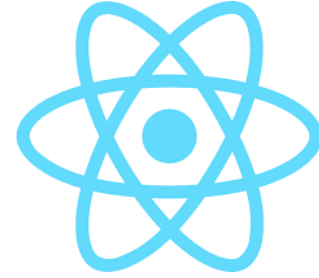


# Components (Bileşenler)

Onur KULABAŞ

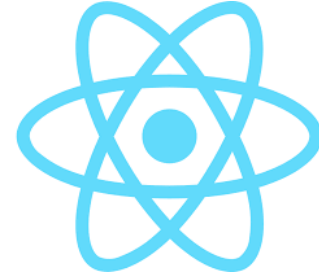
onurkulabas@yahoo.com

React Training  
Istanbul, 2022



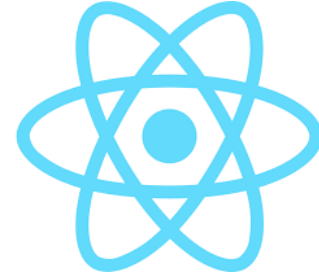
# Components

**Bileşenler (components)**, kullanıcı arayüzünü ayrıştırarak birbirinden bağımsız ve **tekrar kullanılabilen (reusable)** parçalar oluşturmanızı sağlar. Bu sayede her bir parçayı, birbirinden izole bir şekilde düşünerek kodlayabilirsiniz.



# Components

Kavramsal olarak bileşenler, JavaScript **fonksiyonları** gibidir. Bileşenler, **opsiyonel** olarak “**props**” adındaki **girdileri** alırlar ve ekranda görüntülenecek React elementlerini geri döndürürler. Props kavramına bir sonraki modülde değineceğiz.

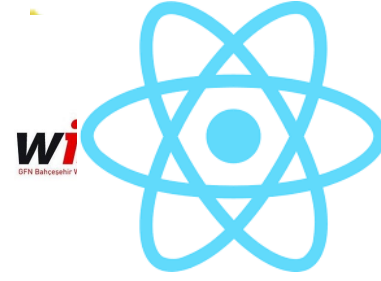


# Örnek Bir Kullanım

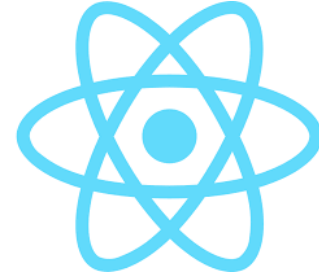
```
function Welcome() {  
  return <h1>Hello World!</h1>;  
}
```

```
const element = <Welcome />;  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

Yukarıdaki örnekte Welcome **fonksiyonu** içindeki Hello World! Metnini basabilmek için aynı **HTML tag'i gibi <Welcome />** şeklinde çağırabiliyoruz.



# Birden Fazla Sayfalı Menu Örneđi

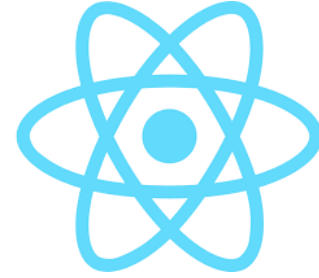


## **Eklenecek Paketler**

Projemizde birden fazla sayfa olacağı için ve sayfalar arasında linkler kurulması gerektiğinden dolayı react-router-dom paketine ihtiyaç duyulmaktadır. Yüklemek için, projenin ana klasöründe iken,

**npm install react-router-dom**

komutu çalıştırılmalıdır.

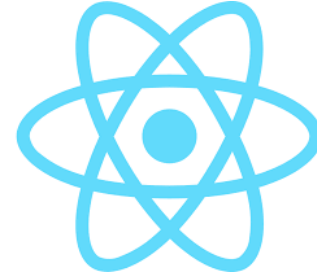


## index.js dosyası

index.js dosyası aşağıdaki şekilde güncellenmelidir.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import { BrowserRouter } from "react-router-dom";

ReactDOM.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
  document.getElementById("root")
);
```



# index.js - Açıklamalar

## Not 1

```
import { BrowserRouter } from "react-router-dom";
```

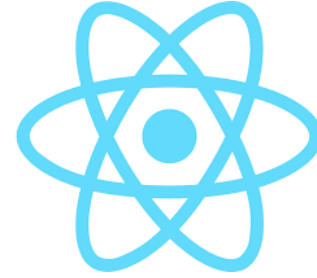
Yukarıdaki satırda `react-router-dom` paketi içinden sadece `BrowserRouter` isimli bileşeni almış olduk.

