

Web Programlama I

Ders 04 - SPA State Kontrolü

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

Eğitmen: Ömür ŞAHİN

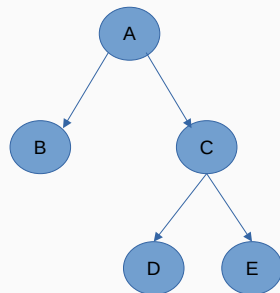
1-Genel Bakış

- Çoklu componentte state kontrolü nasıl gerçekleştirilir?
- React component için birim test nasıl yazılır?

2-State Kontrolü

Parent-Child Ağaç Hiyerarşisi

- Her düğüm child düğüme erişim sağlayabilmektedir.
- Düğümler parent düğüme doğrudan erişim sağlayamaz.
 - Component'lar pek çok farklı noktada tekrar tekrar kullanılabilir.
- Component'ler parent veya sibling componentleri değiştirmeye ihtiyaç duyabilir.
 - C componenti B ile iletişime geçmek isteyebilir ancak C yalnızca D ve E'ye ait özellikleri görebilmektedir.
- Bu durumda ne yapmalı?
 - Bi' şey yapmalı!

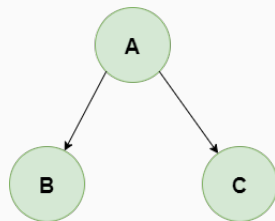


- Parent component tarafından bir X child componenti oluşturulduğunda HTML özelliği gibi bazı özellikler (properties) props aracılığı ile child componente aktarılır.
 - Örnek: `< A test=5 / >`
- props'u değiştirilemez (immutable) bir durum olarak ele almak gerekmektedir.
 - Props okunabilir ancak child component içinde değiştirmemelisiniz.
 - `render()` metodu içinde okuyup kullanabilirsiniz.
 - props'un yanlışıyla değiştirilmesi `render()` metodunun çağırılmasını tetiklemez.

- prop içeriği fonksiyondan oluşabilmektedir.
- Bu fonksiyon parent düğüme aittir.
 - Parent'a ait setState() fonksiyonunu çağırabilir.
 - Diğer kardeş düğümlere (sibling) erişebilir.
- Child düğüm parent ile doğrudan iletişime geçemediği durumlarda parent düğüm prop içerisinde bir callback fonksiyon gönderebilir.

State Lift Up

- Eğer iki component aynı state'e bağlı ise bu state ana componentten çocuk componentlere callback fonksiyonlar aracılığı ile gönderilir.
- Ana component bir callback fonksiyon sağlayarak bu state'in manipüle edilmesini sağlayabilir.
- Recall: Bir component'a ait state güncellendiğinde çocuk componentların da tekrar render edilmesine verilen ad.
- Örneğin: B componenti `x()` fonksiyonunu çağırdığında, `A.callback()`'i de çağıracaktır bu da A'daki state'i değiştirir ise A ve B ile C' de tekrar render edilecektir.



`<A>`

`<B x=callback / >`

`<C x=callback / >`

`< /A>`

- State lift up yöntemi pek çok uygulamada işe yaramaktadır.
 - Ancak componentlar arasında karmaşık ilişkilerin bulunduğu ve birden fazla katmanda state'in taşınması gerektiği durumlarda bu yapı yetersiz kalmaktadır.
- **React Context:** Component ağaçları arasında veri geçişini her seviyede prop olarak göndermeden sağlayan yapıdır.
- **Redux:** State yönetiminde oldukça popüler bir yaklaşımdır.
 - State'ler ortak bir kaynakta tutulur ve bu kaynaktan erişim sağlanır.
 - Component'in her bir değişimde tekrar render edilebilmesi için redux'un veri kaynağına bağlanması gerekmektedir.
 - Redux kullanımına bu ders kapsamında ihtiyacımız bulunmamaktadır ancak ne olduğu hakkında fikir sahibi olmak zorundasın.

- React component'ın HTML üreten render() metodu bulunmaktadır.
- HTML taglarının state değiştiren olayları (eventleri) bulunabilmektedir.
- Bu olayların tetiklenmesi ile HTML üzerinde değişiklikler gerçekleşebilir.
- Bu tür davranışlar nasıl test edilebilir?

- Birim test için mevcut tarayıcıyı kullanmak gereksiz ve ekstra yük getirmektedir.
- NodeJS içinde fake ve headless bir tarayıcının çalıştırılması gerekmektedir.
- Bu da Enzyme, JSDOM veya JEST gibi kütüphanelerle mümkün olmaktadır.
- Temel bir HTML sayfası içine test etmek istediğimiz bileşeni entegre edeceğiz.
- Tıklamalar ve diğer olaylar simüle ederek HTML string'indeki değişiklikler incelenecektir.

- Enzyme kullanarak web sayfası üzerinde işlem yapılacağı zaman hangi HTML elementini kullanacağımızı belirtmemiz gerekmektedir.
 - Örnek: hangi butona tıklanacak, hangi alan doldurulacak vs.
- Elementlerin seçimi için 2 temel dil bulunmaktadır: CSS Selector ve XPath
- CSS Selector .css dosyasındaki yazıma benzerdir.
 - ".foo": class özelliği "foo" olan tüm elemanları seç.
 - "#foo": id değeri "foo" olan elemanı seç.
 - vs.
- Jest kullanarak eleman seçimi ise data-testid özelliği verilerek gerçekleştirilebilmektedir.