

Web Programlama I

Ders 04 - SPA State Kontrolü

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

Eğitmen: Ömür ŞAHİN

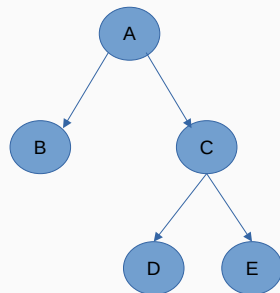
1-Genel Bakış

- Çoklu componentte state kontrolü nasıl gerçekleştirilir?
- React 19'daki yeni state yönetim özellikleri neler?
- Modern kütüphaneler nelerdir?

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!

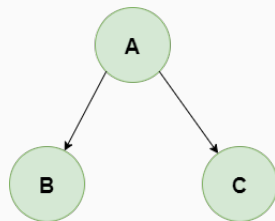


- Parent component tarafından bir X child componenti oluşturulduğunda HTML özelliği gibi bazı özellikler (properties) props aracılığı ile child componente aktarılır.
 - Örnek: `< A test=5 / >`
- props'u değiştirilemez (immutable) bir durum olarak ele almak gerekmektedir.
 - Props okunabilir ancak child 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 yaklaşımdı.
 - Hala popüler olsa da payı azalıyor.
 - Standart yol Redux Toolkit (RTK)
 - Küçük/orta projelerde Zustand tercih ediliyor.

Zustand örneği

```
1 // çok basit kullanım
2 import { create } from 'zustand'
3
4 const useStore = create((set) => ({
5   count: 0,
6   increment: () => set((state) => ({ count:
7     state.count + 1 })),
8 })
9
10 // Component'te kullanım
11 function Counter() {
12   const { count, increment } = useStore()
13   return <button
14     onClick={increment}>{count}</button>
15 }
```