## Not 2

```
<BrowserRouter>  
  <App />  
</BrowserRouter>
```

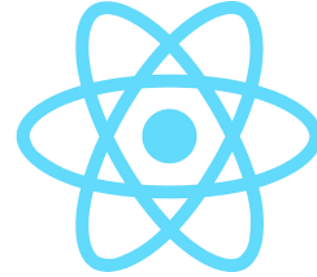
Yukarıdaki satırlarda `<App/>` bileşeni `<BrowserRouter>` bileşeni içine alınarak uygulamanın App ile start alması ve içinde farklı sayfalara bağlantı verebilmesi sağlandı.





## Yeni Eklenecek Dosyalar

src klasörüne sağ tıklayarak 3 adet yeni dosya ekleyin ve isimlerini **Home.js**, **About.js** ve **Contact.js** olarak verin. Bu dosyalar uygulama içinde bir üst menü aracılığıyla erişilecek olan 3 dosyayı temsil edecektir. Dosyaların içi şimdilik boş kalabilir.

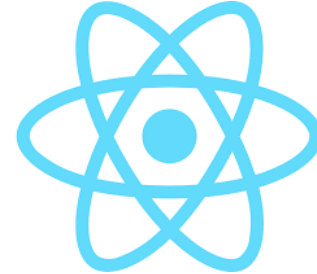


# App.js Dosyası

App.js dosyası aşağıdaki şekilde güncellenmelidir.

```
import { Routes, Route } from "react-router-dom"
import Home from "./Home"
import About from "./About"
import Contact from "./Contact"

function App() {
  return (
    <div className="App">
      <Routes>
        <Route path="/" element={ <Home/> } />
        <Route path="/about" element={ <About/> } />
        <Route path="/contact" element={ <Contact/> } />
      </Routes>
    </div>
  )
}
export default App
```



# App.js - Açıklamalar

## Not 1

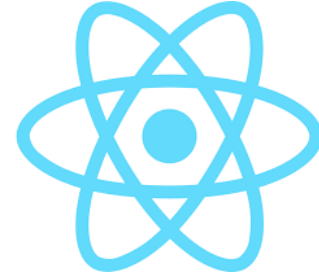
```
import { Routes, Route } from "react-router-dom"
```

Yukarıdaki satırda `react-router-dom` paketi içinden sadece `Routes` ve `Route` isimli bileşenleri almış olduk.

## Not 2

```
import Home from "./Home"
```

Yukarıdaki satırda `./Home` bileşenini `Home` adı ile kullanabilmek için import ettik. Yani artık `<Home/>` yazdığımızda bu sayfadaki içeriği aynen kullanabileceğiz. About ve Contact sayfaları için de aynı kurallar geçerlidir. Uygulamada kullanılacak olan **sayfalar** bu şekilde **eklenmelidir, aksi halde** adres bar'dan o sayfalara **erişim sağlanamaz**.

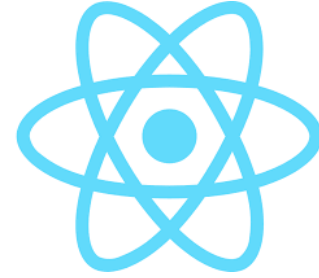


# App.js - Açıklamalar

## Not 3

```
<div className="App">
  <Routes>
    <Route path="/" element={ <Home/> } />
    <Route path="/about" element={ <About/> } />
    <Route path="/contact" element={ <Contact/> } />
  </Routes>
</div>
```

Yukarıdaki satırlarda **Routes** altındaki her bir **Route** bileşeni element attribute'u ile bileşen **adını**; path attribute'u ile sayfaya **nasıl erişileceğini** tanımlar. Örneğin **/about** adresinden **About** bileşeni yani About.js sayfası çağrılacaktır. Tek başına yazılan **/** karakteri ise root yani ana dizini temsil eder.



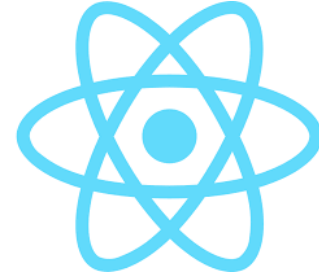
## Yeni Eklenecek Menu Dosyası (Bileşeni)

src klasörüne sağ tıklayarak Yeni Dosya ekleyin ve adını **Menu.js** olarak verin. İçine aşağıdaki satırları ekleyin.

```
import { Link } from "react-router-dom";

function Menu() {
  return (
    <div>
      <Link to="/"><b>Home</b></Link>&nbsp;
      <Link to="/about"><b>About</b></Link>&nbsp;
      <Link to="/contact"><b>Contact</b></Link>
    </div>
  );
}

export default Menu;
```



# Menu.js - Açıklamalar

## Not 1

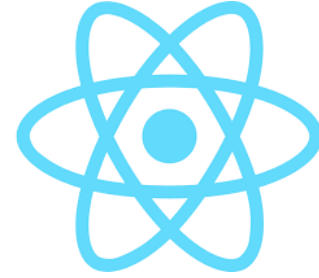
```
import { Link } from "react-router-dom";
```

Yukarıdaki satırda `react-router-dom` paketi içinden sadece `Link` isimli bileşeni almış olduk.

