React İle Proje Geliştirmek

Berkay AKAR

İçindekiler

[Kurulum – Dosya Yapısı 0](#_Toc160377714)

[React için Dosya Yapılanmaları 3](#_Toc160377715)

# Kurulum – Dosya Yapısı

Bu dökümanda bir React projesinde kullanabileceğimiz hemen hemen her paketi ve yapıyı gözlemleyeceğiz. Başlangıç için bir react projesi kurmamız gerekmektedir. Bu aşamada Node.js i kurmamız gerekecektir.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıda gösterildiği gibi ilgili platformdan ilgili paketleri indirebilirsiniz. Temelde kısaca Node.js ten bahsetmek istiyorum. Javascript esasında client side (istemci tarafında) çalışan bir dildi. Tarayıcıda çalıştıdğımız V8 Engine kullanıma sunuldu ve artık sunucu tarafında da (server side) javascript dilini kullanabiliyoruz. Javscript in Server Side bir şekilde sunucu tarafında kullanmamıza olanak sağlayan **V8 Engine Node.JS olarak** adlandırılmaktadır.

Kurulumundan emin olmak için “node -v” veya “node – version” komutlarını çalıştırdığınızda karşınıza bir versiyon numarası geliyorsa bu durumda kurulum başarı ile gerçekleşmiştir manasına gelecektir. (versiyon çıkıyorsa node kurulmuştur)

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Node.js kurulduğunda varsayılanda NPM ( Node js paket yöneticisi ) kurulu olarak gelmektedir. Neticesinde sizler “npm -v” diyerek ilgili npm sürümünü kontrol edebilir, kurulu olup olmadığını gözlemleyebilirsiniz.

metin, ekran görüntüsü, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

React in kendi sayfasında Create React App ile bir react projesi oluşturabilirsin

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Npx npm in belirli bir sürümünden sonra gelen bir yapıdır. Eğer makinenizde npx yoksa aşağıdaki komut ile birlikte npx i kurabilirsiniz.

NPM sadece bir paket yöneticisidir. Paketleri yüklemek, kaldırmak ve güncellemek için kullanırız. NodeJS ile CLI uygulamaları oluşturmaya başladığımızda bazı sorunlar ortaya çıktı.

NPM de kullanacağımız CLI uygulamalarını global olarak bilgisayarımıza yüklememiz gerekiyordu. Bu da dizinlerde yazma izinleri ya da aynı paketin birden fazla versiyonunu kullanamama gibi bazen bazı sorunlara neden oluyordu. Ya da bir CLI uygulamasını hızlıca kullanabilmenin yanında zahmetli kalıyordu. v5.2.0 sürümü ile birlikte NPM npx adında bir aracı duyurdu. Bu araç bir CLI uygulamasının bilgisayarımızda global olarak kurulu olsun ya da olmasın kullanılabilmesini sağlıyor. Örneğin v5.2.0 sürümünden önce create-react-app CLI uygulamasıyla bir proje oluşturmak için önce bu paketi global olarak kurmak ve daha sonra çalıştırmak gerekiyordu.

metin, ekran görüntüsü, iş kartı, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, iş kartı içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu yönüyle farklı node bağımlılığı olan projelerde (nvm — Node Version Manager) gibi araçlardan kaçınmamıza olanak sağlıyor. npx komutu çalıştırıldığında belirtilen paketi global olarak kurulmuş paketler arasında arar. Eğer daha önceden yüklenmiş bir paketse lokalden kullanarak hızlıca çalıştırır. Fakat yüklenmemiş bir paketi kullanmak istersek, önce paketin dosyalarını geçici olarak yüklemek için bekletecek daha sonra komutu çalıştıracaktır. Ayrıca npx ‘in bir şeyleri çalıştırma gücü npm paketleri ile sınırlı değildir. Doğrudan bir url aracılığıyla Github Gist üzerindeki rastgele bir kod parçacığını çalıştırabilirsiniz.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıda görüldüğü gibi NPX versiyonunuda bu şekilde belirleyebilirsiniz. Yada npx bilgisayarınızda kurulu değil ise aşağıdaki komut vasıtası ile birlikte npx kurulumunu global olarak bilgisayarınızda gerçekleştirebilirsiniz.

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bunlar geliştirme ortamı için lazım olan ortamdır. Yani sunucuda buna lazım olmayacaktır. Bu aşamdan sonra ilk react projemizi create react app diyerek kurulumunu gerçekleştirebiliriz.

