Intro till express:

För att installer express i ditt projekt :

npm install --save express

Build a server med express:

const express=require('express')

//anropa express function

const app=express()

app.get('/',(req,res)=> {res.send('HelloWorld!')})

app.listen(3000,()=>console.log('Serverready'))

Spara detta i en index.js/app.js/main.js file. och kör filen.

const express=require('express');

//anropa express function

const app=express();

//alla req som kommer från localhost:3000

//kommer hanteras av den här app.get();

app.get('/',(req,res)=> { res.send('HelloWorld!')});

app.get("/home" , function(req, res) {

res.send("Nu är vi i home route");

//res.write + res.end

});

app.listen(8000,()=>console.log('Serverready'))

Övningar:

* Skapa en get request på /kurs endpoint
* Skicka ett objekt till ex.

{ kursnamn: “node js”, längd: 10 }

Environment Variable:

Den OS där programmet körs.

Vi ska lyssna på dynamisk port med hjälp av environment variable:

const port = process.env.PORT || 3000

app.listen(port,()=>console.log('Serverready'))

//Extra :

/\*

server.on('listening', function() {

console.log(server.address().port);

});

\*/

mer get request:

const express=require('express')

//anropa express function

const app=express()

app.get('/',(req,res)=>res.send('HelloWorld!'))

const kursArr = [

{kursNamn: "node js", längd:9},

{kursNamn: "ArbetsMetodik", längd:4},

]

//hämta hela lista

app.get('/kurser',(req,res)=>res.send(kursArr))

//hämta en specifika kur

app.get('/kurser/1',(req,res)=>res.send(kursArr[0]))

//hämta en specifika kur

app.get('/kurser/2',(req,res)=>res.send(kursArr[1]))

const port = process.env.PORT || 8000

app.listen(port,()=>console.log('Serverready' + port))

Vi ska pröva :

app.get('/kurser/3',(req,res)=>res.send(“objektet finns ej”))

//Objektet finns inte

// ska vi skapa många get endpoint?

Hur kan vi läsa URI parametrar : (Request.params)

app.get("/kurser/:id", (req,res)=>{

res.send(req.params.id)

res.send("some data")

})

skriv om all get “kurser/:id” endpoint :

app.get('/kurser/:id',(req,res)=>{

req.params.id >2 ?

res.status(404).send("objektet finns ej"): res.send(kursArr[req.params.id-1])

}

)

HTTP post req:

const express=require('express')

//anropa express function

const app=express()

//så att den kan läsa från req.body.kursNamn

app.use(express.json());

app.get('/',(req,res)=>res.send('HelloWorld!'))

const kurser = [

{kursNamn: "node js", längd:9},

{kursNamn: "ArbetsMetodik", längd:4},

]

//hämta en specifika kur

app.get('/kurser/:id',(req,res)=>{

req.params.id >kurser.length ? res.status(404).send("objektet finns ej"): res.send(kurser[req.params.id-1])

}

)

app.post("/kurser", (req, res)=>{

const kurs = {

kursNamn: req.body.kursNamn,

längd:9

};

kurser.push(kurs);

