

Web Programlama

12. Hafta

09.12.2024



Mühendislik
Fakültesi
Bilgisayar Mühendisliği

Hazırlayan: Dr. Ercan Ezin

GİRİŞ

Arka Uç Teknolojisine Giriş-Node.JS

Not: Bu dersin içeriği kaynakçada belirtilen materyallerde derlenerek üretilmiştir.


Credit: Content of the course has been picked from various sources which has been generously shared by the authors. See credit page for full list.

NODE JS

- **Heryerde Javascript Çalıştırın**
-
- Node.js açık kaynaklı, ücretsiz ve platformlar arası desteği olan bir JavaScript çalışma-zamanlı ortam ortam sunarak geliştiricilere sunucu oluşturmayı, web uygulamaları, komut satırı uygulamaları ve script yazmalarına olanak sunar.
-




DOWNLOAD

 [Learn](#) [About](#) [Download](#) [Blog](#) [Docs](#) [Contribute ↗](#) [Certification ↗](#)

[Read](#) [Node.js Collab Summit Report →](#)

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

[Download Node.js \(LTS\)](#) 

Downloads Node.js **v22.12.0**¹ with long-term support.
Node.js can also be installed via [package managers](#).

Want new features sooner? Get **Node.js v23.3.0**¹ instead.

[Create an HTTP Server](#) [Write Tests](#) [Read and Hash a File](#) [Stre](#)

```
// server.mjs
import { createServer } from 'node:http';

const server = createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello World!\n');
});

// starts a simple http server locally on port 3000
server.listen(3000, '127.0.0.1', () => {
  console.log('Listening on 127.0.0.1:3000');
});

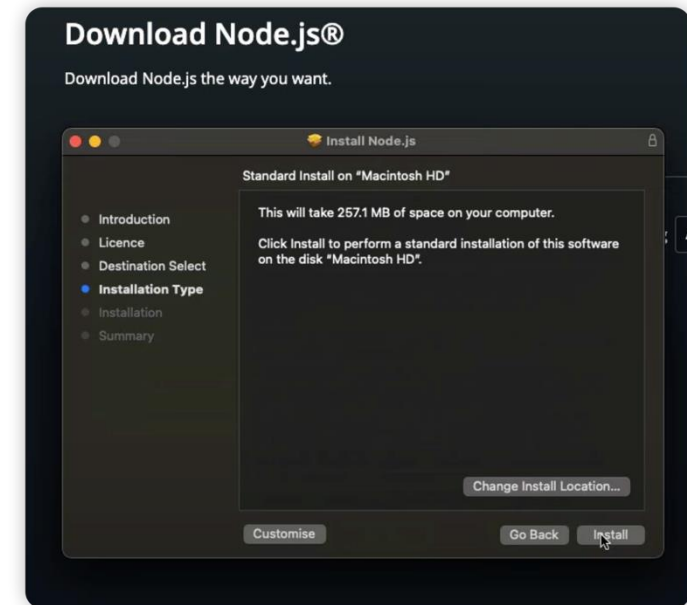
// run with `node server.mjs`
```

JavaScript [Copy to clipboard](#)

Learn more what Node.js is able to offer with our [Learning materials](#).

INSTALLATION

- Windows: <https://www.youtube.com/watch?v=IFcRKoi48hg>
- MacOS: <https://www.youtube.com/watch?v=I53HbzbSwxQ>
- Linux: <https://www.youtube.com/watch?v=IOxmyHKIAcw>



NODE JS : SIMPLE HTTP SERVER

```
// server.mjs
import { createServer } from 'node:http';

const server = createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello World!\n');
});

// starts a simple http server locally on port 3000
server.listen(3000, '127.0.0.1', () => {
  console.log('Listening on 127.0.0.1:3000');
});

// run with `node server.mjs`
```

NODE JS : MODULES

Node.js kullanım kolaylığı açısından modüllere sahiptir. Bir takım hazır fonksiyon barındıran bu modüller temelde JavaScript kütüphaneleridir.

2 tip modül vardır. Ön-tanımlı([built-in](#)) ve sonradan yüklenenler.

Bir modül eklemek için `var http = require('http');`

http modülü en temel modüllerden biridir. Uygulamanın lokal sunucu oluşturması için gereklidir.



```
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-  
Type': 'text/html'});  
  res.end('Hello World!');  
}).listen(8080);
```

NODE.JS: CUSTOM MODULES

Node.js üzerinde kendi modüllerinizi oluşturabilirsiniz. Örneğin aşağıda güncel tarihi geri dönen modül oluşturulmuştur ve kullanımı gösterilmiştir.

ilk_modul.js

```
exports.myDateTime = function () {  
  return Date();  
};
```

UYARI: Bu dosya server.js ile aynı dizinde olmalıdır.

