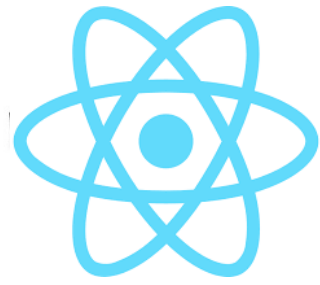


Axios

Onur KULABAŞ

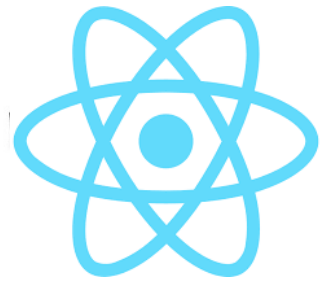
onurkulabas@yahoo.com

React Training
Istanbul, 2022



Axios

Axios, React içinde **servis (API) çağırmak** için kullanacağımız kütüphanedir. Bildiğimiz gibi React sadece bir önyüz kütüphanesidir; yani backend'e etki eden herhangi bir kod yapısına sahip değildir. Bu bağlamda veri okuyup yazabilmek ve haberleşebilmek için web servislerini çağırması, yanıtları alması ve işlemesi gerekmektedir. Axios işte bu konuda kullanılacaktır.



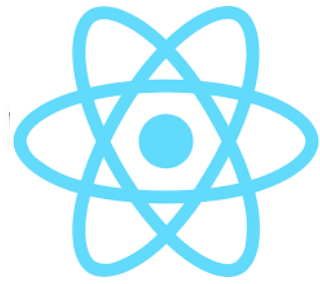
Axios – Post Servisi Çağırarak

```
import { axios } from 'axios';
```

Yukarıdaki kodu login sayfasının en üstüne yapıştıralım. Önceki örneklerde olduğu gibi Axios kütüphanesinden **axios** **objesini import** etmiş oluyoruz.

```
const myButtonClick = async () => {  
  
  let requestBody = {  
    service_val_name:name,  
    service_val_password:password  
  }  
  
  const response = await axios.post ('https://www.mockachino.com/1b9b9eca-13b9-41/login',  
requestBody);  
  
  if (response.data.result=="success") {  
    setResult("Giriş Başarılı");  
  } else {  
    setResult("Hatalı kullanıcı adı veya şifre");  
  }  
}
```

Yukarıdaki kodu da ana function içine yerleştirelim. Daha önce dummy verilerle örnek senaryo ile kontrol ettiğimiz kullanıcı adı ve şifre alanlarını requestBody adındaki object türündeki değişkene atamak suretiyle adresi yazılan **servise post ediyoruz**.



Axios - Notlar

Not 1

Daha önce de örneğini gördüğümüz üzere function'ımızı **async** () keyword'ü ile tanımladık ve paralel yerine seri çalışmasını sağladık. Servisi çağıran satırda da **await axios.post** kalıbını kullandık. **await** keyword'ü ile de servisin ilgili sırada çağrılmasını sağladık.

Not 2

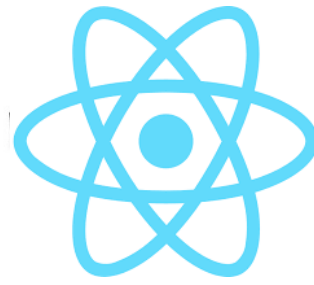
API'ler **get, post, put ve delete** olmak üzere temelde 4 tip çağırma şekline sahiptir. Bu konu ile ilgili detayları backend programlama derslerinde vereceğiz. Fakat React bu tip service call'ları fazlasıyla yapacağından temel işlemleri bilmemiz gerekiyor.

get : Veri getirmeye yarar. CRUD işlemlerinden **Read**, SQL'de SELECT gibidir.

post: Veri göndermeye yarar. CRUD'un **Create**'i, SQL'in INSERT'ü gibidir.

put: Veri güncellemeye yarar. CRUD'un ve SQL'in **UPDATE**'i gibidir.

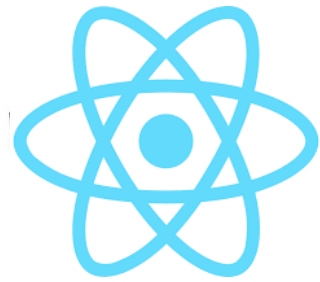
delete: Veri silmeye yarar. CRUD'un ve SQL'in **DELETE**'i gibidir.



Axios - Notlar

Not 3

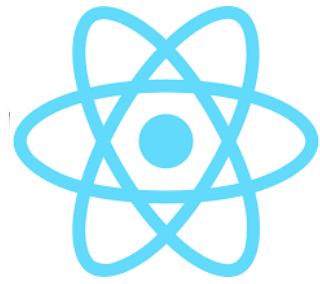
Servisi post metodu ile çağırdığımızda şeklinde çağırdığımız response adlı sabitimize dönen data'yı incelememiz gerekir. Bu amaçla servise post işlemini takiben tarayıcımızda sağ tıklayıp **İncele (Inspect)**'e basalım ve **Network** tab'ında bize dönen değerleri inceleyelim. **Response** içinde data içinde **result** adında bir alt değeri göreceğiz. Orada success yazıyorsa işlem başarılı, aksi halde başarısız (service call işlemi başarılı fakat login hatalı yani girilen kullanıcı ve şifreyle eşleşen bir kayıt yok) demektir.



