

## NodeJS ile Web Programlama Hafta-4 Rapor

Bu haftaki derste şu kodu yazdık:

```
const tasks = {  
  tasks : [  
    {  
      text : "Alışveriş",  
      completed : true  
    },  
    {  
      text : "Temizlik",  
      completed : false  
    },  
    {  
      text : "Ödev",  
      completed : false  
    }  
  ]  
}  
  
console.log(tasks.getTasksToDo())
```

Amacımız tamamlanmayan görevleri belirlemek.

Görevler adı altında görevlere isim vererek ve boolean bir değer ile olup olmadığını (tamamlayıp tamamlamadığını) girdik ve bu şekilde görevlerin olmama durumunda çekme sağlayacağız.

Filter fonksiyonu ile filtreleme yaptık.

```
const tasks = {
  tasks : [
    {
      text : "Alışveriş",
      completed : true
    },
    {
      text : "Temizlik",
      completed : false
    },
    {
      text : "Ödev",
      completed : false
    }
  ],
  getTasksToDo : function () {
    this.tasks.filter((tasks) => {
      return tasks.completed === false
    })
    return this.getTasksToDo
  }
}
```

Görevin completed değeri false olma durumunda return olarak döndüren bir filter komutu yazdık.

Kullanımda da arrow mantığı ile ok kullanarak yaptık.

Return diyerek de toplanan elemanları geri döndürdük.

Şu şekilde de bir çıktı aldık :

```
Node.js v20.11.0
● PS C:\Users\bilmuh\Desktop\ada> node arrow_test.js
[Function: getTasksToDo]
● PS C:\Users\bilmuh\Desktop\ada> node arrow_test.js
[
  { text: 'Temizlik', completed: false },
  { text: 'Ödev', completed: false }
]
○ PS C:\Users\bilmuh\Desktop\ada> █
```

Başka bir versiyonda yapmak gerekirse :

```
getTasksToDo : function (){
    return this.tasks.filter((tasks) => {
        return tasks.completed === false
    })
}
```

Yine aynı çıktıyı alırız :

```
]
● PS C:\Users\bilmuh\Desktop\ada> node arrow_test.js
[
  { text: 'Temizlik', completed: false },
  { text: 'Ödev', completed: false }
]
PS C:\Users\bilmuh\Desktop\ada> █
```

Fonksiyonu tek satır haline getirmek içinse şu şekilde yapabiliriz :

```
getTasksToDo : function (){
    return this.tasks.filter((tasks) =>
tasks.completed === false) }
```

Yine aynı çıktıyı alır:

```
PS C:\Users\bilmuh\Desktop\ada> node arrow_test.js
[
  • { text: 'Temizlik', completed: false },
    { text: 'Ödev', completed: false }
]
PS C:\Users\bilmuh\Desktop\ada>
```

Geçen hafta yazılan dosyalara geri döndük.

app.js ve notes.js dosyasına baktık.

Tüm handler komutunda parantezler ve functionları kaldırıp uyarılama yaptık. Add komutu üzerinden bakmak gerekirse:

```
handler (argv) {
  notes.addNote(argv.title, argv.body);
},
```

Remove, read ve list için de uyarılama yaptık ve hala çalışır durumunda olup olmadığını kontrol ettik. Sonuç olarak çıktımız da bu şekilde :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
yeni not eklendi
PS C:\Users\bilmuh\Desktop\ada> node app.js add --title="fba" --body="mlk"
yeni not eklendi
PS C:\Users\bilmuh\Desktop\ada> node app.js remove --title="fba"
• Note remove
• PS C:\Users\bilmuh\Desktop\ada> node app.js remove --title="fba"
• Silmek istediğiniz not bulunamamıştır.
• PS C:\Users\bilmuh\Desktop\ada> node app.js add --title="ba" --body="mlk"
• Bu başlık daha önce kullanıldı. Not eklenemiyor!!!
PS C:\Users\bilmuh\Desktop\ada>
```

Şimdi de notes.js dosyasına baktık, değiştirilmesi gerekenleri değiştirdik.

Değişime addNote üzerinden başladık.

```
const duplicateNotes = notes.filter((note) => note.title === title);
```

Kodumuzu arrow formda tek satır hale getirdik.

Aynı işlemi remove için de yaptık. O da şu şekilde oldu.

