**Redux Nedir ve Nasıl Çalışır?**

**Redux**, JavaScript tabanlı uygulamalar için kullanılan bir **state management (durum yönetimi)** kütüphanesidir. Uygulamanızdaki tüm verileri merkezi bir "state" objesinde tutarak, bileşenler arası veri akışını daha yönetilebilir hale getirir. React Native gibi uygulamalarda karmaşık veri yönetimini basitleştirmek için yaygın olarak kullanılır.

**Redux'ın Temel Kavramları:**

1. **Store**: Uygulamanın merkezi state'inin (durumunun) tutulduğu yerdir. Tüm state değişiklikleri bu store üzerinden yapılır.
2. **Action**: State'i güncellemek için kullanılan bir JavaScript objesidir. type adında bir özelliği vardır ve state'i değiştirmek için nasıl bir işlem yapılacağını tanımlar. Örnek: { type: 'ADD\_TODO', payload: { text: 'Learn Redux' } }.
3. **Reducer**: Action'lara dayalı olarak state'i değiştiren saf fonksiyonlardır. Her action, store'daki state'i nasıl değiştireceğini bir reducer ile belirtir.
4. **Dispatch**: Bir action'ı tetiklemek için kullanılan fonksiyondur. Dispatch çağrısı, reducer'a action'ı gönderir ve state'in güncellenmesini sağlar.
5. **Selector**: State içindeki belirli bir veriyi almak için kullanılan fonksiyonlardır. Bu, performans açısından optimizasyon sağlar ve kodunuzu daha temiz hale getirir.

### **Redux Toolkit Nedir ve Nasıl Kullanılır?**

**Redux Toolkit**, Redux'ın kullanımını daha kolay ve optimize hale getiren bir araç setidir. Redux Toolkit, özellikle büyük projelerde sıkça kullanılan işlemleri basitleştirir ve "boilerplate" (tekrarlanan kod yazma) miktarını azaltır.

#### Redux Toolkit'in Getirdiği Yenilikler:

1. **ConfigureStore**: Redux store'u oluşturmak için kullanılan, çeşitli middleware'leri otomatik ekleyen bir fonksiyondur.
2. **CreateSlice**: Redux slice'ları oluşturmanın basit bir yoludur. Hem action'ları hem de reducer'ları bir arada tanımlar ve otomatik olarak action creator'ları oluşturur.
3. **CreateAsyncThunk**: Asenkron işlemleri yönetmek için kullanılır (örneğin, bir API çağrısı).

### **Slice Bazlı Dizayn Nedir?**

Redux Toolkit ile birlikte gelen en büyük avantajlardan biri **slice bazlı dizayn** yapabilmektir. Slice, uygulamanın bir parçasına (örneğin kullanıcı verisi, ürün verisi vb.) ait state ve reducer'ları bir araya getirir. Bu, modüler ve bakımı kolay bir yapıya geçiş sağlar.

#### Bir Slice Nasıl Oluşturulur?

* **CreateSlice** fonksiyonu kullanarak state, action ve reducer'ları aynı yapıda tanımlayabilirsiniz.

Bu dizayn, slice'ların bağımsız çalışmasını sağlar ve her bir slice, uygulamanın belirli bir kısmını yönetir. Bu sayede kodun daha temiz, anlaşılır ve yönetilebilir olması sağlanır.

### **Get ve Set İşlemleri Nasıl Yapılır?**

**Redux Toolkit** ile state verilerini almak ve değiştirmek oldukça basittir.

#### Get İşlemi:

State'teki bir veriyi almak için **selector** fonksiyonlarını kullanabilirsiniz.

#### Set İşlemi:

State'teki veriyi değiştirmek için **dispatch** kullanarak action'ı tetikleyebilirsiniz.

Bu şekilde, slice'lar üzerinden hem veri alma (get) hem de veri güncelleme (set) işlemleri kolaylıkla yapılabilir.