Axios – Get Servisi Çağırarak

```
const [firstName, setFirstName] = useState('');
useEffect
(async() =>
{
  const response = await
axios.get('https://www.mockachino.com/c30e50d6-aca1-47/users');
  console.log(response);
  setFirstName(response.data.users[0].first_name);
},
[]
);
```

Yukarıdaki kodu yine ana function içine yerleştirelim. Bu defa bir **get** servisi çağırıyoruz. **response** sabitini bu servisten dönecek olan değere eşitliyoruz. Bu örnekte response değerini **console'a** da yazdırdık ve get servisi çağırıldığı anda tarayıcımızda sağ tıklayıp **İncele (Inspect)**'e basarak **console** tab'inden de response değerini takip edebiliriz. Mevcuttaki response'ın içinde data, onun içinde de users başlığı altında first_name ve diğer alanları görebiliriz. Users altında aynı formatta birden fazla users gelebilir. O nedenle biz bu örnekte firstName adlı değişkene ilk okuduğumuz satırın (yani users[0]) first_name alanını eşitledik.



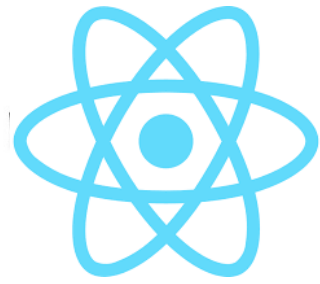
Axios – Login Kontrolü Ek Çalışma

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

Yukarıdaki kodu login sayfasının en üst kısmına yapıştıralım. Bu şekilde react-router-dom kütüphanesinden Navigate objesini kullanacağız. Böylelikle sayfaları yönlendirme (redirect) gibi navigasyon işlerini yapabileceğiz.

```
if (response.data.result=="success") {  
    localStorage.setItem("userName", "onurkulabas")  
    return <Navigate to="/" />  
} else {  
    setResult('Hatalı Kullanıcı Adı veya Şifre');  
}
```

Yukarıdaki satırları myButtonClick'in içine yerleştirelim. Açıklamaları da şu şekilde yapalım.



Axios – Login Kontrolü Notlar

Not 1

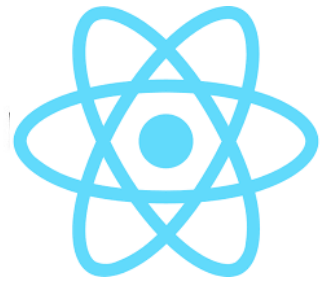
```
localStorage.setItem("userName", "onurkulabas")
```

userName adlı değişken burada session (oturum) mantığında kullanılan ve diğer sayfalar tarafından da okunabilen bir değişkendir.

Not 2

```
return <Navigate to="/" />
```

Sayfayı ana sayfaya (Home.js) yönlendirir. Home ve diğer tüm kontrol gerektiren sayfalarda **userName** adlı localStorage değişken değerinin boş olup olmadığı kontrol edilecektir. Böylelikle login olmamış kullanıcıların ilgili sayfalara adres yazarak gitmesi engellenecektir. Bir sonraki adımda bu örneği yapacağız.

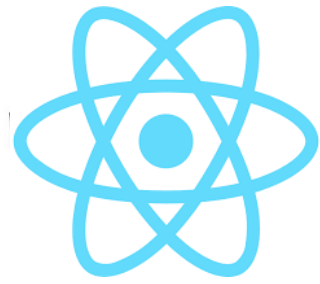


Axios – Login Kontrolü (Home)

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

Örnek olarak Home sayfasını açalım ve yukarıdaki kodu bu sayfanın da en üst kısmına yapıştıralım. Bu sayfada da react-router-dom kütüphanesinden Navigate objesini kullanacağız. Aşağıdaki kodu da ana function içine en üst kısma yerleştirelim.

```
const navigate = useNavigate();  
useEffect  
(  
  ()=>  
  {  
    if (localStorage.getItem("userName")== "")  
    {  
      navigate("/login");  
    }  
  }  
);
```



Axios – Login Kontrolü Notlar

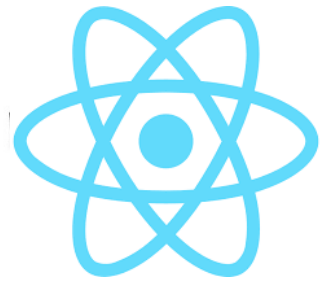
Not 1

```
if (localStorage.getItem("userName")== "")  
{  
    navigate("/login");  
}
```

Yukarıdaki satırla birlikte login aşamasında başarılı olunması halinde set edilen localStorage userName değişkeninin boş olması halinde login ekranına gönderilmesini sağladık.

Ek olarak aşağıdaki satırı da return () ifadesinin içine yerleştirelim ve login olmuş kişiye Merhaba diyelim.

```
Welcome, { localStorage.getItem("username") }
```

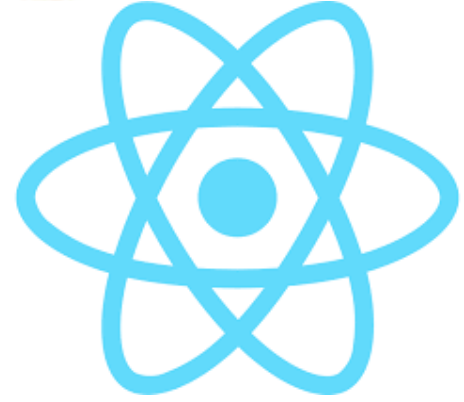


Axios – Login Kontrolü Notlar

Not 2

```
useEffect
(  
  ()=>  
  {  
    ...  
  }  
);
```

Yukarıdaki useEffect yazım şekliyle sayfa açılır açılmaz çalışması istenen kodları (login kontrolü veya daha önceki get örneğindeki gibi sayfa açıldığı anda çalışacak bir service call v.b.) çalıştırıyoruz.



Teşekkürler 😊

Onur KULABAŞ
onurkulabas@yahoo.com

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