EX06 Middleware

```
const express = require("express")

const app = express()

const PORT = process.env.PORT || 1234

app.get('/', (req,res)=>{
    res.send("TEST")
})

app.listen(PORT, ()=>{
    console.log(`The server running on port ${PORT} `);
})
```

מה המשמעות של middleware?

//req => middleware => res

custom middleware

```
const log = (req,res,next)=>{
    console.log(`${req.method} ${req.path}`)
    next()
}

app.get('/', log, (req,res)=>{
    res.send("TEST")
})
```

```
אחד middlewaren אחד app.get('/', [log, log2] ,(req,res)=>{
    res.send("TEST")
})

app.use(log)
```

פונקציית הmiddleware מקבלת 3 פרמטרים req, res נתוני ROUTES רגילים, וכן פרמטר של next המגדיר אפשרות של להמשיך הלאה, במידה ולא מריצים את הNEXT המערכת לא ממשיכה הלאה ולכן יש לשלוח RESPONSE למשתמש

```
app.use((req,res,next)=>{
    console.log(`${req.method} ${req.path}`)
    next()
})
```

Build-in middleware

הוספת אפשרות של שימוש בJSON

```
//build in middleware for json
app.use(express.json())
```

קבצים סטטיים

זה מגדיר תקייה שניתן לגשת לקבצים בתוכה, מתאים לקבצי CSS או לקבצי תמונות וכד'

//static files

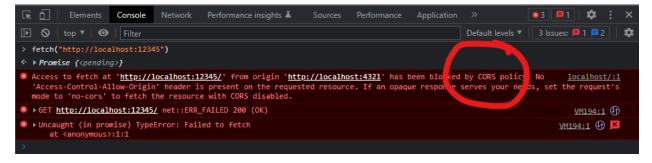
```
app.use(express.static("public"))
//app.use(express.static( path.join( dirname, "public") ))
                                JS server.js X # style.css
NODE-MONGO
                                 EX05 > JS server.js > ...
> EX01
                                   1 const express = require("express")
> EX01-complete
                                       const app = express()
> EX02
                                   3
> EX02-complete
                                   4
                                       const PORT = process.env.PORT | 1234
                                   5
> EX03-complete
                                      //build in middleawre for json
> EX04
                                       app.use(express.json())
> EX04-complete
                                   7
∨ FX05
                                   8
                                       //static files
 > node_modules
                                   9
 ∨ public
                                       app.use(express.static("public"))
                                  10
                                       //app.use(express.static( path.join(__dirname, "public") ))

✓ CSS

                                  11
                                  12
  # style.css
  ≡ test.txt
                                  13
                                  14
                                       app.get('/',(req,res)=>{
 {} package.json
                                  15
                                           res.send("TEST")
 Js server.js
                                  16
                                       })
 מדריך <
                                  17
                                  18
                                  19
                                       app.listen(PORT, ()=>{
                                  20
                                           console.log(`The server running on port ${PORT} `);
                                  21
                                       })
```

<u>צד ג' Middleware</u>

בעיית cors, צפייה בבעיה ב-2 שרתים שרצים במקביל



'התקנת חבילת צד ג

npm i cors

middleware corsב ואז ניתן להוסיף שימוש

```
const cors = require("cors")
app.use(cors())
```

שבפועל פתר את הבעיה

```
Promise {<pending>}
Filter

Filter
```

סיטואציה זו אינה מאובטחת והיא מתאימה רק לAPI שמעוניינים שיגשו אליהם מכל כתובת אך במערכת אצלינו אנו צריכים להגדיר את ההפנייות אליהם אנו מאשרים לגשת זה מתוך הדוקומנטציה של cors

Configuring CORS w/ Dynamic Origin

```
var express = require('express')
var cors = require('cors')
var app = express()
var whitelist = ['http://example1.com', 'http://example2.com']
var corsOptions = {
  origin: function (origin, callback) {
    if (whitelist.indexOf(origin) !== -1) {
      callback(null, true)
    } else {
      callback(new Error('Not allowed by CORS'))
    }
  }
}
app.get('/products/:id', cors(corsOptions), function (req, res, next) {
  res.json({msg: 'This is CORS-enabled for a whitelisted domain.'})
})
app.listen(80, function () {
  console.log('CORS-enabled web server listening on port 80')
})
```

לכן ניצור רשימה לבנה לדומיינים מהם אפשר יהיה לגשת לשרת הAPI שלנו ניצור תקיית config ותחתיה נוסיף קובץ corsOptions.js, המכיל מערך של כתובות שאותן נאפשר

```
∨ NODE-MONGO

                                     EX05 > config > Js corsOptions.js > ...
                                        1 const allowedOrigins = [
  > EX01
                                                  'http://localhost:3000',
 > EX01-complete
                                        2
 > EX02
                                        3
                                                  'http://localhost:5500'
 > EX02-complete
                                        4
                                        5
 > EX03-complete
                                            const corsOptions = {
 > EX04
                                        6
 > EX04-complete
                                        7
                                                 origin: (origin, callback) => {
                                                     if (allowedOrigins.indexOf(origin) !== -1 || !origin) {
 ∨ EX05
                                        8
                                        9
                                                          callback(null, true)

∨ confia

                                       10
 JS corsOptions.js
                                                     } else {
                                                          callback(new Error('Not allowed by CORS'))
  > logs
                                       11
                                       12
  > middleware
  > node_modules
                                       13
                                                 },
                                                 credentials: true,
  > public
                                       14
                                                 optionsSuccessStatus: 200
  {} package-lock.json
                                      15
  {} package.json
                                       16
  JS server.js
                                       17
  Js server2.js
                                            module.exports = corsOptions
 > מדריך
```

```
const allowedOrigins = [
    'http://localhost:3000',
    'http://localhost:5500'
]

const corsOptions = {
    origin: (origin, callback) => {
        if (allowedOrigins.indexOf(origin) !== -1 || !origin) {
            callback(null, true)
        } else {
            callback(new Error('Not allowed by CORS'))
        }
    },
    credentials: true,
    optionsSuccessStatus: 200
}

module.exports = corsOptions
```

תרגיל בית:

העמידו שרת EXPRESS, עם לפחות POUTES 4

1. הוסיפו את חבילת הCORS והשתמשו בה כמו שנלמד ומצורף בקובץ בנוסף

> 2. הוסיפו custom middleware שאתן בונות <u>בקובץ נפרד</u> לכתיבת LOG לכל הקריאות לקובץ נפרד (באמצעות FS) כך שלכל קריאה נוסיף שורה בקובץ הLOG

req.method , req.headers.origin, req.path ,**uuid** את התאריך את השעה דוגמא לשורות בקובץ הלוג

20230228 08:25:24 be66a9c5-431f-a708-ecbe819e7b44 GET

undefined /

20230228 10:37:31 e8192e6d-4081-b1f6-d4e5e2da7f5e GET

undefined /

20230228 12:29:16 9cb4020e-4d9d-be34-a1d39da6fcaa GET

undefined /

יש לעדכן את

SERVER, LOGGER , CORSOPTIONS, TEXTLOG קובץ