

Kadir ÇAPKIN 21360859023

NodeJS ile Web programlama Raporu Hafta – 5

Weather app – 1

Bu hafta derse asenkron fonksiyonların çalışma mantığıyla başladık. Asenkron fonksiyonlar derleme esnasında yukarıdan aşağı bir şekilde kod okunurken kendi fonksiyonunun bitmesi için bloklama yapmaz. Asenkron fonksiyon başlatılır ardından aşağıdaki kodlar asenkron fonksiyonu beklemeden çalıştırılır ve kod asenkron fonksiyonun kendini sonlandırmasıyla biter. Bundan ilk haftada bahsetmiştik nodejs event driven tabanlı non-blocking yapıdadır.

Kod örneğimize bakacak olursak:

```
console.log("Basla");

setTimeout(()=>{
  console.log("2 Saniye bekliyorum")
},2000);

console.log("Bitir");
```

Normalde bu çıktığımız asenkron olmasaydı Basla-2 Saniye bekliyorum- Bitir şeklinde olmasını beklerdik. Asenkron olduğu içinse çıktımız şu şekilde gözükür

```
Basla
Bitir
2 Saniye bekliyorum
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\WeatherApp> |
```

Başka bir kod örneğine bakacak olursak :

```
console.log("Basla");

setTimeout(()=>{
  console.log("2 Saniye bekliyorum")
},0);

console.log("Bitir");
```

Burada ise parametre olarak bu fonksiyon 0 saniye beklenmesi gerektiği için mantık olarak sanki senkron yapıda çalışacakmış gibi düşünüyoruz. Fakat durum böyle değil kod çıktımız

```
Basla
Bitir
2 Saniye bekliyorum
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\WeatherApp>
```

Bunun sebebi asenkron fonksiyonlar Stack mantığında çalışır. Bu stack mantığı şu şekilde çalışır biz bir fonksiyon çağırdığımızda bu bu stack'e eklenir. Main fonksiyonu asenkron fonksiyonla göre her zaman stack'e önce eklenir. Yukarıdaki örneğe bakarsak önce `console.log("basla");` Çıktısı stack'e eklenir çalıştırıldığı gibi stack'ten çıkartılır. Ardından asenkron yapıdaki `setTimeout` fonksiyonuna bakarsak bu fonksiyon stack'e eklenir fakat hemen çıkartılmaz süresi dolana kadar orada bekletilir. Ardından `console.log("Bitir");` fonksiyonu stack'in en üstüne eklenir ve çalıştığında çıkartılır. Burada `setTimeout` fonksiyonun bekleme süresini 0 versek hemen ardından gelen fonksiyon bir main fonksiyonu olduğu için `setTimeout` fonksiyonu stack'in en altında kalmış olur. Bu da en son çıkmasına sebep olur. Bu şekilde dersin ilk kısmını tamamlamış olduk.

İkinci kısımda ise weather app adlı uygulamanın temel çalışma mantığını anladık. Öncelikli olarak bize coğrafi konuma göre hava durumu bilgisi veren ve coğrafi konuma göre yer bilgisi veren iki farklı api gerekli. api belirli uygulamaların kendi açtıkları bilgileri erişmeyi sağlayan bir arayüzdür. (Application programming interface) Bu bilgiler genelde uygulamaya özgüdür ve halka açık olmaz. Çünkü programın işleyişi tutan verilerdir. Api kullanabilmek için öncelikli olarak kodumuzda `npm i request` yazarak `request` modülünü yüklüyoruz. Bu modül web sunucularına talep göndermeye yararmaktadır. Hava durumu bilgisi için aldığım api `api.weatherstack.com/current?access_key=fe043c506076165831ff246584cedde3&query=37.8267,-122.4233` bu şekildedir. Yapacağımız işlem şu şekilde

```
const request = require("request");

const url =
"http://api.weatherstack.com/current?access_key=fe043c506076165831ff246584cedde3&query=37.8267,-122.4233"

request({url:url,json:true},(error,response) =>{
  const data = JSON.parse(response.body);
  console.log(data.current);
  console.log("hava sicakligi" + response.body.current.temperature
    + "konum:" + response.body.current.feelslike);
})
```

Burada yapılan şey bir api sunucusuna bir talep atıp gelen dönen response yani json formatında bir dönüş alıyoruz ardından response.body.current diyerek api içinde query sorgusunda girilen enlem boylam bilgisine göre mevcut konumun sıcaklık ve hissedilen sıcaklık değerlerini getirmiş olduk.

```
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\WeatherApp> node app.js
hava sicakligi14Hissedilen:12
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\WeatherApp>
```

Aldığımız çıktı bu şekilde

Ardından bu api'ya bağlanıp bağlanamadığımızı sorgulamak adına koşul cümleleri ekledik çünkü bağlantıyı yaptıktan sonra belki bilgileri alamayabiliriz veya mevcut konum olmayabilir bunun kontrolünü sağlamak adına error parametresini de kullanıp koşul içerisine alıyoruz.

```
const request = require("request");

const url =
"http://api.weatherstack.com/current?access_key=2as7c2b083d9ceb97b1c8f7dbe738de81&query=37.8267,-122.4233"

request({url:url,json:true},(error,response) =>{
  /*
  console.log("hava sicakligi" + response.body.current.temperature
  + "Hissedilen:" + response.body.current.feelslike);*/

  if(error){
    console.log("servise baglanamadi");
  } else if(response.body.error){
    console.log("bulunamadi");
  }
})
```

Burada bilerek erişim keyimizde bir değişiklik yaptığımızda sonuç olarak bize bir response.error döndürücek bunu test ettik.

```
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\WeatherApp> node app.js
bulunamadi
```

Eğer api keyimiz doğru olsak ve internetimizi kesmiş olsak bu sefer error bloğuna girip servise bağlanamadı şeklinde çıktı alırdık.

Daha sonra yukarıda aynı yaptığımız işlemleri coğrafi konum veren api içinde denedik. Şu an elimde bu api keyi olmadığı için burada testlerimi paylaşamıyorum ama yaptığımız işlem yine şu şekilde olucak

```
const geocodeUrl = 'test';

request({ url: geocodeUrl, json: true }, (error, response) => {
  const longitude = response.body.features[0].center[0];
  const latitude = response.body.features[0].center[1];
  console.log(" Enlem : " + latitude + "Boylam : " + longitude);
});
```

Burada da aynı şekilde bir talep atıyoruz tek farkı farklı bir api sunucusuna talep atıyoruz. Bu api'nin farkı bize yine nesne olarak döndürmesinin yanı sıra bazı özellikleri array olarak döndürülmüş bunun sebebi ise örnek vermek gerekirse izmir adlı bir yeri sorguladığımızda dünya üzerinde farklı izmir adında konumlar bulunabilir bu yüzden aynı isimli şehirleri bir arrayde tutuyor. Buradan center konumu verir birisi meridyen diğeri ise paralel. Eğer bu iki api aynı anda kullanırsak coğrafi konumda aldığımız örneğin bursa bunun koordinatlarını bir değişkende saklayıp ardından weather api'de kaydettiğimiz değişkenleri talep ettiğimiz url içerisine gönderirsek istediğimiz her konumun hava durumu bilgisine ulaşabiliriz.