# Express Routing Harjoitus

Refaktoroidaan edellisen tapaamisen [02 handling forms](https://github.com/Laurea-ammattikorkeakoulu-Tikkurila/TO0065BS-FullStack-Master-2024/blob/main/WS04_Server_Routes_and_FORM_handling/tuntiharjoitukset/02_handling_forms.js) tuntiharjoitus käyttämällä [Express router](https://expressjs.com/en/5x/api.html#router) toiminnallisuutta.

Tee tehtävä uuteen kansioon

Reitit voit tehdä halultessasi eri tiedostoihin tai yhteen tiedostoon..

// ladataan tarvittavat modulit

var express = require('express');

var app = express();

var fs = require('fs');

//haetaan dotenv moduuli

const dotenv = require('dotenv').config();

// luodaan palvelinportti

const PORT = dotenv.parsed.PORT02 || 3002;

// Serve static files from the "public" directory

app.use(express.static('public'));

// luo juurireitti, joka palauttaa selaimeen index.html tiedoston sisällön

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

    //res.send('Hello World!');

    res.sendFile(\_\_dirname + '/index.html');

});

// luo toinen reitti, joka palauttaa selaimeen tekstidata.txt tiedoston sisällön

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

    //res.send('Listing data from a file!');

    res.sendFile(\_\_dirname + "/data/tekstidata.txt");

});

// luo kolmas reitti, joka palauttaa selaimeen JSONdata.json tiedoston sisällön

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

    var data = require(\_\_dirname + '/data/JSONdata.json');

    res.json(data);

});

// parse application/x-www-form-urlencoded

const bodyParser = require('body-parser');

//app.use(bodyParser.urlencoded({ extended: false }));

app.use(bodyParser.urlencoded({

    extended: true

})); // for parsing application/x-www-form-urlencoded

// luodaan reitti, joka parsii .json tiedoston sisällön taulukkomuotoon ja palauttaa sen selaimeen

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

    var data = require(\_\_dirname + '/data/JSONdata.json');

    // Parse the results into a variabke

    var results = '<table border="1"> ';

    for (var i = 0; i < data.length; i++) {

        results +=

            '<tr>' +

            '<td>' + data[i].Name + '</td>' +

            '<td>' + data[i].Email + '</td>' +

            '<td>' + data[i].Date + '</td>' +

            '</tr>';

    }

    res.send(results);

});

// lisätään GET polku (route) joka hakee /public/adduser.html tiedoston ja lähettää sen selaimeen

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

    res.sendFile(\_\_dirname + '/public/adduser.html');

})

// lisätään POST polku (route) joka hakee tiedot ja tallentaa ne JSONdata.json tiedostoon.

app.post('/adduser', function (req, res) {

    // Load the existing data from a file and assign to an array (lista)

    const data = require(\_\_dirname + '/data/JSONdata.json');

    //luodaan uusi henkilö...

    const nimi = req.body.name

    const email = req.body.email

    const company = req.body.company;

    const date = new Date().getDate() + "/" + (1 + parseInt(new Date().getMonth())) + "/" + new Date().getFullYear();

    //...ja pusketaan uusi käyttäjä listan (tiedoston) viimeiseksi

    data.push({

        "Name": nimi,

        "Company": email,

        "Email": company,

        "Date": date

    });

    // Convert the JSON object to a string format

    var jsonStr = JSON.stringify(data);

    // Write data to a file

    fs.writeFile(\_\_dirname + '/data/JSONdata.json', jsonStr, (err) => {

        if (err) throw err;

        console.log("tiedot tallennettu...")

    })

    res.send("Saved the data to a file. Browse to the /details to see the contents of the file");

});

app.get('\*', function (req, res) {

    //res.send('Cant find the requested page', 404);

    res.sendFile(\_\_dirname + '/public/error.html');

});

// käynnistetään palvelin kuuntelemaan valittua porttia

app.listen(PORT, function () {

    console.log('Example 02 app listening on port: ' + PORT);

});