

NODE-JS DERSİ HAFTA -2 Ders Notları

— Komuttadiki Argümana Erişim

— Yargs Modulunun kullanımı ve Not Uygulaması için komutların işlenmesi

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>node app.js hümeýra
teori dersi hafta -2
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  'hümeýra'
]
```

komut çıktılarının incelenmesi

1 çalıştırılabilir dosyanın konum bilgisi .,

2 çalıştırılmak istenilen dosyanın yolu

3 gönderilen parametreler // node app.js hümeýra çimen gibi . Parameter burada hümeýra çimen "

Komuta parametre girişinin yapılması

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>node app.js list --title="merhabaaa"
teori dersi hafta -2
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  'list',
  '--title=merhabaaa'
]
not listeleniyor !!!
```

artık title ile parametre sayısı arttırıldı ancak bu title analiz edilmeli bunun için yargs modulu kullanılır.

Yargs modülünün install edilmesi

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>npm install yargs
```

```
added 16 packages, and audited 17 packages in 2s
```

```
2 packages are looking for funding
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>|
```

```
JS app.js U {} package-lock.json U X
week2 > {} package-lock.json > {} packages > {} "" > {} dependencies > yargs
```

💡 Click here to ask Blackbox to help you code faster

```
1  {
2    "name": "week2",
3    "version": "1.0.0",
4    "lockfileVersion": 3,
5    "requires": true,
6    "packages": {
7      "": {
8        "name": "week2",
9        "version": "1.0.0",
10       "license": "ISC",
11       "dependencies": {
12         "yargs": "^17.7.2"
13       }
14     }
15   }
```

install edilen yargs modulu default son sürüm olarak eklendi sürüm 17.7.2

yargs modul kullanımı

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>node app.js list --title="merhabaaa"
teori dersi hafta -2
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  'list',
  '--title=merhabaaa'
]
not listeleniyor !!!
{ _: [ 'list' ], title: 'merhabaaa', '$0': 'app.js' }
```

burada yargs modulu kullanımında yukarıdakinden biraz farklı ve title in title olduğu belirlendi yanı yazdırmak istesek artık bu title' a erişebilir ve yazdırabiliriz.

```
23 // import yargs module
24 const yargs= require("yargs");
25
26 // yargs argumanlarının incelenmesi
27 console.log(yargs.argv)
```

yargs ile not uygulamasına ait komutların oluşturulması .

```
yargs.command({
  command: 'add',
  describe: "not ekler",
  handler: function() {
    console.log('Bir not eklendi');
  }
})
yargs.command({
  command: 'remove',
  describe: "not siler",
  handler: function() {
    console.log('Bir not silindi');
  }
})
yargs.command({
  command: 'list',
  describe: "notları listeler",
  handler: function() {
    console.log("notlar listelendi")
  }
})

yargs.command({
  command: 'read',
  describe: "seçilen notu gösterir",
  handler: function () {
    console.log("Seçili not okunuyor..");
  }
})
```

node `help` piçeriğinin düzenlenmesi

default help parametrelili komutun çıktısının görüntüsü bu şekilde.

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>node app.js --help
teori dersi hafta -2
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  '--help'
]
"--help" isimli komut bulunamadi
Options:
  --help      Show help                                [boolean]
  --version   Show version number                       [boolean]
```

version ile özelleştirilmeler yapıldı.

```
23 // import yargs module
24 const yargs= require("yargs");
25 yargs.version="1.0.0"
26
```

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>node app.js --version
teori dersi hafta -2
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  '--version'
]
"--version" isimli komut bulunamadi
1.0.0

C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>node app.js add --title="hümeýra çimen"
teori dersi hafta -2
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  'add',
  '--title=hümeýra çimen'
]
yeni not ekleniyor!!!
{ _: [ 'add' ], title: 'hümeýra çimen', '$0': 'app.js' }
undefined
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>
```

Not : title ataması yapılmadığında missing value olarak değerlendirilir title içeriği oluşturulmasa dahi tanımlanmış varsayılır ,sorunsuz bir title girişi için aşağıdaki kontrollerin yapılması gerekmektedir.

Aşağıdaki gibi title parametresi incelenir .

describe ile title in tanımlaması yapılır,

demandOption ile komutta title in kullanılmasını zorunlu kılar. Mutlaka title parametresi olmalı .

Not: Title ile parametrenin tanımlanarak herhangi bir tanımlama yapılmaması durumunda bu seçenek hata vermez ve boolean olarak değerlendirerek bu değeri true olarak varsayar bunu önlemek için type belirtmeliyiz string olarak tip belirtmesi yapıldığında title'a tanımlama yapmak zorunlu hale gelir.

```
// yargs argumanlarının incelenmesi
yargs.command({
  command: 'add',
  // --help ile gözükmesini istediğimiz açıklamanın tanımlanması
  describe: "yeni not ekler",
  builder: {
    title: {
      describe: 'not başligi belirlenir.',
      demandOption: true
    }
  },
  // komut girildiğinde ne yapılacağını belirlenmesi
  handler: function() {
    console.log("Yeni bir not eklendi");
  }
})
```

title parametre atamadan sadece title yazıldığında default true döndürülür. bu ifade boolean ile bu açığın onune geçmek için type belirtilmesi gerekir.

yargs ile komuttadi handler içeriğinin düzenlemesi

```
yargs.command({
  command: 'add',
  // --help ile gözükmesini istediğimiz açıklamanın tanımlanması
  describe: "yeni not ekler",
  builder: {
    title: {
      describe: 'not başligi belirlenir.',
      demandOption: true,
      type: "string"
    },
    body: {
      describe: 'not içeriği belirlenir.',
      demandOption: true,
      type: 'string'
    }
  },
  // komut girildiğinde ne yapılacağını belirlenmesi
  handler: function (argv) {
    console.log("başlık:", argv.title);
    console.log("title içeriği:", argv.body);
  }
});
```

```
Node.js v20.11.1
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> node app.js add --title="humeyra çimen" --body="bilgisayar mühendisi"
başlık: humeyra çimen
title içeriği: bilgisayar mühendisi
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>
```

console.log(...)

