

React;

Theme, yani tema kullanımında

Language, yani dil değişimlerinde

Login, authentication username işlemlerinde CONTEXT kullanılmasını öneriyor.

Prop drilling yapmanın zor olduğu yerlerde contex API kullanmak mantıklı, heşeyi globala bağlamamak lazım, 3 tane useState ile 1 tane useContex kullanılabilir.

Redux🡪 Store, Context-> context ortak saklama alanı ismi,

Özetle, context ve redux aynı değil, benzer şeyler,

Basit projelerde context kullanmak mantıklı ve yeterli, (statik, flood mantığı olan yerlerde)

(mesela dil, bir kere değişir sonra artık ellenmez gibi)

Daha karmaşık/dinamik projelerde (sepete ekle/çıkar gibi) redux kullanmak daha faydalı

1. Önce contexi oluştur. (CREATE)

const ThemeContext = React.createContext('light');

2.Kullanılacak yeri .provider ile sarmala (PROVİDER)

<ThemeContext.Provider value="dark">

<Toolbar />

</ThemeContext.Provider>

3 . .ihtiyaç olan yerde kullanma (CONSUME) useContext hook kullanarak,

contextType = ThemeContext;

render() {

return <Button theme={this.context} />; }

}

Props mantığında; Dede torunun torununa harçlık verecek ama bunu doğrudan ulaştıramıyor.önce oğluna veriyor o onun oğluna uzayıp gidiyor.

Contextte ise şöyle düşünebiliriz; Dede bir banka hesabı açıyor childlarına torunlarına hesabı kullanma yetkisi veriyor torun artık doğrudan parayı banka hesabından alabiliyor veya banka hesabına para aktarabiliyor