Komut satırında sunucuyu çalıştırmak için:
`$node server.js`

server.js

```
var http = require('http');  
var dt = require('./ilk_modul');  
  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write("Saat ve Tarih: " + dt.myDateTime());  
  res.end();  
}).listen(8080);
```


NODE JS : HTTP MODULE

Eğer sunucudan alınacak içerik html veya metin içeriyorsa, bir header eklenmelidir. Burda 200 sayısı herşeyin yolunda gittiğini belirtir.

```
var http = require('http');  
http.createServer(function (req, res) {  
  → res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write('Hello World!');  
    res.end();  
}).listen(8080);
```

NODE JS : READ QUERY STRING

Bir sunucu oluşturduğumuzda **req** isminde bir değişken tanımlarız. Bu değişken kullanıcıdan gelen isteklerde bulunan bilgileri almamıza yardımcı olur.

server.js



```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(req.url);
  res.end();
}).listen(8080);
```

Yukarıdaki server çalıştırılıp tarayıcıda
<http://localhost:8080/harran> sayfası için
istek yollanırsa.

Viewportta /harran yazıldığı görülür.

NODE JS : SPLIT THE QUERY

Sunucuya yapılan isteğin içerdiği değişkenleri suncunun **req** değişkenini kullanarak ayrıştırabiliriz.

Ör : <http://localhost:8080/?yil=2024&ay=Ocak>

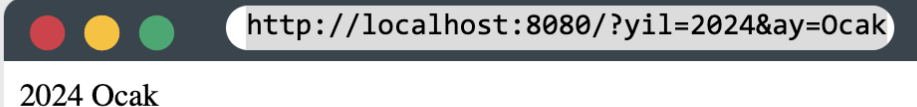
```
var http = require('http');
var url = require('url'); // used for url operations

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
  res.end(txt);
}).listen(8080);
```

true parametresi ile
Otomatik ayrıştırma

```
{
  yıl: "2024",
  ay: "Ocak"
}
```

Viewport Çıktısı



NODE JS : FILE SYSTEM

Node.js ile sunucu üzerindeki dosyaları gelen kullanıcı isteklerine göre manipüle edebilirsiniz. `require('fs')` ile ekleyeceğiniz `file system` modülü ile şu işlemleri yapabilirsiniz. Aşağıdaki listeye tıklayıp tüm operasyonları inceleyebilirsiniz.

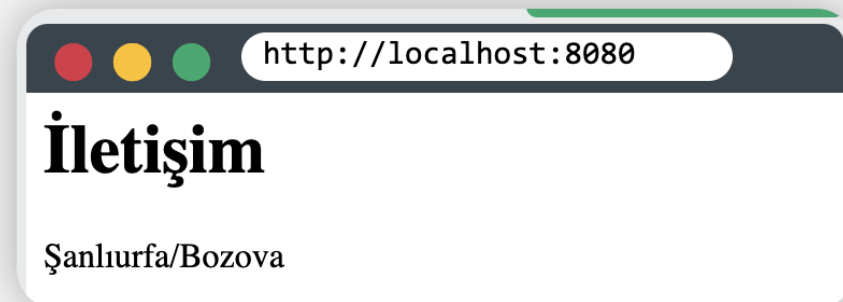
- [Dosya Okuma](#)
- [Dosya Oluşturma](#)
- [Dosya Güncelleme](#)
- [Dosya Silme](#)
- [Dosyayı Yeniden Adlandırma](#)

iletisim.html

```
<html>
<body>
<h1>İletişim</h1>
<p>Şanlıurfa/Bozova</p>
</body>
</html>
```


```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('iletisim.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

Viewport çıktısı:



NODE JS : FILE SERVER

Node.js aracılığıyla query string parametresine bağlı olarak kullanıcıya içerik sunabiliriz.



```
var http = require('http');
var url = require('url');
var fs = require('fs');

http.createServer(function (req, res) {
  var q = url.parse(req.url, true);
  var filename = "." + q.pathname;
  fs.readFile(filename, function(err, data) {
    if (err) {
      res.writeHead(404, {'Content-Type': 'text/html'});
      return res.end("404 Not Found");
    }
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

Yandaki örnekte kullanıcının yollayacağı pathname değişkenine bağlı olarak ilgili html sayfası dönülecek veya bir hata olduğunda **404** bulunamadı hatası dönülecektir.

