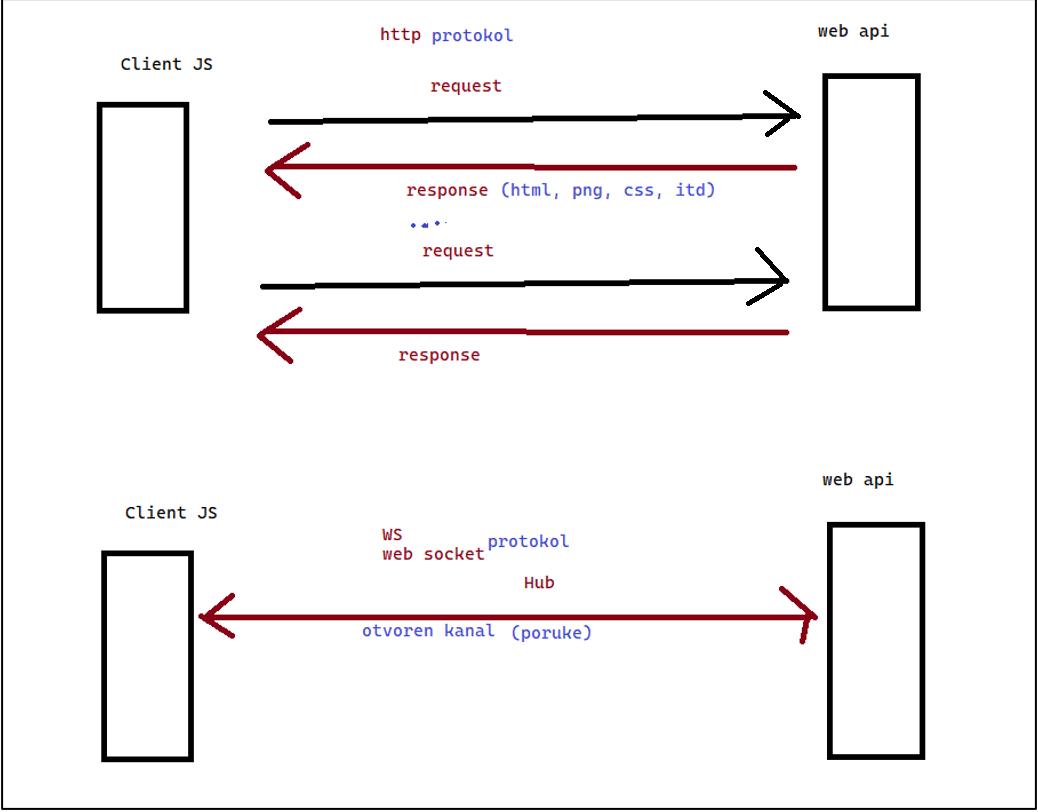
**SignalR**

<https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?WT.mc_id=dotnet-35129-website&view=aspnetcore-8.0>

<https://github.com/aspnet/SignalR-samples>



**Razlika između HTTP protokola i WebSocket (WS) protokola**

1. **HTTP protokol**:
   * Koristi model **request-response**.
   * Klijent šalje zahtjev (request), a server odgovara sa sadržajem (npr. HTML, PNG, CSS, itd.).
   * Komunikacija je uvijek inicirana od strane klijenta, **server ne može poslati poruku klijentu bez zahtjeva**.
   * Ovo je **stateless** komunikacija – svaki zahtjev je zasebna, nova veza.
2. **WebSocket (WS) protokol**:
   * WebSocket omogućava **dvosmjernu komunikaciju** između klijenta i servera.
   * Kanal ostaje otvoren nakon inicijalne uspostave veze.
   * **Server može aktivno slati poruke klijentu** bez da čeka novi zahtjev.
   * Efikasnije za aplikacije koje zahtijevaju **real-time** ažuriranja (npr. chat, notifikacije).

**Šta je SignalR?**

**SignalR** je biblioteka koja omogućava real-time komunikaciju u aplikacijama koristeći WebSocket protokol ili fallback opcije (npr. Server-Sent Events ili Long Polling). Koristi se za scenarije kao što su:

* **Notifikacije** u realnom vremenu.
* Ažuriranja korisničkog sučelja.
* Chat aplikacije.

**Primjer: Kreiranje notifikacija u Angular aplikaciji sa SignalR**

Evo primjera gdje se šalje notifikacija (sa zvučnim signalom i snackbar porukom) kada se **"city"** edituje:

**Backend (C# - ASP.NET Core Web API)**

1. Instalirajte SignalR:

dotnet add package Microsoft.AspNetCore.SignalR

1. Kreirajte **Hub**:

using Microsoft.AspNetCore.SignalR;

public class NotificationHub : Hub

{

public async Task NotifyCityUpdated(string cityName)

{

await Clients.All.SendAsync("ReceiveNotification", $"City {cityName} has been updated!");

}

}

1. Dodajte Hub u Program.cs:

var builder = WebApplication.CreateBuilder(args);

var app = builder.Build();

app.MapHub<NotificationHub>("/notificationHub");

app.Run();

1. Pozovite SignalR Hub kada se **city** edituje (u kontroleru):

private readonly IHubContext<NotificationHub> \_hubContext;

public CityController(IHubContext<NotificationHub> hubContext)

{

\_hubContext = hubContext;

}

[HttpPut("{id}")]

public async Task<IActionResult> UpdateCity(int id, CityDto city)

{

// Logika za ažuriranje grada

await \_hubContext.Clients.All.SendAsync("ReceiveNotification", $"City {city.Name} has been updated!");

return Ok();

}