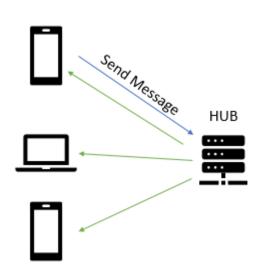


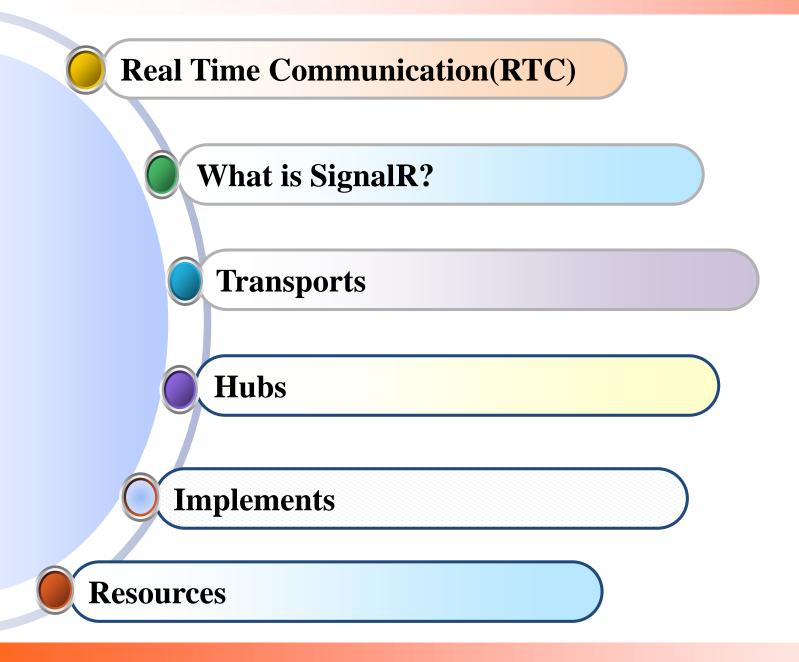
Realtime communication for Mobile App with ASP.NET Core SignalR













Real Time Communication

- *Real Time Communication or RTC, it has always been a hot topic, especially for **mobile**.
- ❖In normal mobile development when we wanted to get updates of data we would make a web request to a server, pull down data, and update our user interface. If you wanted to get bits of data on demand and in real time you would have to integrate WebSockets or some sort of push notifications to receive data as it is happening.



realtime

Admin SDK

do

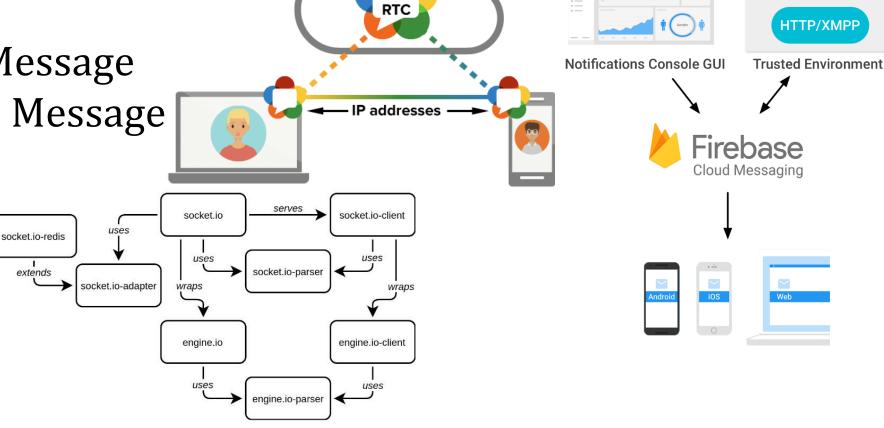
to

Real Time Communication

☐ There are many

communication:

- **❖**Web RTC
- Google Cloud Message
- Firebase Cloud Message
- ❖socket.io
- engine.io
- **❖**SignalR



libs/frameworks

Web



- ASP.NET Core SignalR is an open-source library that simplifies adding real-time web functionality to apps. Real-time web functionality enables server-side code to push content to clients instantly.
- SignalR provides an API for creating server-to-client remote procedure calls (RPC). The RPCs call JavaScript functions on clients from server-side .NET Core code.

