Ödev 3

Ahmet Yusuf Birdir

21360859026

JSON VERİ EKLEME VE SİLME

- Bugün sizlere Node.js ile komut satırında nasıl not ekleyip silebileceğimizi anlatacağım. Bunun için yargs kütüphanesini kullanacağız. Haydi kodu birlikte inceleyelim!
- Öncelikle 2 adet dosya oluşturun. Bu örnekte app.js ve notes.js kullanılacak.
- Daha sonra app.js dosyasını açın ve yargs, ve notes.js dosyalarını dahil edin.

```
const yargs = require("yargs")
const notes = require("./notes.js")
```

JSON VERİ EKLEME VE SİLME

- Şimdi ise note ekleme ve silme komutlarını ekleyelim.
- Daha sonra ise yargs.argv'yi çağıralım.

```
yargs.command({
    command: 'ekle',
    describe: 'yeni not ekler',
    builder:{
        title:{
            describe: 'Not Ekle',
            demandOption: true,
            type: 'string'
        },
        body:{
            describe: 'gövde',
            demandOption: true,
            type: 'string'
        }
    },
    handler: function(argv){
        notes.addNote(argv.title, argv.body)
    }
})
```

```
yargs.command({
    command: 'delete',
    describe: 'notları siler',
    builder:{
        title:{
            describe: 'Not Sil',
            demandOption: true,
            type: 'string'
        },
    },
    handler: function(argv){
        notes.deleteNote(argv.title)

    }
})

console.log(yargs.argv)
```

JSON VERİ EKLEME VE SİLME

- Sırada notes.js dosyası var. İlk olarak dosyaya dosya işelmleri için fs ve renklendirmek için chalk kütüphanelerini ekleyin.(Chalk için versiyon 4 kullanılmıştır.)

```
const fs = require("fs")
const chalk = require("chalk")
```

JSON VERİ EKLEME VE SİLME

- Şimdi ise ekleme ve silme işlemleri için kullanılacak noteları çağırma fonksiyonunu ekleyin. Bu fonksiyon notes.json dosyasındaki note'ları döndürecektir.

```
const loadNotes = function(){
    try{
        const dataBuffer = fs.readFileSync('notes.json')
        const dataJSON = dataBuffer.toString()
        return JSON.parse(dataJSON)
    }
    catch(e){
        return []
    }
}
```

JSON VERİ EKLEME VE SİLME

- Şimdi ise eklenecek notları notes.json dosyasına yazacak olan fonksiyonu ekleyin.

```
const saveNotes = function(notes){
    fs.writeFileSync('notes.json',JSON.stringify(notes))
}
```

JSON VERİ EKLEME VE SİLME

- Şimdi ise ekleme fonksiyonunu yazın. Bu fonksiyon notes.json dosyasındaki note'ları alır ve yeni eklenecek olan note ile başlıkları benzer olan var mı kontrol eder. Eğer benzerlik yok ise note'u ekler, benzerlik var ise hata mesajı döndürür.

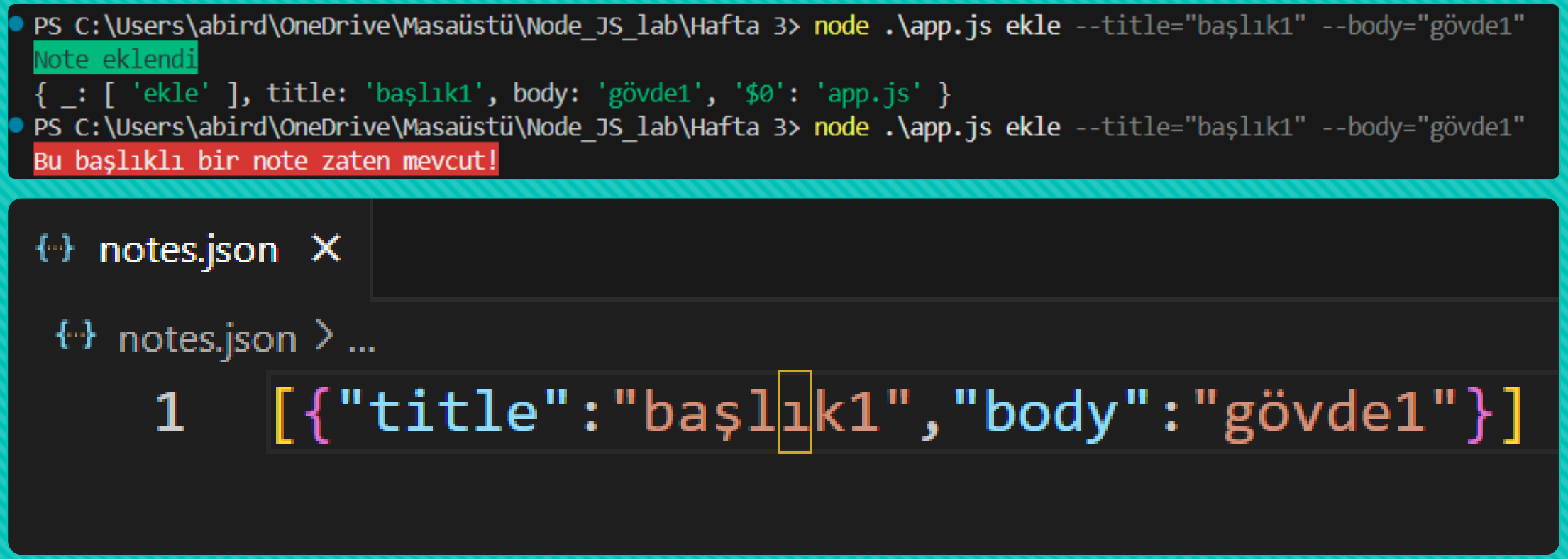
```
const addNote = function(title,body){
    const notes = loadNotes()
    const duplicates = notes.filter(function(notes){
        return notes.title === title
    })
    if(duplicates.length === 0){
        notes.push({
            title:title,
            body:body
        })
        saveNotes(notes)
        console.log("Notlar Eklendi")
    }else{
        console.log("This note title is already exist!")
    }
}
```

