

Back End-utveckling i PHP

50 YHP

idag: repetition
onsdag: olika dataformat I
torsdag: olika dataformat II
fredag: Start med PHP



Jag har ändrat schemat
fredagar 09.30 📌 10.00

REPETITION

Tisdag 23 Januari 2018

Vad har vi pratat om hittills?

- Backend vs Frontend
- Parsning av kod
- Vad man har en Server till
- Objektorienterad programmering
- Javascript i backend med NodeJS

Vad är  ?

**En runtime (parser) för
javascript filer att köra i
terminalen**

och...

**ett standardbibliotek för att
göra server applikationer**

standardbibliotek i Node JS

- Async Hooks
- Buffer
- Child Processes
- Cluster
- Command Line Options
- Console
- Crypto
- Debugger
- DNS
- Domain
- ECMAScript Modules
- Errors
- Events
- **File System**
- Globals
- **HTTP**
- HTTP/2
- HTTPS
- Inspector
- Internationalization
- Modules
- Net
- OS
- **Path**
- Performance Hooks
- Process
- Punycode
- Query Strings
- Readline
- REPL
- Stream
- String Decoder
- Timers
- TLS/SSL
- Tracing
- TTY
- UDP/Datagram
- URL
- Utilities
- V8
- VM
- ZLIB



Vad är PHP?

övning

- skapa en ny mapp för dagens övningar kallad **"repetitionen"**
- Öppna mappen i **VS CODE** och **en terminal**
- Skapa en fil som heter **index.js** med som skriver ut **hello world** med hjälp av console
- kör programet i terminalen med **node index.js**

Moduler i NodeJS

- Installeras och hanteras med verktyget **npm**
- På <https://www.npmjs.com> listas alla existerande bibliotek
- Sparas i mappen `node_modules`

Moduler: Sjukt viktigt

- PRO: Moduler tillåter en att luta sig tillbaka på andras kunskap
- PRO: Man behöver inte "uppfinna hjulet på nytt"
- CON: Andra personers kod körs på din server
- CON: Moduler kan innehålla buggar
- **Som programmerare förväntas man kunna hitta och välja säkra och bra moduler**

Hur hittar man en bra modul?

 `npm install nodemailer`

[how? learn more](#)

 **andris** published 3 days ago

4.4.2 is the latest of 199 releases

github.com/nodemailer/nodemailer

nodemailer.com

MIT

Collaborators [list](#)



Stats

74 069 downloads in the last day

439 751 downloads in the last week

1 298 273 downloads in the last month

One open issue on GitHub

No open pull requests on GitHub

övning

- Hitta ett paket på npm som låter er sätta en **färg på texten i terminalen**
- Installera modulen från terminalen med **npm install**
- Bygg ut er `index.js` så att ni använder modulen
- kör programet i terminalen med **node index.js**

**Frontend
eller
Backend**

Frontend: det användaren ser,
ofta samma som client-side.

Backend: implementation av
problemlösningen, ofta samma som
server-side.

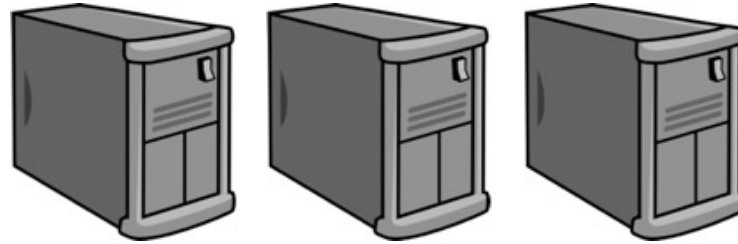


vad gör en server?



