

Web Programlama I

Ders 13 - WebSocket, Deploy, Veritabanları...

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

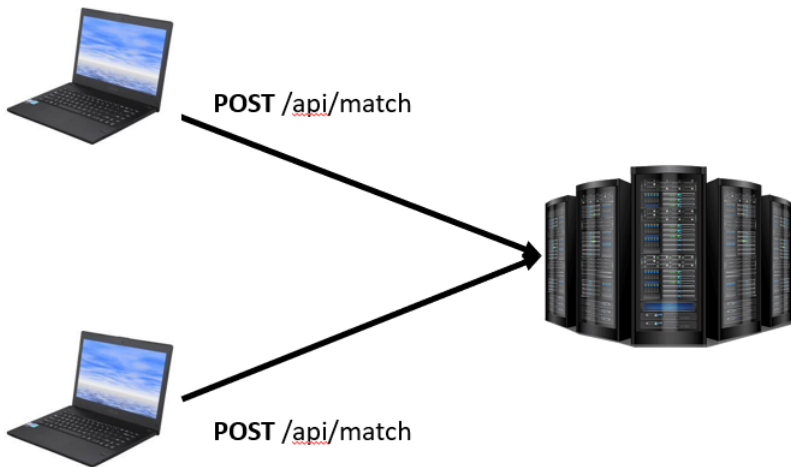
Eğitmen: Ömür ŞAHİN

- Online oyunlara bir yaklaşım
- Uygulamanın deploy edilmesi
- Veritabanları üzerine tartışma

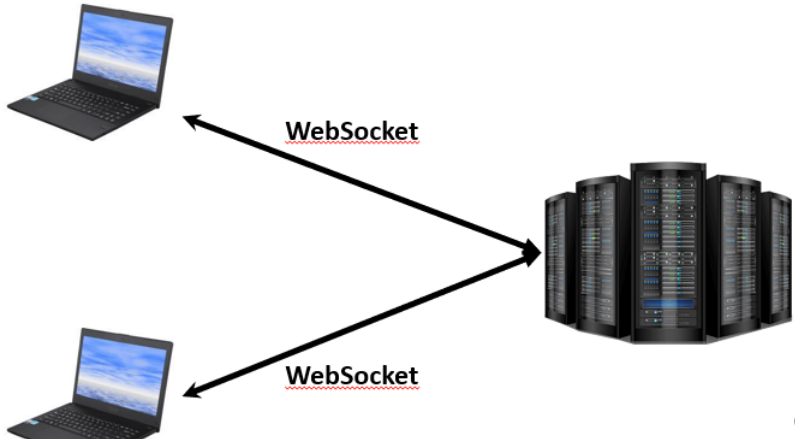
1-WebSocket-Ekstra

- Sıra tabanlı bir oyun tasarlanacağını varsayalım.
- Farklı makinalardaki oyuncular nasıl eşleştirilir?
- Hile nasıl engellenir?
- vs.

- Aynı endpointe istek atılarak (IP:port) oyun başlatma isteği gönderilir.
- En az 2 oyuncu olana kadar bu istekler kuyruğa atılır.



- Eğer bir oyun başlatılırsa her bir kullanıcı arasında sunucu aracılığı ile WebSocket bağlantısı kurulur.
- Hilenin engellenmesi için maçın mevcut durumu sunucuda tutulur ve kullanıcılar sadece bunu görüntüler.
- Her bir durum değişiminde oyuncular WebSocket aracılığı ile bilgilendirilirler.



- A ve B kullanıcıları arasında X ID'li bir oyun oluşturulduğunu varsayalım.
- Sunucu X oyununa A kullanıcılarından gelen talebin gerçekten A kullanıcılarından gelip gelmediğini nasıl bilebilir?
- En az iki seçenek:
 - İlk mesaj HTTP olduğu için aynı yetkilendirme kullanılabilir.
 - Tek kullanımlık tekil bir rastgele token oluşturulur ve yetkilendirme bu token ile sağlanır.
 - Bu daha genel bir yaklaşımdır.
 - Hatırlatma: WS'te yalnızca ilk bağlantı anında yetkilendirmeye ihtiyaç olur sonraki isteklerde HTTP'de olduğu gibi bir token gönderilmez.

- TCP soketler işletim sistemleri kaynaklarını tükettiği için WS oldukça maliyetlidir.
- Her bir WS bağlantısı için bir TCP bağlantısının açık kalması gerekmektedir.
- HTTP'de gelen istek yerine getirildiğinde bağlantı kapatılabilir.
 - TCP bağlantısı 4 temel bilgiye ihtiyaç duyar: İstemci IP'si, İstemci Portu, Sunucu IP'si ve sunucu portu
 - HTTP'de TCP bağlantısının açık tutulması yalnızca performans sebebiyledir.
 - Connection: Keep-Alive HTTP header'ı ile sağlanır.

- İstemci ve sunucu arasında WS kullanılarak farklı tipte iletişim gerçekleştirilebilir.
- Her işlem için ayrı TCP soket bağlantısı gerçekleştirmek oldukça maliyetlidir.
- Bu yüzden aynı WS bağlantısını farklı tipte işlemler için kullanmak gerekmektedir.
- Basit çözüm: JSON verisine "topic" gibi bir alan daha ekleyerek yapılacak işlem belirtilebilir.
 - Veya STOMP gibi mesajlaşma protokolleri kullanılabilir.
- Sunucu da topic alanındaki değere göre ne yapacağına karar verebilir.

```
{  
  "topic": "..."  
  "data": {...}  
}
```

```
{  
  "topic": "update"  
  "data": {  
    "matchId": 42,  
    "boardDto": {...},  
    "isX": true,  
    "opponentId": 1234  
  }  
}
```

2-Deployment

- Farklı firmalar belli bir ücret karşılığında cloud hizmeti sunmaktadır.
- Amazon Web Services (aws) içlerinde en ünlülerinden biridir.
 - Netflix AWS üzerinde çalışmaktadır.
- Otomatik ölçekleme: Eğer daha fazla yük oluşursa, daha fazla node kiralanabilir ve böylelikle yük ölçeklenebilir.