## Not 2

```
<Link to="/about"><b>About</b></Link>
```

Yukarıdaki satırda kullandığımız `<Link to` bileşeni düz HTML'deki `<a href` kullanımına denktir. Bu şekilde yazdığımız 3 adet link aralarında birer boşlukla yan yana yazılacak ve bileşen olarak çağrıldığı sayfalarda ortak bir menü şeklinde davranacaktır.

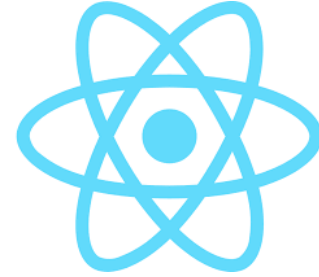


# Menu.js - Açıklamalar

## Not 3

```
export default Menu;
```

Yukarıdaki satırda Menu sayfasını Menu adlı bir bileşen (component) olarak **export** etmiş yani dışarı çıkarmış olduk. Artık diğer sayfalar **Menu'yu önce import edecek**, sonrasında da kod bloku içinde **<Menu/> yazarak** kullanabilecek.



# Home.js

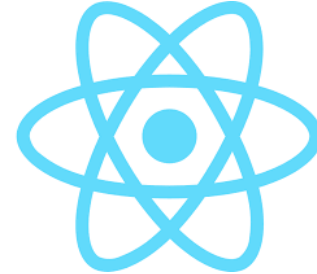
İlk etapta içi boş olan dosyalarımızdan Home.js dosyasını açın ve aşağıdaki satırları içine yazın.

```
import Menu from "../Menu"

function Home() {
  return (
    <div>
      <Menu />
      <h1>This is the Home page!</h1>
    </div>
  );
}

export default Home;
```





# Home.js - Açıklamalar

## Not 1

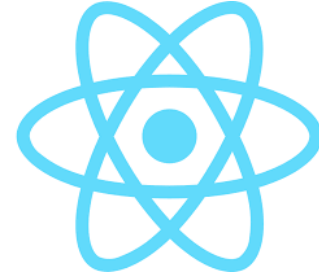
```
import Menu from "../Menu"
```

Yukarıdaki satırda `"../Menu"` bileşenini `Menu` ismiyle kullanabilmek amacıyla import ettik.

## Not 2

```
<Menu />
```

Yukarıdaki satırda bileşenini çağırdık. Böylece Menu.js dosyasındaki içeriği aynen kullanmış olduk, yani ortak menuyu çağırdık.



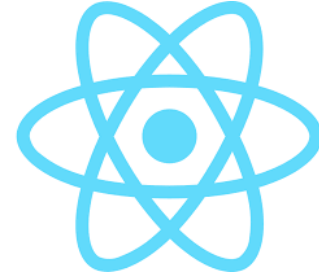
# About.js

İlk etapta içi boş olan dosyalarımızdan About.js dosyasını açın ve aşağıdaki satırları içine yazın.

```
import Menu from "../Menu"

function About() {
  return (
    <div>
      <Menu />
      <h1>This is the About page!</h1>
    </div>
  );
}

export default About;
```



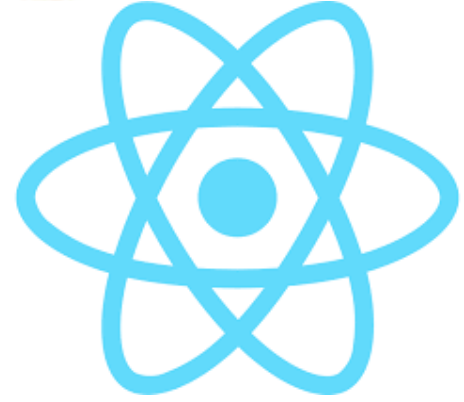
# Contact.js

İlk etapta içi boş olan dosyalarımızdan Contact.js dosyasını açın ve aşağıdaki satırları içine yazın.

```
import Menu from "../Menu"

function Contact() {
  return (
    <div>
      <Menu />
      <h1>This is the Contact page!</h1>
    </div>
  );
}

export default Contact;
```



Teşekkürler 😊

Onur KULABAŞ  
onurkulabas@yahoo.com

<http://tr.linkedin.com/in/onurkulabas>