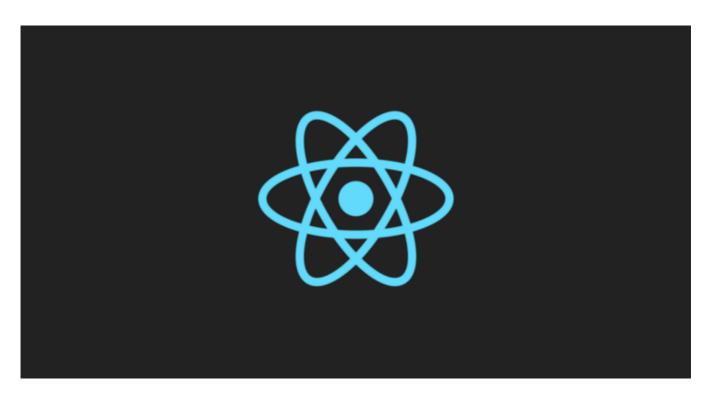


React Hooks'a Genel Bakış (useState ve useEffect Kullanımı)



ReactJS.org üzerinden yapılan tanıtımda, Hooks'un React'in alpha versiyonu olduğu, önceki sürümlerle herhangi bir çakışmasının olmadığı ve bu versiyonu kullanmak zorunda olmadığımız bilgisi veriliyor. Bu açıklamadan, React'in alpha versiyonunu büyük projelerimizde şu an için kullanmamamız gerektiğini ama ilerisi için deneyim sahibi olmamız gerektiğini anlıyoruz.

(Bu makale temel düzeyde React bilgisi gerektirir ve bilgi amaçlıdır. Asıl kaynağa https://reactjs.org/docs/hooks-intro.html adresinden ulaşabilirsiniz.)









React, Hooks'u "basit" ve "ek" olmak üzere iki bölüme ayırmış.

Basit kısım "useState", "useEffect" ve "useContext" bölümlerini içeriyor, ek ise "useReducer", "useCallback", "useMemo", "useRef", "useImperativeHandle", "useLayoutEffect", "useDebugValue" bölümlerini içeriyor.

"useState" ve "useEffect"i aşağıda başlıklar halinde elimden geldiğince açıklamaya çalışacağım.

useState

useState bize state ve setState'i bir arada sunan kullanışlı bir çözüm olmuş. Standart state kullanımımız, bileşenlerimizin (Component) içine önce state tanımını yapmamızı, daha sonra değiştireceğimiz fonksiyonlarda setState yapmamızı gerektiriyordu. En önemli kısım da stateless fonksiyonlarda (adı üstünde) kullanamıyorduk. Artık class tanımlamadan, fonksiyon içerisinde kullanabiliyoruz. Olumsuz yönü ise state içerisinde tuttuğumuz object'i birleştiremiyor (Object.assign) olmamız. Yani aşağıdaki setState işlemini useState ile kullanamıyoruz. React bunun yerine useReducer kullanmamızı öneriyor. Bence sadece component içerisinde kullanılacak bir state'i reducer üzerinde tutmamız çok mantıklı bir seçenek gibi görünmüyor.

```
1  setState(prevState => {
2   return {...prevState, ...updatedValues};
3  });

Object.assign hosted with ♥ by GitHub  view raw
```

useState kullanımı ise aşağıdadır:









stateDegiskeni kısmına degişkenimize vereceğimiz adı yazıyoruz. stateFonksiyonu kısmına ise state'imizi güncelleyeceğimiz fonksiyonun adını yazıyoruz. ilkDeger kısmına ise component'imiz yüklendiğinde state değişkenimizin alacağı değeri yazıyoruz. ilkDeger kısmına fonksiyon tanımlamamız mümkün fakat bu fonksiyonumuz sadece component ilk yüklendiğinde çalışacaktır. Sonraki render işlemlerinde aktif olmayacaktır. React dokümantasyonunda burada çalıştıracağımız fonksiyonu basit tutmamız gerektiği uyarısında bulunuluyor. Aksi durumda component'in oluşturulması uzun süreceği için tercih edilir bir durum olmayacaktır.

State'i değiştirmek için *stateFonksiyonu(yeniDeger)*; şeklinde yazmamız yeterli olacaktır.

State'te yer alan mevcut değer ile işlem yapmak için ise stateFonksiyonu(mevcutDeger => mevcutDeger + 1); şeklinde tanımlamamız yeterli olacaktır.









şeklinde yazmamız yeterli olacaktır.

useEffect

Bu özellik ise componentDidMount, componentDidUpdate ve componentWillUnmount gibi çalışmaktadır. Aslında her render işlemi sonrası çalışabildiği gibi bizim belirttiğimiz değişkenlerin değişmesi sonrasında da çalışabilir ya da sadece ilk render işlemi sonrası çalışabilir.

useEffect, useState gibi stateless fonksiyonlarda çalışabilmektedir.

Kulanımı aşağıdadır. Bu kullanım component her render olduğunda çalışacaktır.









Sadece ilk yüklenme işlemi sonrasında çalışması için ise aşağıdaki değişikliği yapmamız gerekmektedir.

Sadece seçtiğimiz bir değişkenin değeri değiştiğinde çalışması için ise aşağıdaki değişikliği yapmamız gerekmektedir.









componentDidMount, componentDidUpdate ve componentWillUnmount'un aksine, useEffect'i birden fazla kez kullanabiliriz.

Bu özellik sayesinde belirlediğimiz değişkenin değişmesi durumunda useEffect ile istediğimiz kodu çalıştırabiliriz ve diğer değişkenlerin durumundan etkilenmeden sadece çalışması gerektiği durumlarda çalıştırmış oluruz.

Umarım anlatabilmişimdir ve verdiğim bilgiler faydalı olmuştur.

Hooks konusunu bir hafta sonu projesi yaparak geniş çerçevede anlatmayı planlıyorum. Proje içerisinde ReactJS, Next.js, PWA, Hooks, Redux vb. konuları kapsayacak sekilde sıfırdan baslayarak detaylı bir anlatımla bölümler halinde









Esen kalın.

About Help Terms Privacy

Get the Medium app







