

19360859053

Hümeýra ÇİMEN

## Hava Durumu Projesi-2

İlgili url de bozukluk oluşturarak hata kontrolu yapılır .

```
// const  
geocodeUrl="https://api.mapbox.com/geocoding/v5/mapbox.places/bursa.json?proximity=ip&access_token=pk.eyJ1IjoiYmV5cmNtbiIsImEiOiJjbHR6aW00enUwMGRuMmp0NGNyczVseGxuIn0.qgdxfBjW5ccrcv4JUdlyJw"
```

```
const  
geocodeUrl=https://api.mapbox.com/geocoding/v5/mapbox.places/ğğğ.json?proximity=ip&access\_token=pk.eyJ1IjoiYmV5cmNtbiIsImEiOiJjbHR6aW00enUwMGRuMmp0NGNyczVseGxuIn0.qgdxfBjW5ccrcv4JUdlyJw
```

```
hissedilen sıcaklık: 283  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js  
Enlem: 40.919469  
Boylam: 29.165614  
null  
hava sıcaklığı : 284  
hissedilen sıcaklık: 283  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js  
Mapbox API'sine bağlantı kurulamadı!  
null  
hava sıcaklığı : 284  
hissedilen sıcaklık: 283  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5>
```

Wifi connection kesildiğindeki hata kontrolu

```
hissedilen sıcaklık: 283  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js  
Error: getaddrinfo ENOTFOUND api.weatherstack.com  
    at GetAddrInfoReqWrap.onlookupall [as oncomplete] (node:dns:118:26) {  
  errno: -3008,  
  code: 'ENOTFOUND',  
  syscall: 'getaddrinfo',  
  hostname: 'api.weatherstack.com'  
}  
hava durumu servisine bağlantı kurulamadı !  
Mapbox API'sine bağlantı kurulamadı!  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5>
```

Başarılı bağlantı olduğunda başarılı çıktı elde edilir.

```
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js
Enlem: 40.182766
Boylam: 29.06773
null
hava sıcaklığı : 284
hissedilen sıcaklık: 283
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> |
```

NOT: callback fonksiyon callback fonksiyonların hepsi asenkron değildir, senkron da olabilir .

Burada **asekron** örneğine ait kullanım mevcuttur

```
setTimeout(() => {
    console.log("2 saniye bekleme time out")
}, 2000);
```

Burada **senkron** örneğine ait kullanım mevcuttur

```
// senkron callback fonksiyon
const names=["ali","veli","emre"];
// const shortNames=names.filter(name=> name.length<4) // filter ile kısa isimleri bulduk
// console.log(shortNames)

const shortNames=names.filter(name=> {
    return name.length<4
}) // filter ile kısa isimleri bulduk

console.log(shortNames)
```

Enlem ve boylam bilgilerinin **senkron** olarak getirilmesi:

```
18  const geocode =(address,callback) =>{
19      const data ={
20          latitude:0,
21          longitude:0
22      }
23      return data
24  }
25
26  const data=geocode("bursa")
27  console.log(data)
28  }
```

```
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node callbackTest.js
{ latitude: 0, longitude: 0 }
```

Enlem ve boylam bilgilerinin **asekron** olarak getirilmesi:

```
// Asenkron bir veri getirme işlemi
const geocode = (address, callback) => {
  setTimeout(() => {
    const data = {
      latitude: 0,
      longitude: 0,
    }
    callback(data);
  }, 2000);
};

geocode("bursa", (data) => {
  console.log(data);
});

PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node callbackTest.js
{ latitude: 0, longitude: 0 }
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> 
```

asekron toplama işlemi fonksiyonu

```
45 //asekron işlem için setTime kullanıyoruz asekron olan herhangi bir fonksiyon kullanılabilir
46 const add = (a, b, callback) => {
47   // Simulate asynchronous behavior using setTimeout
48   setTimeout(() => {
49     // Calculate the sum
50     const sum = a + b;
51     // Call the callback function with the calculated sum
52     callback(sum);
53   }, 2000);
54 };
55
56 // 1+4 degeri asekron toplamaişlemine gönderilir sonuç sum a atanır ve arrow fonksiyon için de sum yazdırılır
57 add(1, 4, (sum) => {
58   console.log(sum); // Log the result after 2 seconds
59 });

PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node callbackTest.js
5
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> 
```

// 1+4 degeri asekron toplamaişlemine gönderilir sonuç sum a atanır ve arrow fonksiyon için de sum yazdırılır.

