

# Web Programlama I

---

## Ders 11 - Web Soketler

28.12.2021

Erciyes Üniversitesi  
Bilgisayar Mühendisliği Bölümü

**Eğitmen:** Ömür ŞAHİN

- WebSocket ile hangi problemin çözüldüğünü anlamak
- React/NodeJS uygulamasına nasıl WebSocket eklendiğini öğrenmek

# 1-WebSocket

---

- Bir chat uygulamasını tarayıcı üzerinde nasıl gerçekleştirebilirsiniz?
- Çok da kolay değil :)

# Seenek 1: Server-side

- Mevcut sayfa GET kullanılarak HTML ile oluřtur.
- POST mesajı sonucu gncellenmiř HTML sayfası dnen yeni bir mesaj oluřtur.
- Durum 1: Tek bir mesaj yazılmıř olsa bile btn mesajlar tekrar indirilir.
- Durum 2: Mevcut kullanıcı diğerkullanıcıların yazmıř olduėu mesajı sayfayı yenileyene kadar veya yeni mesaj yazana kadar gremez.

## Seenek 2: AJAX

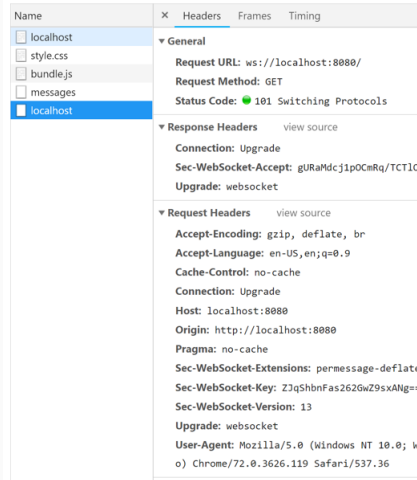
- AJAX kullanarak yalnızca listelenecek yeni mesajları ek
- X milisaniye ierisinde aynı eylemi tekrarla
- Problem 1: Yeni mesajı grmeden nce X ms beklemek zorundasınız.
- Problem 2: Yeni mesaj olmasa da istekte bulunacak ve bo yere trafik oluturacaktır.
- X seimi Problem 1 ve 2 arasında bir tradeoff oluturur.
  - Kk X seimi kullanılabilirliėi artırırken maliyeti de artıracaktır.

## Seenek 3: Web Soketler

- HTTP deęil WS protokolüyle gerekleşir.
  - Pek ok tarayıcı WS'i destekler.
- WS ift taraflı iletişimi mümkün kılar.
  - Sunucu güncellemeleri dinleyen tarayıcıları veri göndermeye karar verebilir.
- Sunucu her bir istemci için aktif bir TCP bağlantısı tutar.
- Yeni mesaj geldięinde sunucu bütün istemcilere broadcast yapar.
- Tarayıcılar yalnızca bildirimleri bekler ve yeni bir mesaj geldięinde HTML üzerinde güncelleme yapar.
- Sunucular yalnızca yeni veri erişilebilir olduęunda gönderir. Böylece ekstra trafik oluşturmaz.

# WebSoket Protokolü

- Genellikle TCP üzerinde çalışır.
- HTTP değildir ancak ilk mesajın syntax'ı HTTP ile aynıdır.
- NOT: URL'de ws protokolü kullanılır. bir protokoldür.
  - Örnek: ws://localhost:8080
  - wss HTTPS gibi şifrelenmiş versiyonudur.



The screenshot shows the 'Headers' tab in a web browser's developer tools. The 'Name' column on the left lists several resources: 'localhost', 'style.css', 'bundle.js', 'messages', and 'localhost'. The 'localhost' entry at the bottom is selected. The right pane shows the details for this selected resource, which is a WebSocket connection. The 'General' section displays the 'Request URL' as 'ws://localhost:8080/', the 'Request Method' as 'GET', and the 'Status Code' as '101 Switching Protocols'. The 'Response Headers' section shows 'Connection: Upgrade', 'Sec-WebSocket-Accept: gURaMdcj1p0CmRq/TCT10', and 'Upgrade: websocket'. The 'Request Headers' section shows various headers including 'Accept-Encoding: gzip, deflate, br', 'Accept-Language: en-US,en;q=0.9', 'Cache-Control: no-cache', 'Connection: Upgrade', 'Host: localhost:8080', 'Origin: http://localhost:8080', 'Pragma: no-cache', 'Sec-WebSocket-Extensions: permessage-deflate', 'Sec-WebSocket-Key: ZJqShbnFas262GwZ9sxANG=', 'Sec-WebSocket-Version: 13', 'Upgrade: websocket', and 'User-Agent: Mozilla/5.0 (Windows NT 10.0; w o) Chrome/72.0.3626.119 Safari/537.36'.