En server låter flera användare & datorer
dela på samma information

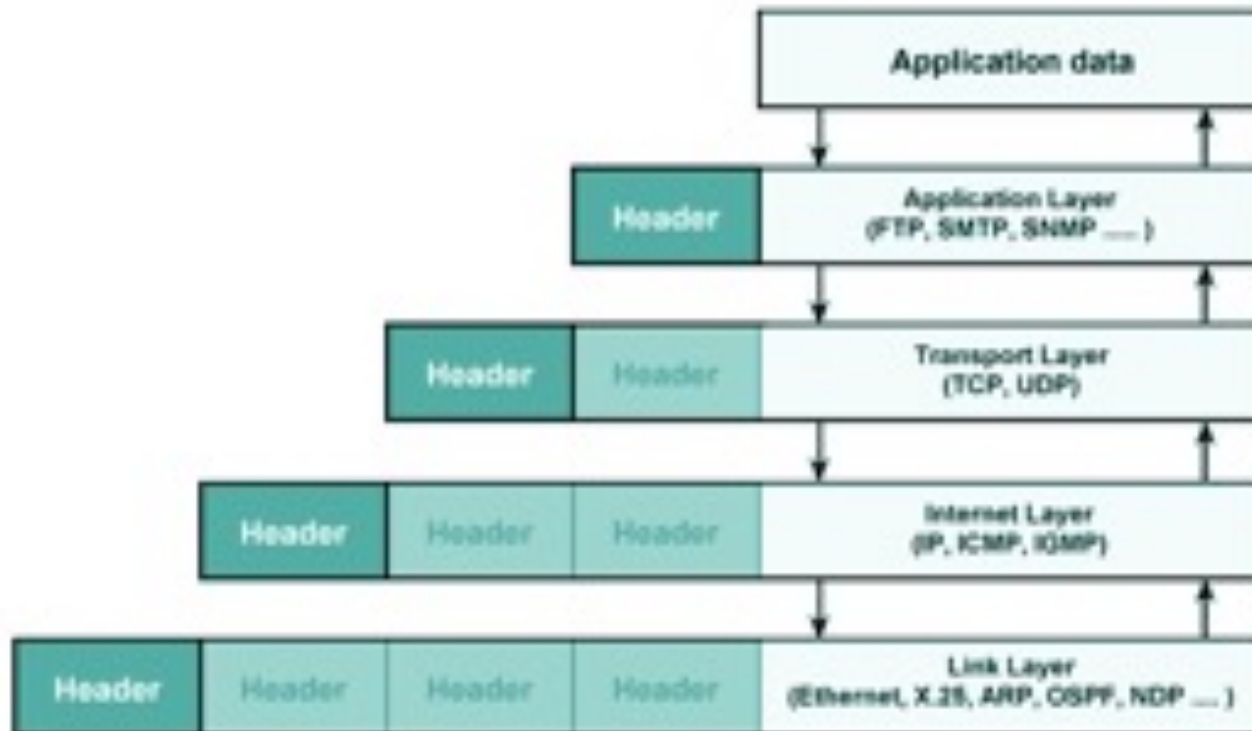
eller låter en användare nå samma
information från många olika sidor



Molnet är massor av datorer som arbetar
tillsammans

Internet (tcp/ip)

Hur skickas information mellan datorer?



övning

- Skapa en **index.html** fil i er mapp och fyll den med lämpligt innehåll
- I er **index.js** importerar ni standardbiblioteket **http** och **fs** och skapar en server som visar html sidan.
- Varje gång någon går ut på sidan ska den personens **IP adress skrivas ut** med `console.log` i en **valfri färg**
- När servern startar ska **servers IP** och **PORT skrivas ut** i en **valfi färg**

**vad är en
Parser?**



**En parser är ett program som läser
och tolkar filer och kod**

Kallas ofta för "runtime"

Exempelvis: NodeJS, Webbläsare, PHP,
C#...

Många andra program, exempelvis
film/bildvisning, musikspelare...

Hur funkar en enkel parser?

- Läs filen symbol för symbol
- leta efter efter kända tecken, som "{", "var", ";", "function" etc etc
- Växlar mellan olika lägen

exempelvis hur gör en webbläsare?

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function myFunction() {
        document.getElementById("demo").innerHTML = "Paragraph changed.";
      }
    </script>
  </head>
  <body>
    <h1>A Web Page</h1>
    <p id="demo">A Paragraph</p>
    <button type="button" onclick="myFunction()">Try it</button>
  </body>
</html>
```


övning

Skapa en ny fil i VS CODE med namnet **code.txt**.

- Skapa en **parser.js** fil där ni importerar standardbiblioteket **fs** och ert paket för att färga om terminalutskriften (**chalk**)

När filen körs ska följande hända:

- Filen ska läsas och skrivas ut symbol för symbol, om symbolen är "<" ska färgen ändras till **rosa** och vid ">" ska färgen ändras tillbaka till vanlig.

Tips:

- Använd **process.stdout.write** istället för **console.log**
- Läs hela filen på en gång med **fs.readFile** och kör sen en for loop för att gå igenom texten.

Tack!