`yargs.parse()` fonksiyonu, `yargs` modülü ile tanımlanan komut satırı argümanlarını işlemek ve analiz etmek için kullanılır. Genellikle bir `yargs` nesnesi oluşturulduktan sonra, `parse()` fonksiyonu kullanılarak komut satırı argümanları işlenir ve `yargs` nesnesinin içindeki verilere erişim sağlanır.

```
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> node app.js add --title="humeyra"
teori dersi hafta -2
yeni not ekleniyor!!!
{ _: [ 'add' ], title: 'humeyra', '$0': 'app.js' }
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2>
```

`process.argv` Node.js ortamında bulunan özel bir dizi (array)'dir. Bu dizi, bir Node.js uygulamasının başlatılması sırasında komut satırına yazılan argümanları içerir.

```
yeni not ekleniyor!!!
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\HÜMEYRA\\Documents\\GitHub\\NodeJs\\week2\\app.js',
  'add',
  '--title=humeyra'
]
```

`yargs.argv` ifadesi, `yargs` kütüphanesinin sunduğu bir özelliktir ve `yargs` nesnesi üzerinden komut satırı argümanlarını işlemek için kullanılır. Node uygulaması için girilen komutların `yargs` ile oluşturulmasında kullanılır.

Yani, `console.log(yargs.argv)`'yi kaldırıp `yargs.parse()` ekleyerek, komut satırı argümanlarının açıkça ayrıştırılmasını sağlarsınız ve programınızın akışı üzerinde daha fazla kontrol elde edersiniz.

NODE-JS ile JSON İşleme

1 . Book adında bir obje oluşturuldu .

```
const book={
  title:"kitap başlığı",
  author:"kitap yazarı"
}
```

2. Book objesi bir json formatına `stringify` ile dönüştürüldü, sonucunda ise string formatında bir dönüş alındı.

```
8 // Bu adımda, JSON.stringify yöntemi kullanılarak book nesnesini JSON formatına çeviriyoruz.
9
10 const bookJSON= JSON.stringify(book)//json nesnesini stringe çevirmek için kullanılan metoddur.
11 console.log(typeof bookJSON)//string olarak geliyor
12 console.log(bookJSON)// {"title":"kitap başlığım","author":"kitap yazarı"} bu şekilde geliyor
13
dosya oluşturuldu başlık: undefined yazar: undefined
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> node dene-json.js
string
{"title":"kitap başlığım","author":"kitap yazarı"}
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> 
```

Burada görüntülenen title ve author sadece string olarak mevcuttur erişim yapılamaz deneyelim . *undefined*

```
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> node dene-json.js
string
{"title":"kitap başlığım","author":"kitap yazarı"}
title gelecek mi bakalım undefined
yazar gelecekmi bakalım: undefined
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> 
```

3. string ifade bir json için dosyaya yazılır.

```
`const fs = require('fs');
fs.writeFileSync('1-json.json', bookJSON);`
```

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

```
const book = {
  title: "kitap başlığım",
  author: "kitap yazarı"
};
```

```
// JSON.stringify yöntemi kullanılarak book nesnesini JSON formatına
çeviriyoruz.
const bookJSON = JSON.stringify(book);
```

```
console.log(typeof bookJSON); // "string" olarak gelir
console.log(bookJSON); // {"title":"kitap başlığım","author":"kitap
yazarı"}
```

```
// Ancak, bookJSON bir string olduğu için title ve author özelliklerine
doğrudan erişim sağlanamaz.
// Doğru kullanım, JSON verisini bir dosyaya yazmak olacaktır.
fs.writeFileSync("deneme.json", bookJSON);
console.log("Dosya oluşturuldu.");
```

```
// Dosyadan okuma yapılır ve buffer'a alınır.
const dataBuffer = fs.readFileSync('deneme.json');
```

```
// Buffer tipindeki veriyi bir stringe dönüştürürüz.
const dataJSON = dataBuffer.toString();
```

```
// String ifadesini JSON veri nesnesine dönüştürürüz.  
const data = JSON.parse(dataJSON);
```

```
console.log("Çekilen title:", data.title); // "kitap başlığım" olarak gelir  
console.log("Çekilen author:", data.author); // "kitap yazarı" olarak gelir
```

```
Node.js v20.11.1  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> node dene-json.js  
string  
{ "title": "kitap başlığım", "author": "kitap yazarı" }  
Dosya oluşturuldu.  
Çekilen title: kitap başlığım  
Çekilen author: kitap yazarı  
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week2> █
```

HÜMEYRA ÇİMEN

İlgili uygulama dosyalarına adresinden ulaşabilirsiniz:

<https://github.com/hmyrcmn/NodeJs/tree/main/week2>