JSON VERİ EKLEME VE SİLME

- Son olarak silme fonksiyonunu ekleyin. Bu fonksiyon notes.json dosyasındaki note'ları alır ve silinecek note'un başlığı ile aynı olan var mı kontrol eder. Var ise siler ve yeşil arka planlı başarı mesajı döndürür. Yok ise kırmızı arka planlı hata mesajı döndürür.
- En sonda ise bu fonksiyonları dışa aktarıyoruz ki app.js dosyası ile çağırabilelim.

```
const deleteNote = function(title){
    const notes = loadNotes()
    const filteredNotes = notes.filter(function(notes){
        return notes.title !== title
    })
    if(filteredNotes.length === notes.length){
        console.log(chalk.bgRed("note does not exist"))
    }else{
        console.log(chalk.bgGreen("note is deleted"))
        saveNotes(filteredNotes)
    }
}

module.exports = {
    addNote : addNote,
    deleteNote: deleteNote
}
```



JSON VERİ EKLEME VE SİLME ÖRNEK

```
PS C:\Users\abird\OneDrive\Masaüstü\Node_JS_lab\Hafta 3> node .\app.js delete --title="başlık2"
Böyle bir note yok
{ _: [ 'delete' ], title: 'başlık2', '$0': 'app.js' }
PS C:\Users\abird\OneDrive\Masaüstü\Node_JS_lab\Hafta 3> node .\app.js delete --title="başlık1"
note silindi
{ _: [ 'delete' ], title: 'başlık1', '$0': 'app.js' }
```

notes.json

notes.json

```
1  []
```

JSON VERİ EKLEME VE SİLME ÖRNEK

FONKSİONLAR

- Burada 3 Çeşit fonksiyon tanımlaması yapacağız ve farklarını inceleyeceğiz.
- Geleneksel fonksiyon tanımlama, Ok fonksiyonu uzun versiyonu ve ok fonksiyonu kısa versiyon
- Temel bir fark 'this' Bağlamı :
- Geleneksel fonksiyonlarda, this anahtar kelimesi fonksiyonun çağrıldığı bağlama göre değişir.
- Ok fonksiyonları ise lexical this (söz dizimsel bağlam) kullanır, yani this değeri fonksiyonun tanımlandığı çevreden gelir, çağrıldığı yerden değil.

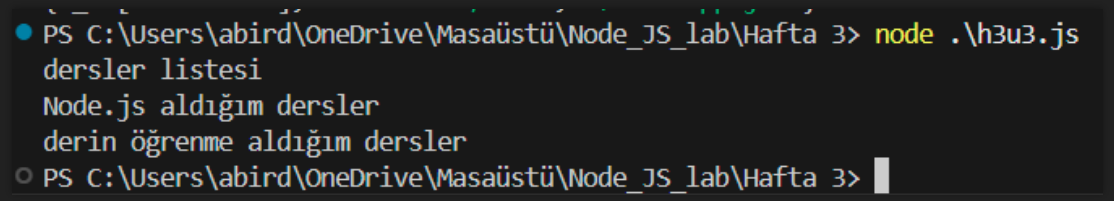
```
const kareAl = function(x){
    return x*x
}
const kareAl2 = (x) => {
    return x*x
}
const kareAl3 = (x) => x*x
```

ÖRNEK

```
const eventDers = {
    name: 'dersler',
    dersListesi : ["Node.js", "derin öğrenme"],
    dersleriYazdir: function(){
        console.log(this.name + " listesi")
        this.dersListesi.forEach((ders) => {
            console.log(ders + " aldığım " + this.name)
        })
    }
} //* arrow fonksiyonu normal fonksiyonların aksine parent'ın kullandığı this objelerine erişebilir.
eventDers.dersleriYazdir()
```

ÖRNEK

- Burada ok fonksiyonu yerine geleneksel fonksiyon tanımlaması yapılsaydı this.name kısmı geldiği yerde undefined hatası alınacaktı.
- Ancak örnekte doğru bir tanımlama olduğundan bu hata ile karşılaşılmadı.



SON