Örnek url:

url: localhost:8080/iletisim.html

url: localhost:8080/hakkinda.html

url: localhost:8080/urunler.html

Eğer böyle bir sayfa yoksa 404

NODE JS : URL MODULE

Url modülü ile kullanıcıların yönetilmesi için birçok operasyon yapılabilmekte. Örneğin kullanıcının uygulamanın IP adresi, yolladığı parameterler alınıp gerekirse çeşitli operasyonlarda kullanılabilmekte.

```
var url = require('url'); //url modulünü yükler:  
var adr = 'http://localhost:8080/default.htm?yil=2024&ay=Ocak'; //Adresi ayrıştırır.  
var q = url.parse(adr, true); // parse metodu sunucuya yapılan isteğin adresini ayrıştırıp geri döner.  
console.log(q.host); // Ayrıştırılan host değişkenine erişim bu şekilde olur.  
console.log(q.pathname); // Ayrıştırılan pathname değişkenine erişim bu şekilde olur.  
console.log(q.search); // Aynı şekilde istek içerisindeki tüm parametreler bu değişkenle erişilir.
```

```
var qdata = q.query; // Ayrıştırılan parametrelere doğrudan erişmek için query alınıp  
console.log(qdata.ay); //Daha sonra bireysel olarak parametrelere ulaşılır.
```

Konsol çıktısı:

```
localhost:8080  
/default  
?yil=2024&ay=Ocak  
Ocak
```

NODE JS : NPM

Node.js için bir kütüphane yöneticisi olarak NPM(Node Package Manager) yaygın olarak kullanılmaktadır. Node.js kurulumu sırasında otomatik olarak yüklenen npm yüzlerce kütüphaneyi ücretsiz olarak sunmaktadır. Kütüphaneler www.npmjs.com adresinden inceleyebilirsiniz.

Bir kütüphaneyi projenize yüklemek için komut satırında şu komutu kullanabilirsiniz.

```
$npm install <module adi veya adları>
```

Ör: `$npm install async`

Veya: `$npm install express mongoose body-parser`

Daha sonra bir modülü projede kullanmak için : `var asy = require('async');`

NODE JS : NODEMAILER

Node.js projelerinizde bazen kullanıcıya mail

Yollamanız gerekebilir. Beğişik amaçları olan bu işlem için nodemailer modülü kullanılır. Spam gibi durumlarla karşılaşmamak ve Mail Server maliyetlerinden kaçınmak için Gmail gibi mail servislerinin sunduğu imkanları kullanarak yerel bilgisayarımızdan veya sunucudan mail yollayabiliriz. Gmail SMTP servisleri için basitleştirilmiş yetkilendirme protokolünü etkinleştirmeniz gerekir. [Link](#)

```
var nodemailer = require('nodemailer');

var transporter = nodemailer.createTransport({
  service: 'gmail',
  host: 'smtp.gmail.com',
  port: 587,
  auth: {
    user: 'mailadresiniz@gmail.com',
    pass: 'mailsifreniz'
  }
});

var mailOptions = {
  from: 'mailadresiniz@gmail.com',
  to: 'alici@yahoo.com',
  subject: 'Node.js Deneme',
  text: 'Merhaba dunyali!'
};

transporter.sendMail(mailOptions, function(error, info){
  if (error) {
    console.log(error);
  } else {
    console.log('Email Yollandi: ' + info.response);
  }
});
```

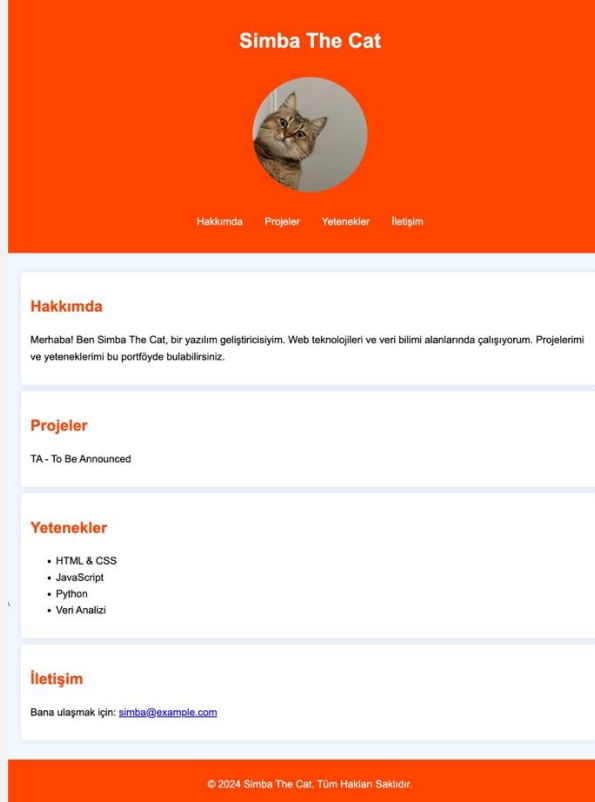

NODEMAILER EXAMPLE APPLICATION

Nodemailer konusunda projenizde kullanıcıya mail attığınız bir geliştirme istenecektir.

