6.Hafta

Furkan Kazım Çam 21360859036

Callback Fonksiyonu

Bir fonksiyonun diğer bir fonksiyonu parametre olarak alıp çağırmasına callback fonksiyonu denir. İki tipi vardır senkron ve asenkron olmak üzere.

Asenkron callback fonksiyonu örneği

```
setTimeout( () => {
console.log('Two seconds time out')
}, 2000)
```

Senkron callback fonksiyonu örneği

```
const names = ['Ali', 'Berk', 'Cengiz']
const shortNames = names.filter( (name) => {
return name.length <= 4
})
```

Önceki haftaki konumuza geri dönelim. Konuyla ilişkili bir callback fonksiyonu yazmaya çalışalım.

Fonksiyonun adına gecode verin.

```
const geocode = (address, callback) => {
const data = {
latitude: 0,
longitude: 0
}
}
```

Şu anda bu asenkrondur ve verimliliği düşüktür. Bunu senkron hale getirelim.

```
const geocode = (address, callback) => {
const data = {
latitude: 0,
longitude: 0
}
return data
}
Const data = geocode('Bursa')
console.log(data)
```

senkron ve asenkron fonksiyonlarının çıktılarının aynı olduğunu göreceksiniz.

Şimdi bu sorgulara gecikme ekleyelim.

```
const geocode = (address, callback) => {
setTimeout( () => {
const data = {
latitude: 0,
longitude: 0
}
return data
}, 2000)
}
```

Undefined çıktısını alıcaksınız. Çıktının böyle olmasının sebebi geocode fonksiyonun herhangi bir değer döndürmemesidir. Bu js'e özel bir durumdur diyebiliriz. Kodu callback fonksiyonu hale getirirsek problemi ortadan kaldırmış oluruz.

```
const geocode = (address, callback) => {
setTimeout( () => {
const data = {
latitude: 0,
```

```
longitude: 0
}
callback(data)
}, 2000)
}
geocode('Bursa', (data) => {
console.log(data)
})
```

2 saniye gecikmenin ardından istediğimiz çıktıyı almamız gerekmektedir.

Yazdığımız kodlları yorum satırı haline getirelim. Farklı bir örnek ile devam edelim.

```
add(1, 4, (sum) => {
console.log(sum)
})
const add = (a, b, callback) => {
setTimeout( () => {
callback(a + b)
}, 2000)
}
```

Çalıştırdığınızda 2 saniye gecikme sonrası 5 cevabını almalısınız.

Callback fonksiyonunu öğrendiğimize göre önceki hafta ki kodumuza geri dönüp uygulamaya başlayalım.

Geocode'a şehir girdisini verip çıktı olarak enlem ve boylam almaya çalışacağız.

Utils adında bir klasör oluşturalım. Bu işlemin amacı örnek olarak 5 farklı şehrin enlem ve boylam bilgisini elde etmek istiyorum tek tek url değiştirmek zor ve gereksiz.

Utils klasöründe geocode.js adlı bir dosya açıp içine aşağıdaki kodları yazın.

```
const geocode = (address, callback) => {

const url = ‘https://api.mapbox.com/geocoding/v5/mapbox.places/' +
encodeURIComponent(address)

+

‘.json?access_token=pk.eyJ1IjoiaWZzNSIsImEiOiJja215YnU1NHIwMmU4MnVvMnl0dnl2
YndjIn0.iIitdw0flCDP

jlnjP0Kl7g&limit=1' ’

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

if (error) {

callback(‘Unable to connect to location services’, undefined)

} else if (response.body.length === 0) {

callback(‘Unable to find location. Try another search.’, undefined)

} else {

callback(undefined, {

latitude: response.body.features[0].center[0]

longitude: response.body.features[0].center[1]

location: response.body.features[0].place_name

})

}

})

}
```

Fonksiyonun bu kısmını app.js’in içinden silmemiz gerekir.

Kodumuza postmanı require edip geocode fonksiyonunu ise export etmemi gerekir. Bunları ekleyip bir de fonksiyonu çağırdığımız kısmı da ekleyelim.

```
geocode(‘Bursa’, (error, data) => {

console.log(‘Error’, error)

console.log(‘Data’, data)

})
```

App.js’e dönüp gecocode.js dosyasını require edelim.

const geocode = require(‘./utils/geocode’)

Çalıştırırsak çıktılarımızın doğru olduğunu görebiliriz.

App.jsteki geocode ve forecast fonksiyonlarını silin.

Hava durumu bilgileri için weatherstack apisi kullanıyorduk şimdi utils klasörü içinde forecast dosyası oluşturup enlem ve boylam bilgisi ile hava durumu bilgilerini el etmeye çalışalım. Bu forecast dosyasını app.js içinde require etmeyi unutmayalım.

const forecast = require(‘utils/forecast’)

forecast dosyasını ise aşağıdaki gibi doldurmamız gerekmektedir.

const request = require(‘request’)

const forecast = (latitude, longitude, callback) => {

conts
url=‘http://api.weatherstack.com/current?access_key=81c6957beda8a6484399ecabcfdd7eae&query=' + latitude

+ ’,’ + longitude + ‘&units=f’

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

if (error) {

callback(‘Unable to connect to weather service’, undefined)

}

else if (response.body.error) {

callback(‘Unable to find location’, undefined)

} else {

callback(undefined, ‘Hava şu anda: ' + response.body.current.weather_descriptions[0] + '

Hava sıcaklığı şu anda: ' + response.body.current.temperature + ' derece ve hissedilen sıcaklık: ' +

response.body.current.feelslike + ' derece')

}

geocode(‘Bursa’, (error, data) => {

console.log(‘Error’, error)

console.log(‘Data’, data)

forecast(-75.7088, 44.1545, (error, data) => {

console.log(‘Error’, error)

```
console.log('Data', data)
}
})
}
module.exports = forecast
```

kodu dinamikleştirmek için forecast fonksiyona verdiğimiz enlem boylam bilgisini api den gelen bilgiyi verelim.

```
geocode('Bursa', (error, data) => {
console.log('Error', error)
console.log('Data', data)
forecast(data.latitude, data.longitude, (error, data) => {
console.log('Error', error)
console.log('Data', data)
})
})
```

Çalıştırdığımızda eğer error alırsak program kapanmasın diye if else blokları eklemeliyiz.

```
geocode('Bursa', (error, data) => {
if (error) {
return console.log(error) // if an error, it will not continue to run.
}
forecast(data.latitude, data.longitude, (error, forecastData) => {
if (error) {
return console.log(error)
}
console.log(data.location)
console.log(forecastData)
```

```
})
})
```

Kullanıcıdan alınan girdiye göre bilgileri çıktılamak istersek ise yargs kullanarak yapabiliriz.

```
const address = process.argv[2]
geocode(address, (error, data) => {
if (error) {
return console.log(error) // if an error, it will not continue to run.
}
forecast(data.latitude, data.longitude, (error, data) => {
if (error) {
return console.log(error)
}
console.log('Data', data)
})
})
```

Node app.js Boston yazarak çalıştırabilirsiniz. Boston yerine istediğiniz şehri yazarak istediğiniz şehrin hava durumu bilgilerini yazdırabilirsiniz.