time chat applications• Collaborative tools Live• dashboards and monitoring• Real-time notifications	<ul style="list-style-type: none">• Enhanced user experience with real-time updates.• Efficient communication between server and clients.• Scalability to handle a large number of concurrent users.