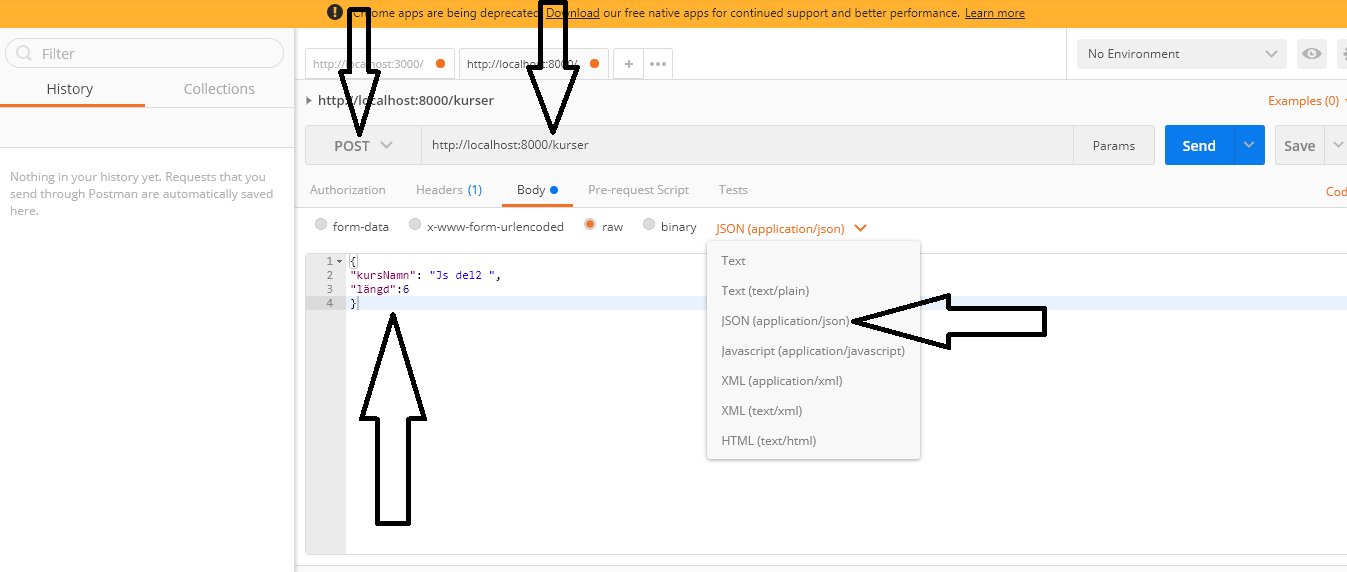
res.send(kurser);

})

const port = process.env.PORT || 8000

app.listen(port,()=>console.log('Serverready' + port))

Testa med Postman: <https://www.getpostman.com/>



Man väljer metoden, endpoint , dataformat till JSON och data.

Man ska kunna :

Environment variable, HTTP metoder, request params.

förståelse: Restful api, API endpoint .

Intro till express:

För att installer express i ditt projekt :

npm install --save express

Build a server med express:

const express=require('express')

//anropa express function

const app=express()

app.get('/',(req,res)=> {res.send('HelloWorld!')})

app.listen(3000,()=>console.log('Serverready'))

Spara detta i en index.js/app.js/main.js file. och kör filen.

const express=require('express');

//anropa express function

const app=express();

//alla req som kommer från localhost:3000

//kommer hanteras av den här app.get();

app.get('/',(req,res)=> { res.send('HelloWorld!')});

app.get("/home" , function(req, res) {

res.send("Nu är vi i home route");

//res.write + res.end

});

app.listen(8000,()=>console.log('Serverready'))

Övningar:

* Skapa en get request på /kurs endpoint
* Skicka ett objekt till ex.

{ kursnamn: “node js”, längd: 10 }

Environment Variable:

Den OS där programmet körs.

Vi ska lyssna på dynamisk port med hjälp av environment variable:

const port = process.env.PORT || 3000

app.listen(port,()=>console.log('Serverready'))

//Extra :

/\*

server.on('listening', function() {

console.log(server.address().port);

});

\*/

mer get request:

const express=require('express')

//anropa express function

const app=express()

app.get('/',(req,res)=>res.send('HelloWorld!'))

const kursArr = [

{kursNamn: "node js", längd:9},

{kursNamn: "ArbetsMetodik", längd:4},

]

//hämta hela lista

app.get('/kurser',(req,res)=>res.send(kursArr))

//hämta en specifika kur

app.get('/kurser/1',(req,res)=>res.send(kursArr[0]))

//hämta en specifika kur

app.get('/kurser/2',(req,res)=>res.send(kursArr[1]))

const port = process.env.PORT || 8000

app.listen(port,()=>console.log('Serverready' + port))

Vi ska pröva :

app.get('/kurser/3',(req,res)=>res.send(“objektet finns ej”))

//Objektet finns inte

// ska vi skapa många get endpoint?