metin, yazı tipi, grafik, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

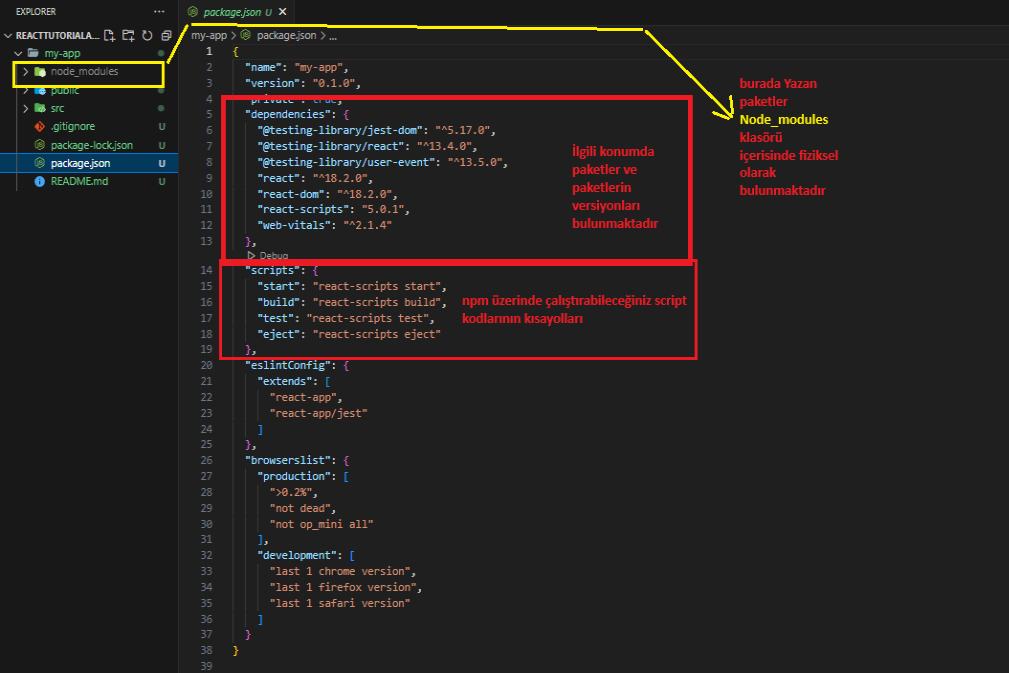
# React için Dosya Yapılanmaları

Kurulum ve paket yükleme işlemleri bu aşamada gerçekleşmektedir. Aşağıda proje klasörlemesi ve dosyaları gözükmektedir.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Dosyaları ve paket yapılanmalarını anlatalım. **Package.json** bizlere paket yapılanmalarını ve paketlerin versiyonlarını göstermektedir. ( **node\_modules** klasörü içerisinde package.json içerisinde yer alan paketler fiziksel olarak yüklenmektedir. )



* **Public** klasörü altında olanların tamamı build alındığında build klasörü içerisine gitmektedir. (index.html bu klasördedir. Uygulama üzerinden ulaşılabilecek html css resim ve js dosyalarının tutulduğu yerdir. Statik dosyaları bu klasörde tutabiliriz. İmport edilmeden direkt olarak kullanabilmemize olanak sağlamaktadır. Aslında bakıldığında src klasörü içerisinde de dosyalar konulabilir ve import dosyası ile çıkışı sağlanabilir. Ancak sizler build aldığınızda her seferinde ilgili statik dosyalar guid bir isimlendirme ile export olacaktır. Bu sebepten ötürü her buildde ismi değişeceğinden ötürü sabit bir isimlendirme gereksinimi olduğu zaman bu klasör içerisinde sabit bir tanımlama yapabilirsiniz )

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

* **Gitignore** : versiyon kontrol sistemlerinde git sistemine gönderilmeyecek olan dosyalar için ilgili konfigürasyon yapılmaktadır.
* **Readme.md** : ilgili proje için readme.md dosyasıdır. Proje tanıtımının gerçekleştiği dosya.

İlgili olarak bizim bakmamız gereken yer genel olarak src klasörünün altıdır. Src klasörüne tıklandığında ilk olarak bizler src klasörüne baktığımızda index.html ve index.js dosyasını incelemeye başlayalım.

* **İndex.js :** react ve react.dom kütüphaneleri burada import olmaktadır. Css importu gerçekleşmiş ve reportwebvitasl kısmına gerek yok onu kaldırabiliriz.

