

Redux: Genel olarak state'i merkezileştirip uygulamanın karmaşıklığını azaltarak daha yönetilebilir bir state yapmaya çalışır.

Action Yapısı : Nedir bu action?Örneğin önceki projede sepete ekleme-silme için AddToCart ve removeCart operasyonu vardı.Yani actions bizim operasyonlarımız,sil-ekle-şunu yap-bunu yap vb.Actions redux ile ilk iletişimi kurdugumuz noktadır.

Reducer:Action yapısı Reducer ile kontrol altına alınıyor.**Reducer** ilgili alanın içerip içermediğine bakıyor. State ve action'ı parametre alan ve yeni state'i dönen pure bir fonksiyondur. Action'ın type'ına göre state'i değiştirir ve yeni state'i geri döner. Reducer fonksiyonunu store çalıştırıyor.

Store: State'in tutulduğu yerdir. Parametre olarak reducer'ları alır. Reducer'lar birden fazla olduğu zaman bunları bir arada göndermek için **combineReducer** fonksiyonunu kullanırız.Bu yapının tamamamını içeren sistemede **Store** diyoruz.

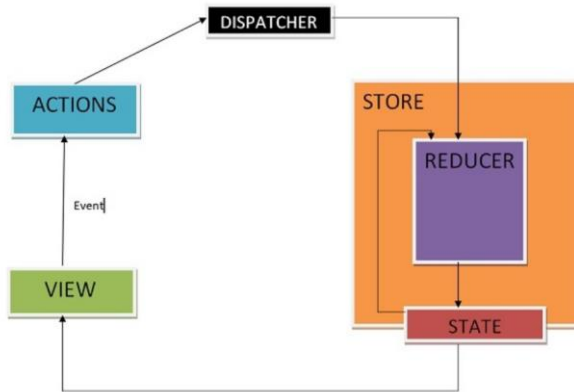
Dispatch: Action'ı parametre alarak reducer'ı tetikler. Bunun sonucunda state değiştirilir.

Özet : İlk başta npm ile redux ve react-redux kur.Burada redux ,redux ın kendisidir.React-redux ise componentlerimizi redux a bağlamak için kullanılan kütüphanedir.

Componentin Reducera Bağlanması

App.js e gel.(projenın başlatıldığı component)

Redux'ın çalışma mantığı.



1. Kullanıcı view'de ,action'u tetikleyecek bir işlem yapar.
2. Dispatch, gelen action'ı store'da mevcut state ile birlikte reducer'a parametre geçer.
3. Bu işlem sonucunda state'in ilgili kısımları değişir ve yeni bir state oluşur. Burada reducer için **Object.assign()** veya **Object spread opretaor** yöntemlerinin önemi anlaşılıyor ,eğer state birden fazla veri tutuyorsa.
4. State güncellendiği gibi abone olan tüm component'ler tekrar render edilir.
5. En sonunda bu state tekrar view'e yansır.

Redux Thunk: Api ye bağlanma gibi asenkron operasyonlarda karşımıza Redux Thunk çıkıyor.

-npm install redux-thunk

*yoksa actionTypesını yaz sonra kendi actionunu yaz

*reducerını yaz ve onu listeye ekle(reducer ın içeriisindeki indexe)

Middleware: araya giren mesela benim iki operasyonum var ben araya başka bir işlem sokuyorum. Sonra aldığım hata→

Error: Actions must be plain objects. Instead, the actual type was: 'function'. You may need to add middleware to your store setup to handle dispatching other values, such as 'redux-thunk' to handle dispatching functions. See <https://redux.js.org/tutorials/fundamentals/part-4-store#middleware> and <https://redux.js.org/tutorials/fundamentals/part-6-async-logic#using-the-redux-thunk-middleware> for examples.

→bu durumda configureStore a git ve ekle.

```
import thunk from "redux-thunk";
```

configureStore n son hali aşağıdaki gibi olmalı

```
You, seconds ago | 1 author (You)
1 | import {applyMiddleware, createStore} from "redux"
2 | import rootReducer from "./index"
3 | import thunk from "redux-thunk";
4 |
5 | export default function configureStore(){
6 |   return createStore(rootReducer,applyMiddleware(thunk))
7 | }
You, 4 hours ago • store configurasyonu yapıldı
```

Hata çözüldü.