## app.js içindeki kodların asekron dönüşümü

### geocode asekron dönüşümü

.js ilgili dosyaları require edelim .

konsoldan girilen konum bilgisine göre sorgu oluşturularak asekron bilgilerin getirilmesi için konum bilgisinin girildiğinden emin olalım .

```
You, 4 days ago | 1 author (You) | 💡 Click here to ask Blackbox to help you code faster
1  const request = require("postman-request");
2  const geocode = require("./utils/geocode");
3  const forecast = require("./utils/forecast");
4
5  const address = process.argv[2];
6  if (!address) {
7    console.log("adres bilgisini parametre olarak veriniz ");
8  }
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js
adres bilgisini parametre olarak veriniz
hata {
  latitude: 46.2823441,
  longitude: -63.104385199999996,
  location: 'Undefined, Charlottetown, Prince Edward Island C1C 1E9, Canada'
}
hata Undefined, Charlottetown, Prince Edward Island C1C 1E9, Canada
hata Girilen konum bilgisi bulunamadı
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js BURSA
hata {
  latitude: 40.182766,
  longitude: 29.06773,
  location: 'Bursa, Bursa, Türkiye'
}
hata Bursa, Bursa, Türkiye
hata { temperature: 22, feelslike: 24 }
hata Bursa, Bursa, Türkiye
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5>
```

### geocode.js içeriği

```
const request=require('postman-request')
//verilen adres bilgisine göre enlem ve boylam bilgisinin asekron fonk
olarak getirilmesi
const getGeocode = (address, callback) => {
  const url =
    "https://api.mapbox.com/geocoding/v5/mapbox.places/" +
    encodeURIComponent(address) +
    ".json?proximity=ip&access_token=pk.eyJ1IjoiaGl5cmNtbiIsImEiOiJjbHR6aW00enUwMGRuMmp0NGNYczVseGxuIn0.qgdxfBjW5ccrcv4JUd1yJw";

  setTimeout(() => {
    request({ url: url, json: true }, (error, response) => {
      if (error) {
```

```

        callback("Mapbox API'sine bağlantı kurulamadı!", undefined);
    } else if (response.body.features.length === 0) {
        callback("Belirtilen konum bilgisi bulunamadı!", undefined);
    } else {
        const longitude = response.body.features[0].center[0];
        const latitude = response.body.features[0].center[1];
        const location=response.body.features[0].place_name;
        // error burada ilk basta konumlanır
        callback(undefined, { latitude, longitude ,location});
    }
    });
}, 2000);
};

```

```
module.exports=getGeocode
```

**encodeURIComponent** adres içindeki değişik fromattaki verilere dair önlem için kullanıldı .  
 Url deki konum içeriğine temizlenen konum bilgisi yerleştirildi ve asekron fonksiyon  
 içerisinde önceki işlemler taşındı.

Enlem boylam konum bilgisi dönüldürüldü ve forecast içeriğine gönderildi.

### forecast.js içeriği

Gönderilen enlem boylam bilgisine göre sorgu oluşturuldu ve sonuç asekron fonksiyon  
 içeriğinde yazdırıldı.

```

const request = require('postman-request')

const forecast = (longitude, latitude, callback) => {
    const url =
"http://api.weatherstack.com/current?access_key=77ce69c14f63baae1cc568a54e347187&query=" + encodeURIComponent(longitude) + "," +
encodeURIComponent(latitude)

    request({ url: url, json: true }, (error, response) => {

        if (error) {
            callback("Hava durumu servisine bağlantı kurulamadı.",
undefined);
        } else if (response.body.error) {
            callback("Girilen konum bilgisi bulunamadı", undefined);
        } else {
            const temperature = response.body.current.temperature;
            const feelslike = response.body.current.feelslike;
            callback(undefined, { temperature, feelslike });
        }
    })
};

module.exports = forecast;

```

Sonuç: İzmir ve İstanbul için oluşturulan sorgularda başarılı bir şekilde çıktılar elde edilmiştir.

```
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js izmir
hata {
  latitude: 38.414733,
  longitude: 27.143412,
  location: 'Izmir, İzmir, Türkiye'
}
hata Izmir, İzmir, Türkiye
hata { temperature: 22, feelslike: 24 }
hata Izmir, İzmir, Türkiye
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> node app.js istanbul
hata {
  latitude: 41.0091982,
  longitude: 28.9662187,
  location: 'Istanbul, Türkiye'
}
hata Istanbul, Türkiye
hata { temperature: 23, feelslike: 24 }
hata Istanbul, Türkiye
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week5> 
```

HümeYra ÇİMEN

*Uygulama dosyalarına ilgili linkden ulaşabilirsiniz week5 app.js dosyası içinde .*

<https://github.com/hmyrcmn/NodeJs/tree/main/week5>