```
const notesToKeep = notes.filter((note) => note.title !== title);
```

Ardından da loadNotes için :

```
const loadNotes = () => {  
  try {  
    //dosya olmama durumunda hata almamak  
    // için try catch bloguna alırız  
    // not bilgisi getirecek  
    // eski notlara bakıp ona göre ekleme  
    // yapıcak  
    const dataBuffer =  
    fs.readFileSync("notes.json"); //dosyadan  
    //okuma işlemi  
    const dataJSON = dataBuffer.toString();  
    //okudugu veriyi json formatına çeviriyor  
    //parse the string and return  
    return JSON.parse(dataJSON); //JSon  
    //formatına donusur  
  } catch (e) {  
    return [];  
  }  
};
```

Şimdi de saveNotes :

```
const saveNotes = (notes) => {  
  //notes isminde array alıp dosyaya  
  yazdıracak  
  const dataJSON = JSON.stringify(notes);  
  //json formatı ama stringe çevrilmiş halı  
  fs.writeFileSync("notes.json", dataJSON);  
  //notes.json dosyasına datajson gönderılcek  
};
```

Remove :

```
const removeNote = (title) =>{  
  //başlık silme fonks  
  const notes = loadNotes(); //notları aldık  
  //notları tespit etmem gerekiyor ama  
  filtreleme ile cunku not olmama ihtimalı de  
  var  
  const notesToKeep = notes.filter((note) =>  
note.title !== title);  
  if (notes.length > notesToKeep.length) {  
    //başlangıctaki not sayısı sonrakinden  
    buyukse silinmiştir aksi halde silme yoktur  
    console.log(chalk.green.inverse("Note  
remove")); //inverse terse çekiyor renkleri  
    ondeki ile arkadaki  
    saveNotes(notesToKeep);  
  } else {  
    console.log(chalk.red.inverse("Silme  
istediğiniz not bulunamamıştır."));  
  }  
};
```

Son olarak addNote :

```
const addNote = (title, body) => {
  //notları dosyaya kaydedicez
  const notes = loadNotes(); //array
  //aynı başlık olma durumunda kosula
  //note.title // notesin içindeki
  //elemanlardan bir tanesi
  const duplicateNotes = notes.filter((note)
=>note.title === title);
  if (duplicateNotes.length === 0) {
    //hiçbir eşleşme yok,yeni not eklenebilir
    notes.push({
      title: title,
      body: body,
    }); //arraye eleman eklemek
    console.log(chalk.green.inverse("yeni not
eklendi")); // arrayin içindekileri consola
    yazdırma
    saveNotes(notes);
  } else {
    //o başlık daha once alınmış
    console.log(
      chalk.red.inverse("Bu başlık daha once
kullanıldı.Not eklenemiyor!!!")
    );
  }
};
```

Bu yeni versiyon hali ile de kodu çalıştırdık. Çıktılar şu şekilde oldu :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\bilmuh\Desktop\ada> node app.js add --title="ba" --body="mlk"
Bu başlık daha önce kullanıldı.Not eklenemiyor!!!
● PS C:\Users\bilmuh\Desktop\ada> node app.js remove --title="fba"
Silmek istediğiniz not bulunamamıştır.
● PS C:\Users\bilmuh\Desktop\ada> node app.js remove --title="ba" --body="mlk"
Note remove
● PS C:\Users\bilmuh\Desktop\ada> node app.js add --title="ba" --body="mlk"
yeni not eklendi
PS C:\Users\bilmuh\Desktop\ada> 
```

List ve read fonksiyonlarımızı henüz yapmamıştık, şimdi de onları oluşturduk.

Yeni fonksiyon eklemekle başlayalım: listNotes ve onu export edelim.

```
const listNotes = () => {

}

module.exports = {
  //sag ile eşleşmek zorunda,sol farklı ısm
  alabilir
  getNotes: getNotes,
  addNote: addNote,
  removeNote: removeNote,
  listNotes : listNotes
};
```

App.js de listeleme komutu içerisine dönelim.