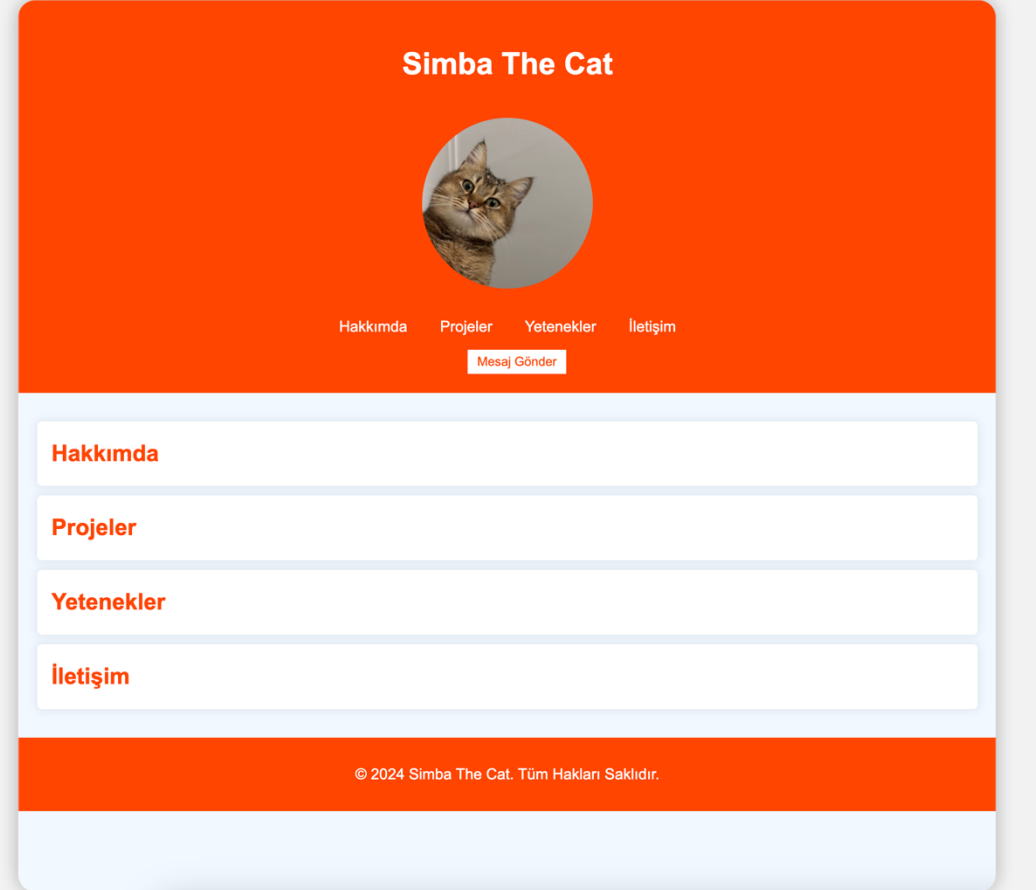
PROJE FAZ 2

Teslim Tarihi: 22.12.2024. Esneme olmayacaktır.

Proje isterleri dersin websitesinde yayınlanmıştır.



Faz 1



Faz 2

SEND MESSAGE

Simba The Cat



Hakkımda Projeler Yetenekler İletişim

Mesaj Gönder

Mesaj Gönder

Email (Zorunlu)

Konu (Zorunlu)

Mesajınız (Zorunlu)

Gönder

Yeni Mesaj: Merhaba Proje Faz 2



Website Contact: <> <hru.ceng.students@gmail.com>
to dr.ercan.ezin ▾

Gönderen: dr.ercan.ezin@gmail.com

Bu satırlar mesajın gövdesidir.

Mesajınız alındı Inbox x



Website Confirmation <hru.ceng.students@gmail.com>
to me ▾

Merhaba,

Mesajınızı aldık. Teşekkürler!

← Reply

→ Forward





EOF

REFERENCES / CREDITS

1. Stepp M, Miller J, Kirst V. Web Programming Step by Step. Step by Step Publishing; 2012.
2. CS193X Web Programming Fundamentals Course Slides at Stanford Uni by Victoria Kirst
3. <https://www.w3schools.com/nodejs/>
4. <https://nodejs.org/en>