

Express.js
More middleware

Εισαγωγή

- Είδαμε μέχρι τώρα ορισμένα πράγματα για το middleware
- Είπαμε ότι οι Middleware συναρτήσεις που έχουν πρόσβαση στο
 - request object (req)
 - response object (res)
 - next middleware function
- στον κύκλο αιτήματος-απάντησης (request-response cycle) της εφαρμογής

Εισαγωγή

- Θα δούμε τώρα το **param middleware** το οποίο είναι επίσης χρήσιμο γιατί τρέχει για ορισμένες παραμέτρους.
- Θέλουμε, σε ορισμένες περιπτώσεις, όταν στο url υπάρχουν ορισμένες παράμετροι να τρέχουμε ένα στάνταρ κομμάτι κώδικα...
- Επίσης όμως στον προγραμματισμό δε θέλουμε να υπάρχει ο ίδιος κώδικας ξανά και ξανά

Εισαγωγή

- Θέλουμε να τον γράφουμε μια φορά και να τον χρησιμοποιούμε σε πολλά σημεία
- Όταν χρησιμοποιούμε `express.js`, αυτό το κάνουμε με το `middleware`..
- Για να δούμε πως στο δικό μας παράδειγμα...

- Μέχρι τώρα, η μόνη παράμετρος που χρησιμοποιούμε είναι το id.
- Έστω ότι θέλουμε σε όλα τα requests που κάνουν χρήση του id να ελέγχουμε αν όντως αυτό υπάρχει και αν όχι να εμφανίζουμε ένα μήνυμα λάθους.

```
const routes = express.Router();

//landmark Routes
routes.route('/')
  .get(landmarkController.getAllLandmarks)
  .post(landmarkController.addLandmark)

routes.route('/:id')
  .get(landmarkController.getLandmarkById)
  .patch(landmarkController.updateLandmarkById)
  .delete(landmarkController.deleteLandmarkById)

//lets export our module!
module.exports = routes;
```

- Στον controller ορίζουμε τη middleware function που θα την κάνουμε export για να την χρησιμοποιήσουμε στον router.

- Στον ορισμό της βλέπουμε ότι υπάρχει και η μεταβλητή val που κρατάει την τιμή της παραμέτρου (το id σε αυτή τη περίπτωση)

//req.params object-> used to access Route parameters

```
tourRoutes.js  JS tourController.js  JS landmarkController.js X  JS landmarkRou...
organized_3 > controllers > JS landmarkController.js > ...
6 //convert json to javascript object
7 landmarks = JSON.parse(landmarks);
8
9
10 exports.checkID = (req, res, next, val) => {
11   console.log(`Id is: ${val}`);
12
13   var id = req.params.id*1;
14   var landmark = landmarks.find(element => element.id === id)
15   if(!landmark){
16     return res.status(404).json({
17       status: 'fail',
18       message: 'Invalid ID! Sorry!'
19     });
20   }
21   next();
22 };
```

↓
ΔΕ ΟΧΙ ΞΕΧΝΑΜΕ

Στον router: **Param method** με 2 ορίσματα:

1. Ορίζουμε την παράμετρο που αφορά η μέθοδος
2. Ορίζουμε το middleware function

Έχουμε `app.param([name], callback)`

- Name: name of the parameter or an array of them

```
ized_3 > routes > JS landmarkRoutes.js > [?] <unknown>
//import express
const express= require('express');
//import landmarkController module
const landmarkController=require(`_${__dirname}../cont
//lets create a new router-> it is a middleware
const routes = express.Router();

routes.param('id', landmarkController.checkID);

//landmark Routes
routes.route('/')
.get(landmarkController.getAllLandmarks)
.post(landmarkController.addLandmark)

routes.route('/:id')
.get(landmarkController.getLandmarkById)
.patch(landmarkController.updateLandmarkById)
.delete(landmarkController.deleteLandmarkById)

//lets export our module!
module.exports= routes;
```

Πολλές Middleware Functions

- Μια ακόμη δυνατότητα που έχουμε είναι να χρησιμοποιούμε περισσότερες από μια middleware συναρτήσεις σε ένα request.
- Στο παράδειγμα μας, μέχρι τώρα χρησιμοποιούσαμε μόνο 1...

```
//landmark Routes
routes.route('/')
  .get(landmarkController.getAllLandmarks)
  .post(landmarkController.addLandmark)
  |
routes.route('/:id')
  .get(landmarkController.getLandmarkById)
  .patch(landmarkController.updateLandmarkById)
  .delete(landmarkController.deleteLandmarkById)
```

→ 1 συναρτηση

Πολλές Middleware Functions

- Έστω ότι θέλουμε να ελέγξουμε τα στοιχεία που εισάγει ο χρήστης πριν τα αποθηκεύσουμε στο json αρχείο μας.
- Θα μπορούσαμε να δημιουργήσουμε μια middleware συνάρτηση που να κάνει έναν τέτοιο έλεγχο..
- Και να την συνδέσουμε στην ήδη υπάρχουσα middleware συνάρτηση στο post request

Πολλές Middleware Functions

- Έστω ότι θέλουμε να ελέγξουμε αν το νέο landmark που πάει να αποθηκεύσει ο χρήστης στο json έχει ίδιο id με κάποιο στη βάση...
- Πως θα το κάναμε?

Πολλές Middleware Functions

- Στον landmarkController

```
exports.checkBodyId = (req, res, next) => {
  var id = req.body.id*1;
  var landmark = landmarks.find(element => element.id == id );
  if(landmark){
    return res.status(404).json({
      status:"fail",
      message: "exists already"
    });
  }
  next();
};
```

Πολλές Middleware Functions

- Στο landmarkRoutes:

```
9
0 //landmark Routes
1 routes.route('/')
2 .get(landmarkController.getAllLandmarks)
3 .post(landmarkController.checkBodyId, landmarkController.addLandmark)
4
```

To be continued...