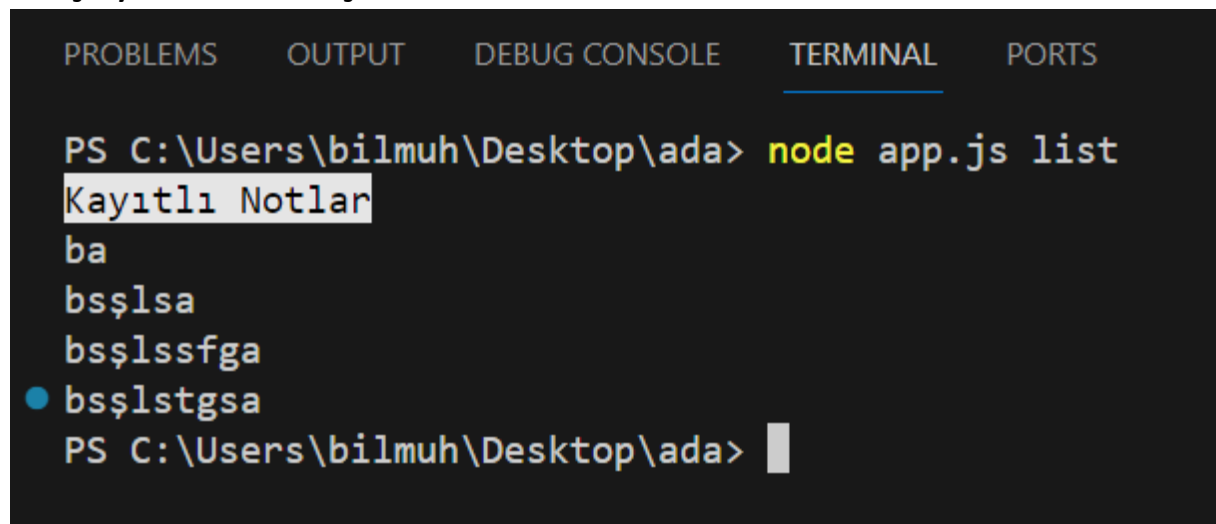
Az önce yazılan fonksiyonu çağırdık.

```
yargs.command({
  command: "list",
  describe: "mevcut notu listeler",
  handler () {
    notes.listNotes()
  },
});
```

Yazılan fonksiyonun içini doldurduk.

```
const listNotes = () => {
  const notes = loadNotes()
  console.log(chalk.inverse('Kayıtlı Notlar'))
  notes.forEach((note) => {
    console.log(note.title)
  });
}
```

Notları loadNotes fonksiyonu ile yükledik ardından kayıtlı notlar başlığı altında,notların başlıklarını yazdırdık. Çıktı olarak da şöyle bir sonuç aldık :



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\bilmuh\Desktop\ada> node app.js list
Kayıtlı Notlar
ba
bsşlsa
bsşlssfga
● bsşlstgsa
PS C:\Users\bilmuh\Desktop\ada> 
```

Şimdi read yazmadan önce,addNote üzerinden bir değişikliğe gittik bu değişiklik performans açısından artım sağladı.

Filter içerisinde ;elemanların başlıklarını kontrol etme sırasında eğer başlık ilk seferde bulunursa boş yere diğer başlıkları taramasına gerek kalmayacak.

Bu işlem için de find fonksiyonu kullanacağız. Önceki kodla find kullanılma haline bakacak olursak :

```
//const duplicateNotes =  
notes.filter((note) =>note.title === title);  
const duplicateNote = notes.find((note) =>  
note.title === title);
```

Find ve filter kullanımına bakmak gerekirse filter, bulunan başlıkların tamamını tararken; find ise bulma durumunda komuttan çıkar.

Bununla beraber koşul parametremiz de değişir. Artık boyut olarak değil var mı, yok mu diye bakacağız.

İlk durumda length in 0 olup olmamasına bakmıştık. Artık eşleşme olup olmama durumuna bakıyoruz.

Çünkü find komutunu kullanırken bulma durumuna göre diğer başlıklara bakmıyoruz. Ve çıktı olarak da boolean değer döndürüyor.

Filterde ise length ile sıfır olma durumunda eşleşme olmadığını aksi halde de eşleşme olduğunu belirtiyordu.

Koda da bakarsak şu şekilde :

```
//if (duplicateNotes.length === 0) {  
if (!duplicateNote) {  
  //hiçbir eşleşme yok,yeni not eklenebilir  
  notes.push({  
    title: title,  
    body: body,  
  }); //arraye eleman eklemek  
  console.log(chalk.green.inverse("yeni not eklendi"));  
  saveNotes(notes);  
} else {  
  //o başlık daha önce alınmış  
  console.log(  
    chalk.red.inverse("Bu başlık daha önce kullanıldı.Not eklenemiyor!!!")  
  );  
}
```

Şimdi read e geçtik (app.js). Okuma işleminde de title ı aldık. Builderi ona göre yazdık.

