# React-Native Android

Chocolatey kullanarak Node ve JDK (Java SE Development Kit) yüklüyoruz.

Yönetici olarak komut istemi açtıktan sonra,

choco install -y nodejs-lts microsoft-openjdk11

Ardından Android Studio kuruyoruz.

Android Studio more actions altından SDK Manager / Sdk Tools ekranından,

Android SDK-Build -Tools 34, Android Emulator, Android SDK ve

Google Play Services, Google USB Driver, Google Web Driver seçip uygula (apply) diyoruz.

Daha sonra Denetim Masası / Kullanıcı Hesapları / Kullanıcı Hesapları sayfasına solda Ortam Değişkenlerini Değiştir seçeneğine girerek Yeni bir User Variable oluşturuyoruz.

Variable Name = ANDROID\_HOME

Variable value = C:\Users\Erkan Hoyman\AppData\Local\Android\Sdk (SDK Manager – Location)

React Native CLI kurmak için, yönetici oalrak açtığımız komut istemi satırında,

>npm install -g react-native-cli

Yeni bir proje oluşturmak için,

# React-Native with Expo Go

>npm install -g expo-cli

Yeni proje oluşturmak için,

npx create-expo-app AwesomeProject

Projemizi çalıştırmak için,

>npm start veya

>npx expo start ile Metro Bundler çalıştırıyoruz.

# React Navigation

Öncelikle bazı paketleri yüklememiz gerekiyor.

npm install @react-navigation/native

npx expo install react-native-screens react-native-safe-area-context

npm install @react-navigation/native-stack (Stack Navigation)

npm install @react-navigation/bottom-tabs (Bottom Tab Navigation)

# Redux Toolkit

<https://redux-toolkit.js.org/> sitesinden quick start adımlarını uyguluyoruz.

Öncelikle gerekli paketleri yüklüyoruz.

>npm install @reduxjs/toolkit react-redux

Bir store.js dosyası oluşturup **configureStore** ile **store** tanımlayarak dışarıya açıyoruz.

Ardından App.js içinde store.js ve **Provider** import ederek, tüm navigator ve screen’lerimizi provider ile sarmallıyoruz. Props olarak store dosyamızı geçiyoruz.

<Provider store={store}>

….

</Provider>

Projemizde takip etmek istediğimiz her bölümle ilgili **slice** dosyaları (counterSlice.js) oluşturuyoruz.

Ardından slice dosyamızdaki action’ları dışarıya açıyoruz. Dosyamızı da default olarak reducer ismiyle dışarıya export ediyoruz.

Tüm slice dosyalarımızı temsil eden reducer’ları store.js sayfasında import edip, reducer objelerine ekliyoruz.

import counterReducer from './counterSlice'

export const store = configureStore({

  reducer: {

    counter: counterReducer,

  },

})

Redux state ve action’lara (reducer’lara) istediğimiz bir componentte erişebilmek için **useSelector** ve **useDispatch** hooks kullanıyoruz.

import { useSelector, useDispatch } from "react-redux";

import { decrement, increment, incrementByAmount } from "./store/redux/counterSlice";

Sayfaya import ettikten sonra useSelector ile state değerlerimizi çekiyoruz.

  const count = useSelector((state) => state.counter.value);

useDispatch ile de reducer’larımızı (action-fonksiyonlar) çağırıyoruz.

  const dispatch = useDispatch();

onPress={() => dispatch(increment())}