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

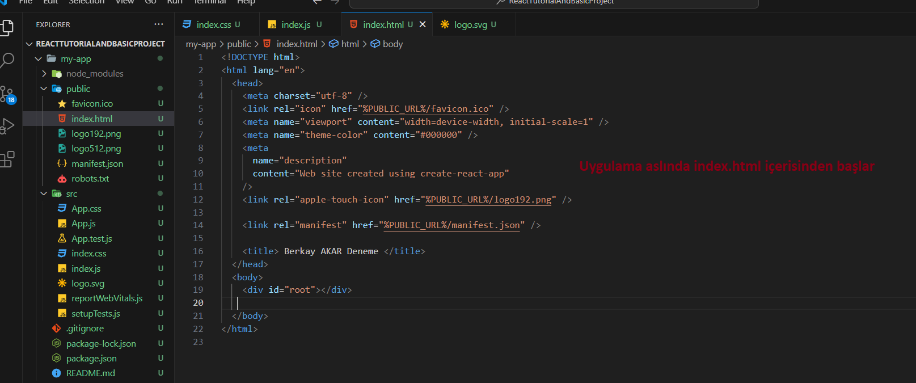
Açıklama otomatik olarak oluşturuldu**

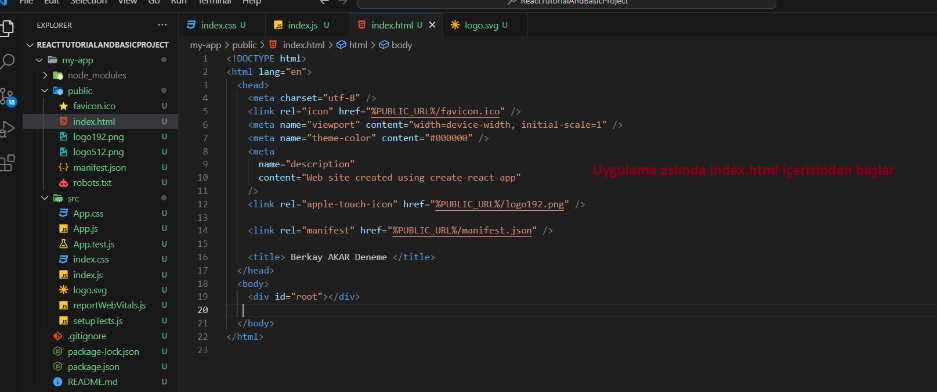
Kodlardan anlaşılacağı üzere const root değişkeni vasıtası ile index.html içerisinde root isimli dom manipülasyonu ile almaktadır. İlgili root içerisinde app.js ismindeki komponenti basmaktadır. App.js bir komponenttir. Bu komponenti render etmektedir. En basit yapılandırma ile index.js aşağıda gözüktüğü gibidir.

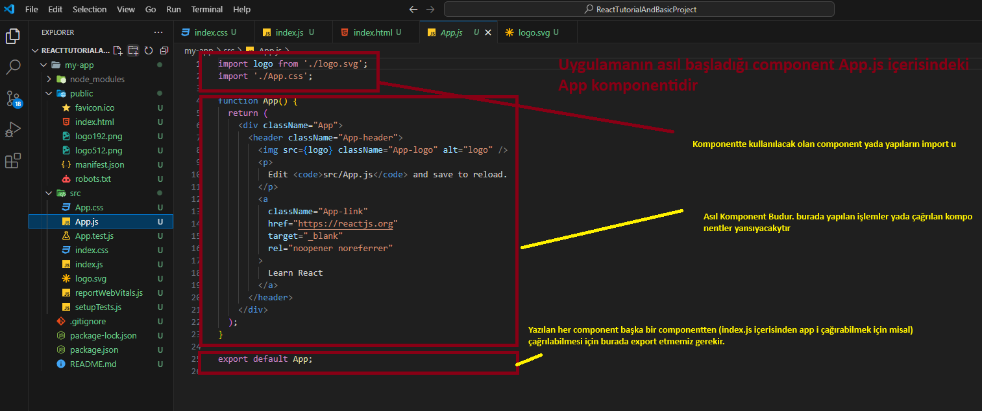
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

* **App.js** : bu bir react komponentidir. React komponentinde bir komponent oluşturuyorsunuz (class component ve fonction component) ilgili function içerisinde kodlarınızı yazıp render ediyorsunuz ve ilgili componenti export olarak dışarı açıyorsunuz.



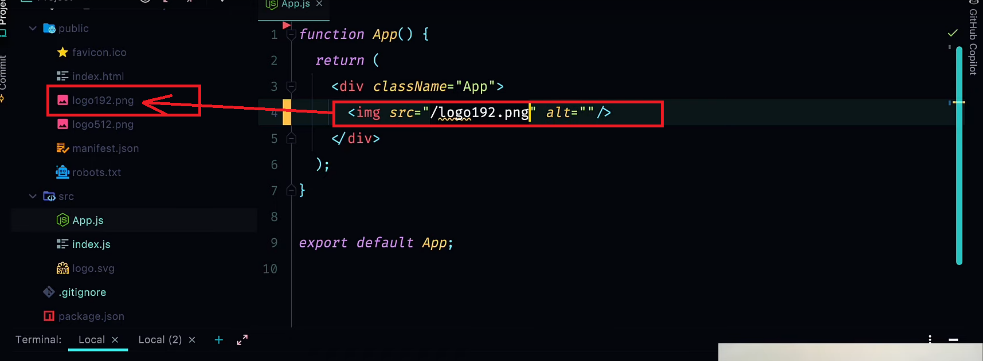




Temel bir react projesindeki komponent sıralaması bu şekilde gerçekleşmektedir. En başlangıçta app.js ve index.js bulunmaktadır.

Dosya yapılandırması bu şekildedir.

Public klasörü dışarı açık bir klasördür. Wwwroot gibi düşüebiliriz. Buraya eklediğimiz statik değerlere direk / diyerek ulaşabilmekteyiz.



src içerisindeki statik olan değerlere (örnek vermek gerekirse logo.svg ) gibi bir değere ulaşmak isteseydik o zaman bizlerin yapması gereken şey import edip o şekilde kullanmamız gerekecektir.