```
yargs.command({  
  command: "read",  
  describe: "secilen notu gosterir",  
  builder: {  
    title: {  
      describe: "Not basligi",  
      demandOption: true,  
      type: "string",  
    },  
  },  
  handler () {
```

```
    console.log("not gosteriliyor");  
  },  
});
```

Notes.js içerisinde fonksiyon oluşturup export ettik,

```
const readNotes = (title) => {  
  
}  
  
module.exports = {  
  //sağ ile eşleşmek zorunda,sol farklı isim  
  alabilir  
  getNotes: getNotes,  
  addNote: addNote,  
  removeNote: removeNote,  
  listNotes : listNotes,  
  readNotes : readNotes  
};
```

Handler içerisinde argv aldık ve parametre olarak argv.title gönderdik.

```
yargs.command({  
  command: "read",  
  describe: "secilen notu gosterir",  
  builder: {  
    title: {  
      describe: "Not basligi",  
      demandOption: true,  
      type: "string",  
    },  
  },  
  handler (argv) {  
    notes.readNotes(argv.title)  
  },  
});
```

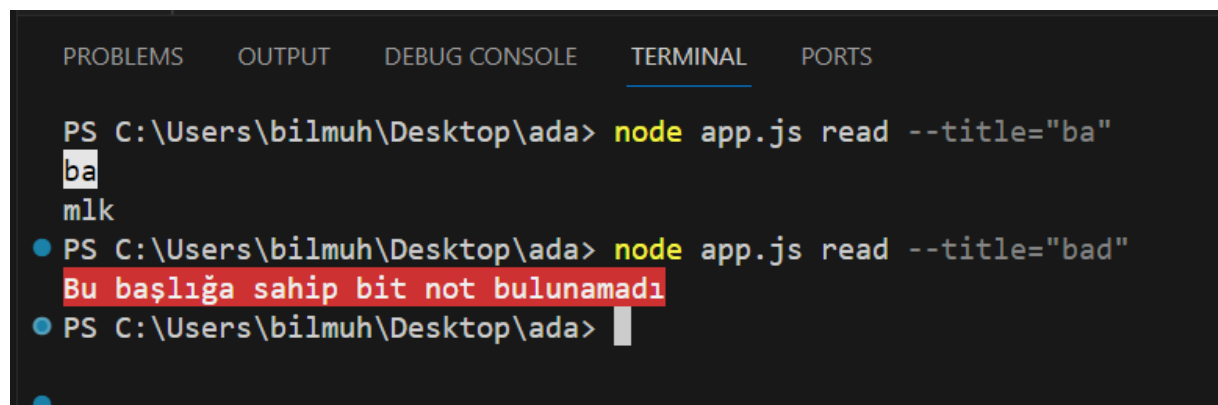
Tekrar notes.js dosyasına döndük ve içeriğini doldurduk.

Notlarımızı loadNotes ile çektik. Filtreleme olarak yine find fonksiyonunu kullandık, boş yere fazladan aramamış olduk.

```
const readNotes = (title) => {
  const notes = loadNotes();
  const note = notes.find((note) =>
note.title ===title)
  if(note){
    //note varsa
    console.log(chalk.inverse(note.title))
    console.log(note.body)
  }
  else{
    //not yoksa
    console.log(chalk.red.inverse("Bu başlığa
sahip bir not bulunamadı"))
  }
}
```

Find ile tarama yaptık. Note olma durumunda da if ile koşula soktuk ve girilen başlığa ait notun başlık ve body bilgisini çıktı olarak atadık, aksi halde notun olmadığına dair bir çıktı verdik

Konsolda çalışmaya bakarsak da :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\bilmuh\Desktop\ada> node app.js read --title="ba"
ba
mlk
● PS C:\Users\bilmuh\Desktop\ada> node app.js read --title="bad"
Bu başlığa sahip bir not bulunamadı
● PS C:\Users\bilmuh\Desktop\ada>
```