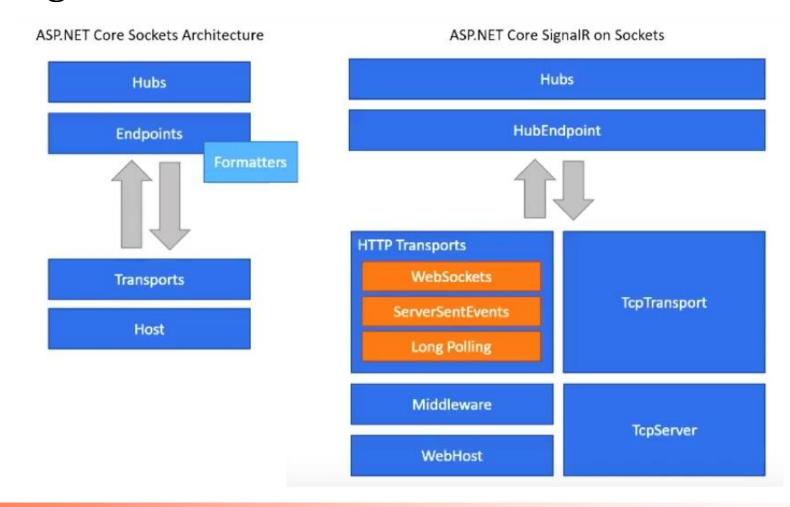


At the center of SignalR is a "Hub" that handles incoming and outgoing messages. A client sends a message to the Hub and the Hub distributes the message to all clients that are connected:

Send Message HUB



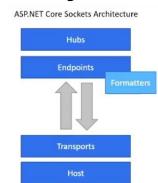
Features of SignalR for ASP.NET Core:

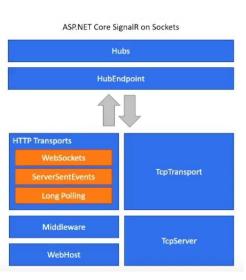




*****Features of SignalR for ASP.NET Core:

- ✓ Handles connection management automatically.
- ✓ Sends messages to all connected clients simultaneously. For example, a chat room.
- ✓ Sends messages to specific clients or groups of clients.
- ✓ Scales to handle increasing traffic.
- ✓ Supports WebSocketss, Server-Sent Events, & Long Polling and figures out which to use automatically.







❖Good candidates for SignalR:

- ✓ Apps that require high frequency updates from the server. Examples are gaming, social networks, voting, auction, maps, and GPS apps.
- ✓ Dashboards and monitoring apps. Examples include company dashboards, instant sales updates, or travel alerts.
- ✓ Collaborative apps. Whiteboard apps and team meeting software are examples of collaborative apps.
- ✓ Apps that require notifications. Social networks, email, chat, games, travel alerts, and many other apps use notifications.

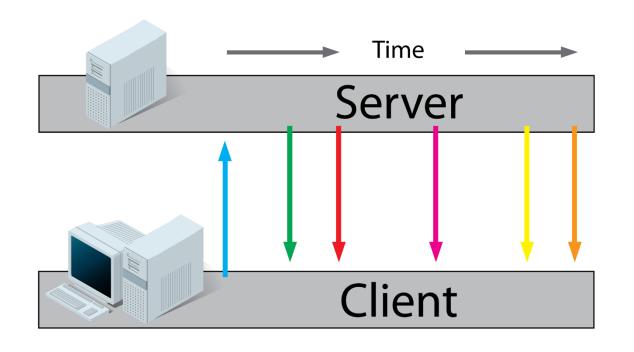


- SignalR supports several techniques for handling real-time communications:
- ✓ Server-Sent Events
- ✓ WebSockets
- ✓ Long Polling
- SignalR automatically chooses the best transport method that is within the capabilities of the server and client.



☐ Server-Sent Events

- 1. A client requests a webpage from a server using regular HTTP
- 2. The client receives the requested webpage and executes the JavaScript on the page which opens a connection to the server.
- 3. The server sends an event to the client when there's new information available.

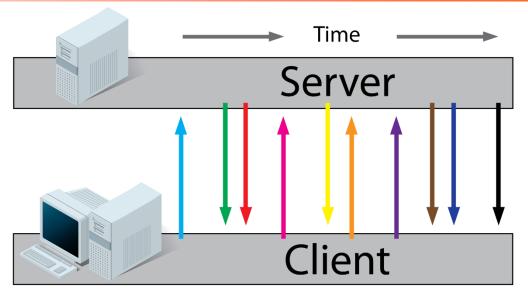


- •Real-time traffic from server to client, mostly that's what you'll need
- •You'll want to use a server that has an event loop
- •Connections with servers from other domains are only possible with correct CORS settings



☐ WebSockets

- 1.A client requests a webpage from a server using regular HTTP.
- 2. The client receives the requested webpage and executes the JavaScript on the page which opens a connection with the server.
- 3. The server and the client can now send each other messages when new data (on either side) is available.



- •Real-time traffic from the server to the client **and** from the client to the server
- •You'll want to use a server that has an event loop
- •With WebSockets it is possible to connect with a server from another domain.
- •It is also possible to use a third party hosted websocket server.

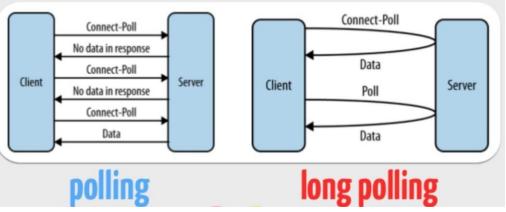


☐ Long Polling

- 1.A client requests a webpage from a server using regular HTTP.
- 2. The client receives the requested webpage and executes the JavaScript on the page which requests a file from the server.
- 3. The server does not immediately respond with the requested information but waits until there's **new** information available.
- 4. When there's new information available, the server responds with the new information.

