Getting Started - Node.js

·      Node.js laster man ned fra<https://nodejs.org/en/download/>

·      I terminalen på mac må man først navigere til mappen som inneholder serverfilen.

·      Skriv inn «npm init -y» for å opprette en default package.json.

·      Skriv videre inn «npm install express socket.io» for å installere frameworket express og biblioteket socket.io.

·      Det kan være lurt å installere nodemon slik at man slipper å restarte serveren manuelt hver gang man forandrer noe i kildekoden. Den gjøres ved å skrive inn «npm install nodemon --save-dev». Deretter må man legge til «"dev": "nodemon server.js"» under «script»: i package.json. Det blir slik:

"scripts": {

   "start": "node server.js",

   "dev": "nodemon server.js",

   "test": "echo \"Error: no test specified\" && exit 1"

}

·      For å starte serveren normalt skriver man inn «node» og navnet på serverfilen. I mitt tilfelle blir dette «node server.js».

·      For å starte serveren og la den ta i bruk nodemons autorestart, skriver man inn «npm run dev».

Node.js filesystem

for å skrive, lese og modifisere filer og mapper kan man importere en native modul som heter fs, kort for filesystem. Skriv inn “const fs = require(‘fs’)” i toppen av programmet.

Funksjoner du kan bruke der er som følge:

* fs.readFile("filplassering", "skrifttype: vanligvis utf-8", "Callback funksjon");
* fs.writeFile("filplassering", "data", "Callback funksjon");
* fs.appendFile("filplassering", "data", "Callback funksjon");

Og mange flere, men disse er greie å begynne med. Flere funksjoner finner du på: <https://www.w3schools.com/nodejs/nodejs_filesystem.asp>.

De tre funksjonene over gjør som de sier. Her er et **eksempel** på en ferdig funksjon for å skrive tekst til en fil:

writeText = function(data){

fs.writeFile("data.txt", data, function(err, data){

  if(err)throw err;

});

}

(Om dokumentet ikke finnes, lages det av funksjonene write og append).

Express og Express View Engine

for å sende javascript og css til brukeren på en enkel måte kan du bruke Express. Express er den mest populære “frameworken” til node.js, og dette av god grunn. Det tar seg av utrolig mye og gjør livet ditt 27x lettere. For å installere det må du være i server mappen i konsollen/terminalen og skrive “npm install express --save”. For å navigere til en et gitt “directory” bruk kommandoen cd(change directory) og skriv adressen. f.eks i mitt tilfelle: *“cd dropbox/my\ folder/software\ projects/rainforest/\_server*”.

inkluder så modulen express i programmet ditt:

const express = require("express");

(Vi bruker “const” over “var” for å være sikker at vi ikke kan forandre på konstanten express senere)

For å bruke express til å sende javascript og css når brukeren navigerer på ditt domene setter vi app = express for å kunne bruke express frameworket. Så bruker vi app.use funksjonen for å sette en “static” mappe med alle filene klienten vil trenge. Denne blir da sendt når noen navigerer til denne serverens domene.

var app = express();

app.use(express.static(path.join(\_\_dirname, "/\_frontend")));

Moduler

* body-parser er en modul man bruker for å parse POST requests fra klienten slik at serveren kan lese dem. Brukt i egentlig alle node servere som har noe brukergrensesnitt hvor brukeren sender data til serveren. F.eks login, create user, etc.
* Egne moduler:

Egne moduler kan man lage for å ha en mer ryddig hovedfil. Disse er nesten som javascript bibliotek, bare at her må man spesifisere at man skal importere de. Det må også spesifiseres hvilke funksjoner som skal eksporteres i modulen.

I app.js:

var mymod = require("./myModule");

I modulen:

function myFunction(){

  console.log("Useless function");

}

module.exports.myFunction = myFunction;

Da kan funksjonen myFunction brukes i app.js som en vanlig funksjon laget i samme fil.

BASIC Node Server med Express

For å lage en helt basic server som viser både html, css og javascript trenger du bare 8 linjer med kode: (path er en integrert node modul som ikke må installeres)

const express = require("express");

const path = require("path");

var app = express();

app.use(express.static(path.join(\_\_dirname, "/\_frontend")));

app.listen(8080, function(){console.log("Server Started on port 8080...")});

app.get('/', function(req, res){

  res.sendFile(\_\_dirname + '/\_frontend/landing-page.html');})

app.get er en GET request Handler. Når noen spør om bare nettsiden f.eks localhost:8080 vil de komme til localhost:8080/ (hjemmesiden). Da vil serveren sende html filen landing-page.html, og css og javascripten blir hentet av html-filen fra static mappen.