Sonra da debugging işlemine baktık(notes.js içerisinde)  
addNotes içerisinde hiç else e düşmezse bunu sorgulatmak için duplicatenotes i yazdırabiliriz.

```
const addNote = (title, body) => {  
  //notları dosyaya kaydedicez  
  const notes = loadNotes(); //array dondurcek  
  //note.title // notesin içindeki elemanlardan  
  bir tanesi  
  //const duplicateNotes = notes.filter((note)  
=>note.title === title);  
  const duplicateNote = notes.find((note) =>  
note.title === title);  
  //if (duplicateNotes.length === 0) {  
  
  console.log(duplicateNote)  
  
  if (!duplicateNote) {  
    //hiçbir eşleşme yok,yeni not eklenebilir  
    notes.push({  
      title: title,  
      body: body,  
    }); //arraye eleman eklemek  
    console.log(chalk.green.inverse("yeni not  
eklendi")); // arrayin içindekileri consola  
yazdırma  
    saveNotes(notes);  
  } else {  
    //o başlık daha once alınmış  
    console.log(  
      chalk.red.inverse("Bu başlık daha once  
kullanıldı.Not eklenemiyor!!!")  
    );  
  }  
};
```

Çıktısına bakarsak da şu şekilde :

```
• PS C:\Users\bilmuh\Desktop\ada> node app.js add --title="başlık" --body="içerik"
undefined
• yeni not eklendi
• PS C:\Users\bilmuh\Desktop\ada> node app.js add --title="başlık" --body="içerik"
{ title: 'başlık', body: 'içerik' }
• Bu başlık daha önce kullanıldı.Not eklenemiyor!!!
PS C:\Users\bilmuh\Desktop\ada>
```

Çıktıda undefined ve başlık bilgisi verdi.

İkinci olarak sorunun nerde olduğunu düşünürsek oraya breakpoint koyarız.

```
const addNote = (title, body) => {
  //notları dosyaya kaydedicez
  const notes = loadNotes(); //array
  //aynı başlık olma durumunda kosula
  //note.title // notesin içindeki
  //const duplicateNotes =
  notes.filter((note) =>note.title === title);
  const duplicateNote = notes.find((note) =>
  note.title === title);
  //if (duplicateNotes.length === 0) {

  console.log(duplicateNote)
  debugger
}
```

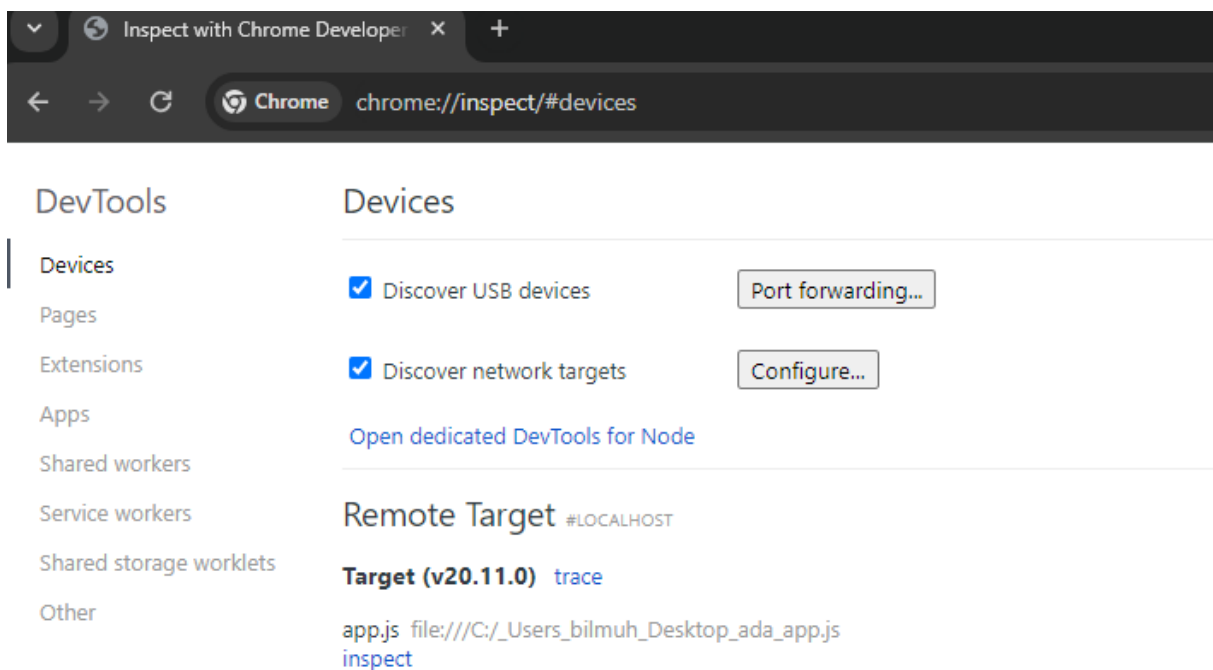
gibi.