- WS protokolüne istek yapılacağı zaman tarayıcı aşağıdaki Header bilgileri ile HTTP ile aynı syntax'a ait bir mesaj oluşturur.
  - **Sec-WebSocket-Extensions**
    - WS eklentileri ile ilgili özellikler belirtilir (mesajlar nasıl sıkıştırılacak vs)
  - **Sec-WebSocket-Key**
    - Sunucuya bunun bir WS bağlantısı olduğunu, HTTP bağlantısı olmadığını belirtmek için kullanılır.
    - Rastgele bir anahtar kullanılır.
  - **Sec-WebSocket-Version**
    - Tarayıcıda yürütülen WS protokolünün versiyonu
  - **Upgrade: websocket**
    - Standart HTTP header'ıdır. İstemcinin farklı bir protokole (burada WS) geçmek istediğini belirtir.

- Eğer sunucu WS bağlantılarını destekliyor ve kabul ediyorsa aşağıdaki içerikleri barındıran bir HTTP mesajı ile cevap verecektir.
- **Connection: Upgrade**
  - Mevcut HTTP bağlantısını bir başka şeyle güncelleyeceğini tarayıcıya söyleyecektir.
- **Upgrade: websocket**
  - Sıradaki isteğin kullanacağı protokol.
- **Sec-WebSocket-Accept**
  - Sonraki bütün istekler için WS protokolünü kullanacağını doğrulamak için kullanılır.
  - Tarayıcıdan gönderilen bir hash anahtar barındırır.
- **HTTP durum kodu: 101**
  - Protokolün değiştirilmesini temsil eder.

- WS bağlantısı kurulduğunda byte data veya string TCP üzerinden gönderilebilir.
- Gelen mesajlar beklenebilir.
  - Tarayıcı ve sunucu arasında çift yönlü bir iletişim vardır.
- Mesajın yapısını geliştirici belirlemektedir.
  - STOMP gibi bir protokol kullanılabilir.
- Bu ders kapsamında yalnızca JSON objeleri göndereceğiz.

# Neden ilk mesaj HTTP ile yapılıyor?

- Sunucuya yalnızca tek bir TCP socketinin dinlenmesine izin verilir.
  - 80 veya 443
- Ters proxy'ler de dahil olmak üzere mevcut web altyapılarına entegre edilmesi kolaydır, genellikle doğrudan bir sunucuyla değil, proxy'ler ve gatewaylerle iletişime geçilir.
- WS HTTP'ye göre daha sonra ortaya çıkmıştır.
  - İlk versiyonu 2009 yılında Chrome çıkarmıştır.
- WS protokolünü kolay bir şekilde var olan mevcut web altyapısına entegre etmek için HTTP ile kullanılmaktadır.

- JavaScript'te WebSocket'leri bütün tarayıcılarda kullanabilirsiniz.
- **WebSocket(url)**
  - WS objesi oluşturur ve verilen url'e bağlanmaya çalışır. Burada url ws veya wss protokolündedir.
- **WebSocket.send(payload)**
  - Verilen payload'ı servera gönderir.
- **WebSocket.onmessage**
  - Sunucudan gelen mesajı kontrol etmeye yarayan callback fonksiyon
- **WebSocket.close()**
  - Bağlantı kapatmak için kullanılır.

- Backend tarafında WS desteği programlama diline ve kütüphanelere bağlıdır.
- Bu ders kapsamında Express server'ın "ws" ve "express-ws" kütüphanelerini kullanacağız.
- Express tarafında "ws://" ile başlayan endpointleri ele alacağız.