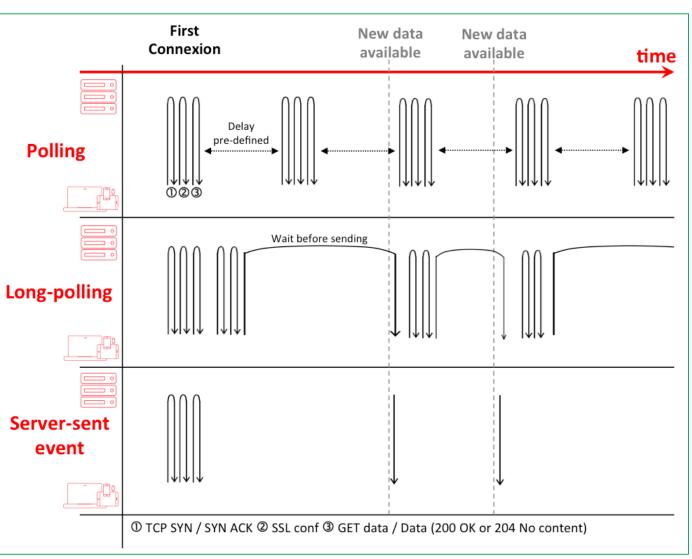
5. The client receives the new information and immediately sends another request to the

server, re-starting the process.





- Long/short polling (client pull)
- WebSockets (server push)
- Server-Sent Events (server push)
- Client pull—client asking server for updates at certain regular intervals
- Server push—server is proactively pushing updates to the client (reverse of client pull)

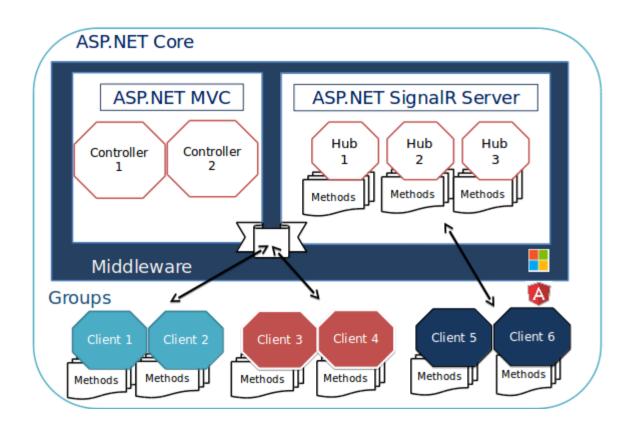




Hubs

SignalR uses hubs to communicate between clients and servers.

A hub is a high-level pipeline that allows a client and server to call methods on each other. SignalR handles the dispatching across machine boundaries automatically, allowing clients to call methods on the server and vice versa. You can pass strongly-typed parameters to methods, which enables model binding





Hubs

- SignalR provides two built-in hub protocols:
- ✓ a text protocol based on JSON
- ✓ a binary protocol based on MessagePack.

Hubs call client-side code by sending messages that contain the name and parameters of the client-side method. Objects sent as method parameters are deserialized using the configured protocol. The client tries to match the name to a method in the client-side code. When the client finds a match, it calls the method and passes to it the deserialized parameter data.



Implements

❖ Hub Backend: Using Visual Studio 2019 with ASP.NET Core app.

```
using Microsoft.AspNet.SignalR;
using Microsoft.AspNet.SignalR.Hubs;
using Microsoft.Owin.Cors;
using Microsoft.Owin.Hosting;
using Owin;
```

Client: Desktop, Web, Mobile

using Microsoft.AspNet.SignalR.Client;

DEMO



Resources

- https://github.com/signalr
- https://github.com/aspnet/SignalR
- https://github.com/aspnet/AspNetCore/tree/master/src/SignalR
- https://codeburst.io/polling-vs-sse-vs-websocket-how-to-choosethe-right-one-1859e4e13bd9
- https://github.com/aspnet/SignalRsamples/tree/master/AndroidJavaClient
- https://dotnet.microsoft.com/apps/aspnet/real-time
- https://docs.microsoft.com/enus/aspnet/core/signalr/introduction?view=aspnetcore-2.2



Resources

- https://github.com/shahedc/SignalRCoreSamples
- https://montemagno.com/real-time-communication-for-mobile-with-signalr/
- https://github.com/jamesmontemagno/MotzCodesLive/tree/master/Twitch-SignalRSaturdays



Contacts

Trần Duy Thanh

duythanhcse@gmail.com

0987773061

Group hỗ trợ học lập trình:

https://www.facebook.com/groups/communityuni/

Blog chia sẻ kiến thức lập trình:

https://techtalk.vn/blog/

https://duythanhcse.wordpress.com/



END



Hey! Coding is easy!