Hur kan vi läsa URI parametrar : (Request.params)

app.get("/kurser/:id", (req,res)=>{

res.send(req.params.id)

res.send("some data")

})

skriv om all get “kurser/:id” endpoint :

app.get('/kurser/:id',(req,res)=>{

req.params.id >2 ?

res.status(404).send("objektet finns ej"): res.send(kursArr[req.params.id-1])

}

)

HTTP post req:

const express=require('express')

//anropa express function

const app=express()

//så att den kan läsa från req.body.kursNamn

app.use(express.json());

app.get('/',(req,res)=>res.send('HelloWorld!'))

const kurser = [

{kursNamn: "node js", längd:9},

{kursNamn: "ArbetsMetodik", längd:4},

]

//hämta en specifika kur

app.get('/kurser/:id',(req,res)=>{

req.params.id >kurser.length ? res.status(404).send("objektet finns ej"): res.send(kurser[req.params.id-1])

}

)

app.post("/kurser", (req, res)=>{

const kurs = {

kursNamn: req.body.kursNamn,

längd:9

};

kurser.push(kurs);

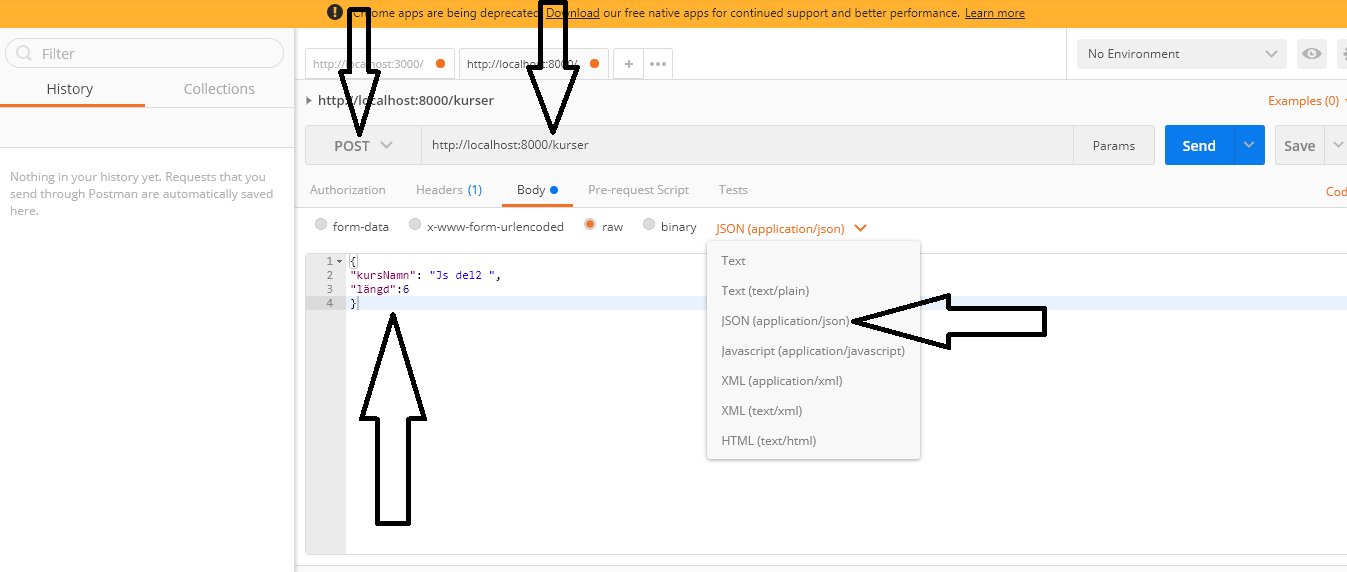
res.send(kurser);

})

const port = process.env.PORT || 8000

app.listen(port,()=>console.log('Serverready' + port))

Testa med Postman: <https://www.getpostman.com/>



Man väljer metoden, endpoint , dataformat till JSON och data.

Man ska kunna :

Environment variable, HTTP metoder, request params.

förståelse: Restful api, API endpoint .