Terminalde de şu şekil çalıştırırız:

```
PS C:\Users\bilmuh\Desktop\ada> node inspect app.js add --title="başl" --body="knmkj"
< Debugger listening on ws://127.0.0.1:9229/67f31f97-0e51-4875-a712-f0fc05d0f3a0
< For help, see: https://nodejs.org/en/docs/inspector
<
connecting to 127.0.0.1:9229 ... ok
< Debugger attached.
<
Break on start in app.js:1
> 1 const yargs = require("yargs");
  2 const notes = require("./notes"); //nokta bulunduğ u konum demek oluyor
  3 //require içersinde obje geliyor bu yuzden gecer isim vermek gerekiyor
debug>
debug>
debug>
debug>
debug>
debug>
```

Ardından Chrome a girdik ve şunu arattık :

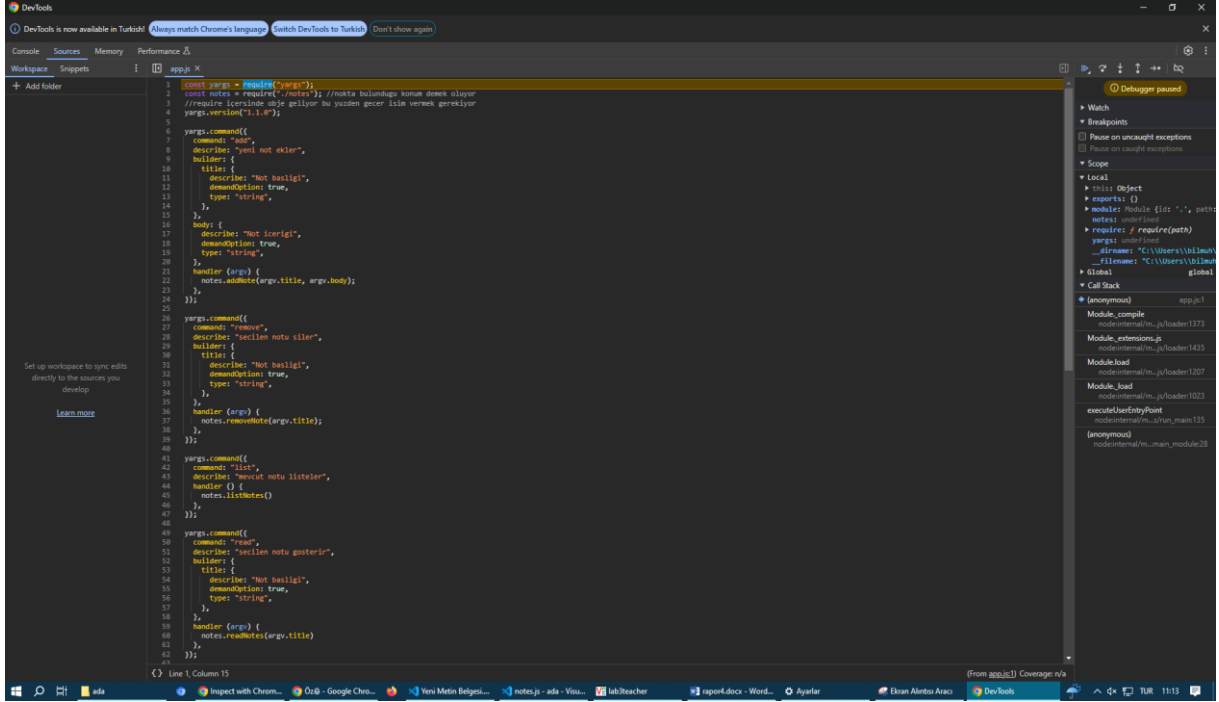
chrome://inspect/#devices



Inspecte tıkladık.



Çıkan ekran görüntümüz şu şekilde :



Sağ üstteki oka basma durumunda debugging başlayacak.

Ama hemen başlatmadık.

İlk önce soldaki add folder ile dosyayı bulduk. Notes.js de oka bastık ve kod debugging bulana kadar çalışma işleminde oldu. Ve bu durumun içerisinde de console a yazma işlemini yaptık.