

Web Programlama I

Ders 05 - SPA Route

09.11.2021

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

Eğitmen: Ömür ŞAHİN

- SPA sayfaları arasında geçiş yönlendirmelerini anlamak
- Client-side ve server-side rendering arasındaki ve her ikisine de neden ihtiyaç duyduğumuzu anlamak

Tek Sayfa Uygulamaları (Single Page Application, SPA)

- Yalnızca tek bir HTML sayfasından oluşmaktadır.
- Bütün değişiklikler JS ile DOM manipülasyonu sağlanarak gerçekleştirilir.
- Bir sayfadan bir başka sayfaya geçiş DOM'un JS ile değiştirilmesi anlamına gelmektedir.
- Bu değişim bir ilüzyondur. Aslında yeni bir sayfa çağırılmamaktadır.
 - Backend tarafında bir talepte bulunulmayacaksa muhtemelen yeni bir HTTP isteği atılmaz.
- Tarayıcıdaki adres satırı da herhangi bir HTTP isteği atmadan değiştirilebilir.

- React içerisinde SPA desteklemek için kullanılmaktadır.
 - Bu kütüphane Facebook tarafından yapılmamıştır. React yalnızca HTML render etmek için kullanılır.
- Hangi component'ın gösterileceği adres satırındaki URL'e bağlıdır.

- Tek bir index.html içerisinde 3 sayfa bulunmaktadır.
- `"/foo"` çağırıldığında JS ilk div'i görüntüler ve diğerlerini gizler.
 - **localhost:8080/foo**
- URL adresi değişir ve index.html tekrar render edilir.

```
<div>  
<div> /foo sayfası < /div>  
<div> /bar sayfası < /div>  
<div> / sayfası < /div>  
< /div>
```

- Tarayıcıda **localhost:8080** açılır.
- Tarayıcı index.html ve diğer statik dosyaları (bundle.js, css dosyaları vs.) indirir.
- Varsayılan yol `"/`
 - **localhost:8080**'nin eş değeri **http://localhost:8080/**'dir.
- **/foo** linkine tıklandığında adres çubuğu HTTP isteği yapmadan **http://localhost:8080/foo** olarak değişir.
- React-router adres çubuğundaki değişikliği algılayarak sayfayı tekrar render eder.

- Eğer kullanıcı adres satırından doğrudan **localhost:8080/foo** adresini çağırırsa ne olur?
- Veya tarayıcıyı yenilerse ne olur?
- Böyle bir durumda tarayıcı TCP üzerinden localhost:8080 ile bağlantı kurar ve **/foo** kaynağına erişmeye çalışır.
- Eğer **/foo** kaynağı bulunmazsa 404 kodu döner.
 - Çünkü bu sayfa JS tarafından dinamik olarak üretilmekteydi.
- Bu durumun iki olası çözümü bulunmaktadır.

- Sunucuya asla 404 hatası vermemesi gerektiği belirtilir.
- Bulunmayan talep geldiğinde index.html içeriği gönderilir.
- React-router da URL'e bağlı olarak sayfayı render eder.
- Eğer URL bilinen bir yol belirtmiyorsa JS bir hata sayfası render eder.
- Bu çözüm en iyi çözüm değildir ancak uygulaması oldukça kolaydır. Derste bu çözümü kullanacağız.

Çözüm 2: Server-Side Rendering

- Sunucu HTTP isteği aldığı anda dinamik sayfalar server tarafından oluşturulur.
 - Yani React server üzerinde çalışır ve HTML sayfası HTTP isteğinin payload'ında döndürülür.
- Sunucunun JS kodu çalıştırabilir olması gerekmektedir.
 - NodeJS gibi.
- Dikkate alınması gereken çok fazla nokta bulunmaktadır. Bu yüzden uygulaması zordur.
 - Frontend'in AJAX/WebSocket gibi bağımlılıklara sahip olduğu durumlar veya Redux gibi veri kaynaklarının başlatılması...

- JS olmadan sayfalar arası geçiş sağlanabilir.
 - Örneğin react-router <a> tagının onclick eventini geçersiz kılmıştır ve JS olmadan a çağrıları çalışmayacaktır. SSR kullanıldığında a tagının varsayılan davranışı kullanılabilir.
- Peki neden önemli?
- **Search Engine Optimization (SEO)**
 - Web sayfanızı Google, Bing gibi arama motorlarında görmek istediğinizde
 - Sayfa taramalarını crawler adı verilen programcılar yapmaktadır ve pek çoğu JS çalıştırmamaktadır. Çalışanlarda bile pek çok işlev gerçekleşmemektedir.
 - Crawler'lar bütün uygulamayı boş bir index.html sayfası görebilirler.
 - React için NextJS, Vue için NuxtJS kütüphaneleri popülerdir.