There is no cloud
it's just someone else's computer

- Önemli cloud sağlayıcılarından biridir.
- Şu anda kullanımı kolay ve ücretsizdir.
 - Ücretsiz olması ilerleyen zaman diliminde değişebilir.
- NodeJS uygulamalarını destekler
 - Ve pek çok türde uygulamayı da desteklemektedir.

- İlk olarak www.heroku.com adresinden bir hesap oluşturulmalıdır.
- Heroku komutlarını komut satırı aracılığı ile çalıştırmanızı sağlayan CLI'ı kurun.
- Web arayüzünden uygulamanın adını girerek yeni bir uygulama oluşturun. Örnek: bs436-wp
 - Bu isim tekil olmalıdır.

- **heroku plugins:install heroku-builds**
 - builds plugin'ini kurmak için yalnızca 1 sefer çalıştırılır.
- **heroku login**
 - Kimliklendirme gerektiren komutları çalıştırmak için giriş yapmayı sağlar.
 - Not: Eğer windows kullanıyorsanız GitBash ile çalışmayabilir, dahili terminali kullanarak çalıştırabilirsiniz.
- **heroku builds:create -a bs436-wp**
 - Mevcut klasördeki bütün dosyaları zip'ler ve deploy eder.
 - Not: ".gitignore" kullanarak hariç bırakılacak dosyalar belirlenebilir.

- "package.json" içerisindeki engines ile ekstra ayarlamalar yapılabilir ancak zorunlu değildir.
 - örnek: Node ve Yarn versiyonları belirtilebilir.
- **process.env.PORT** ile hangi portun kullanılacağı belirtilmelidir.
 - Aksi takdirde Heroku reverse-proxy'yi nasıl kullanacağını bilemez.
- Uygulama heroku tarafından "yarn build" ve "yarn start" komutlarını kullanarak otomatik olarak build edecektir.

Quiz

Giriş yapmadınız

[Giriş](#)

Quiz Oyunu

Quiz oyununa hoş geldiniz. Bu oyunda, 4 şıktan oluşan bir dizi soru soracağız. Doğrudur. Eğer sorulardan birini yanlış cevaplarsan, yenilirsin! Oyunu kazanmak için doğru cevaplamak zorundasın :)

Online oyuncu sayısı: 1

Oynamak için giriş yapmanız gerekmektedir!

- WebSocket oldukça fazla sistem kaynağına ihtiyaç duymaktadır bu yüzden maliyetlidir.
- Çoğu ücretsiz cloud hizmet sağlayıcıları düşük sayıda WS bağlantısına izin vermektedir.
- Kullanıcının kısa süreli inaktif olması durumunda (30-60 saniye gibi) bağlantıyı sonlandırmaktadır.
 - Zorla kapatılması durumunda otomatik olarak tekrar bağlanma ile problem çözülebilir.

3-Veritabanları

Neden veritabanı?

- Veriyi bir yerde saklamak için kullanılır.
- Eğer memory'de tutulursa uygulama tekrar başlatıldığında her şey kaybolur.
- Tamamlanmış bir uygulamada çoğu zaman bir veritabanına ihtiyaç duyulur.
 - Heroku gibi cloud sağlayıcılar veritabanı hizmeti de sunar.

Veritabanı seçiminde 3 temel kural

- Yeni bir projeye başlıyorsanız veya ne yapacağınıza karar veremiyorsanız Postgres seçin.
- Eğer MySQL kullanıyorsanız ve Postgres'e geçiş çok maliyetliyse MySQL ile devam edin.
- Veritabanları konusunda uzun bir deneyiminiz varsa, tam olarak ne yaptığınızı biliyorsanız ve bir veritabanının Postgres'e karşı üstünlüğünü, performansını nesnel olarak ölçebiliyorsanız, o zaman, seçtiğiniz veritabanını kullanın.

- Açık kaynak ancak Oracle tarafından geliştirilmektedir. Bu firmanın ticari ürünü olan Oracle veritabanı olduğunu unutmayalım.
 - Teoride her iki veritabanı da birbiriyle rekabet halindedir.
- Pek çok durumda MySQL, Postgres'e denktir ancak çoğu zaman yeni özellikler biraz daha geç eklenir.
 - Örnek: JSON veri tipinde NoSQL özellikleri gibi.

- En ünlü NoSQL veritabanıdır.
 - Özellikle NodeJS olmak üzere pek çok tutorial'da oldukça popülerdir.
- Tek ilişki hiyerarşik düzenidir ve genellikle JSON formatında tutulur.
- Oldukça hızlıdır ve ayağa kaldırması da çok basittir.
- Postgres/MySQL oldukça hızlı bir şekilde JSON veritipini kaydedebilmektedir.
 - Örnek: 2014 yılında Postgres benchmark testlerinde MongoDB'den daha hızlı JSON veri işlemlerini çözümlmekteydi.

- Bir gün veriler arasında ilişki kurmak zorunda kalırsanız oldukça büyük problemlere yol açacaktır.
- Daha kötü senaryo: İlişkisel verilere ihtiyaç duymanız durumunda bile MongoDB'yi sadece yaratmış olduğu hype sebebiyle seçmek.
- Veya Google/Amazon vs. firmaların ele almak zorunda olduğu kadar büyüklükte verilerle uğraşmayacağınız durumlarda MongoDB'yi seçmek...