



BATCH : B45-46-71  
LESSON : **REACTJS**  
DATE : 16.04.2022  
SUBJECT : **FUNDAMENTALS**



 [techproeducation.com](https://techproeducation.com)

 [info@techproeducation.com](mailto:info@techproeducation.com)

 +1 (917) 768-7466



## Uygulama oluřturma

Terminal'de ařağıdaki kodlar sırayla alıřtırılır

```
npx create-react-app my-app  
cd my-app  
npm run start
```

my-app isminde bir react  
projesi oluřturur

my-app klasrne girer

Projeyi alıřtırır



## Uygulama oluřturma

```
Windows PowerShell
Starting the development server...
Compiled successfully!

You can now view myapp in the browser.

  http://localhost:3000

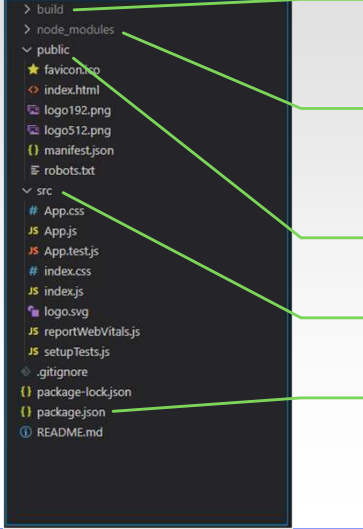
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Terminate batch job (Y/N)? _
```

Uygulamayı durdurmak için klavyeden  
CTRL + C ye basılır



## React Klasör Yapısı



**build**, proje tamamlandıktan sonra, **npm run build** komutu ile yayına hazırlandığında oluşan klasördür.

**node\_modules** klasörü react ta kullanacağımız tüm kütüphaneleri içerir. **npm install** komutu ile kurduğumuz kütüphaneler burada tutulur. Proje paylaşılrken bu klasör paylaşılmaz

**public**, projedeki statik dosyaların tutulduğu klasördür.

**src**, projenin kaynak dosyalarını tutmaktadır. React kodlarını barındıran dosyalar burada bulunmalıdır.

Proje ile ilgili genel bilgiler ve projede kullanılan kütüphane bilgileri **package.json** dosyasında bulunur.



## Package.json

```
package.json > ...
1 {
2   "name": "test",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^5.14.1",
7     "@testing-library/react": "^11.2.7",
8     "@testing-library/user-event": "^12.8.3",
9     "react": "^17.0.2",
10    "react-dom": "^17.0.2",
11    "react-scripts": "4.0.3",
12    "web-vitals": "^1.1.2"
13  },
14  "scripts": {
15    "start": "react-scripts start",
16    "build": "react-scripts build",
17    "test": "react-scripts test",
18    "eject": "react-scripts eject"
19  },
```



Projede kullanılan npm kütüphanleri. Eğer node\_modules klasörü yoksa yada sorun varsa, npm install komutu çalıştırılırsa burada yazan paketler tekrar kurulacaktır.



# Package.json

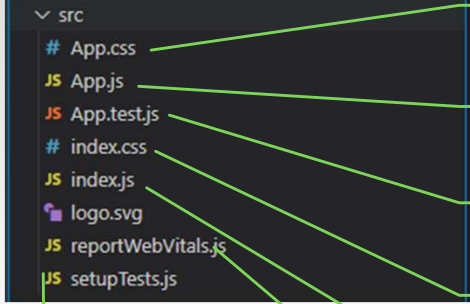
```
package.json > ...
1 {
2   "name": "test",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^5.14.1",
7     "@testing-library/react": "^11.2.7",
8     "@testing-library/user-event": "^12.8.3",
9     "react": "^17.0.2",
10    "react-dom": "^17.0.2",
11    "react-scripts": "4.0.3",
12    "web-vitals": "^1.1.2"
13  },
14  "scripts": {
15    "start": "react-scripts start",
16    "build": "react-scripts build",
17    "test": "react-scripts test",
18    "eject": "react-scripts eject"
19  },
20}
```

- **npm start:** uygulamayı lokalde çalıştırır
- **npm build:** uygulamayı yayına hazırlar. build klasörünü oluşturur
- **npm test:** test fonksiyonlarını çalıştır
- **npm eject:** react'ın build aracından ve yapılandırma seçeneklerinden memnun değilseniz, kendi build aracınızı oluşturmak için kullanabilirsiniz.





## Src klasörü



App isimli componentin css dosyası

App isimli component

App isimli componentin test fonksiyonu

Uygulamanın genel css dosyası

Uygulamanın giriş dosyası

Uygulamanın performansını ölçmek için kullanılan dosya

Test ayar işlemleri yapılır



## Dokümantasyon







# React

- › Rendering
- › Component hiyerarşisi
- › Component oluşturma
- › JSX

## Fundamentals



## Uygulamanın render edilmesi

public/index.html

```
<body>
  <div id="root"></div>
</body>
```

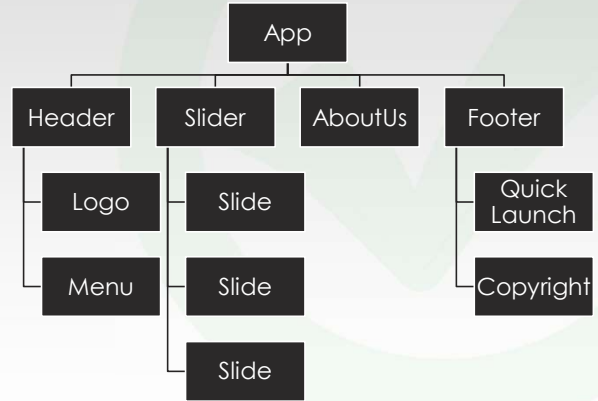
src/index.js

```
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

Bu aşama uygulama yüklenirken React Virtual DOM un oluşturulduğu kısımdır. index.html içinde «root» ismiyle tanımlanan div, react ana dosyası olan index.js içindeki `document.getElementById("root")` komutu ile seçilir. ReactDOM.render methoduyla hem Virtual DOM oluşturulur hem de Ract app bu elementin içine yerleştirilir.



## Component Hiyerarşisi



REACT app, componentlerden oluşmaktadır. Componentler iyi dizayn edildiğinde uygulamayı yazmak daha kolaylaşmaktadır. React ta bir ana component bulunmaktadır. Diğer tüm componentler bu ana componentte bağlıdır.



## Component oluşturma

```
const MyComponent = () => {  
  return (  
    <div>  
      ...  
    </div>  
  );  
}  
  
export default MyComponent;
```

Component ismi büyük harfle başlamalıdır. (PascalCase)

Dışarıdan parametre alırsa buraya yazılır

Component in içeriği burada oluşturulup kullanıldığı yere return edilir. Bu kısımda JSX yazılmalıdır

Component in başka bir yerden import edilebilmesi için burada export edilmelidir.



## Component kullanma

```
import MyComponent from './MyComponent';
```

Component import edilir

```
const App = () => {  
  return (  
    <div className="App">  
      <MyComponent/>  
    </div>  
  );  
}  
  
export default App;
```

İmport edilirken  
kullanılan isimle  
eklenir