

MVC

Εισαγωγή

- Είδαμε πως φτιάχνουμε ένα mongoose schema & mongoose model για την εφαρμογή μας.
- Είπαμε ότι τα Mongoose models είναι υπεύθυνα για την αναζήτηση, τη δημιουργία, την ενημέρωση και την αφαίρεση εγγράφων από τη βάση δεδομένων MongoDB.

→ αποθηκεύουμε τα models για resource

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure for '2_SMART TOURISM_MVC'. The 'models' folder is expanded, and 'landmarkModel.js' is selected. A red bracket highlights this file. The main editor window shows the code for 'landmarkModel.js' with the following content:

```
1  const mongoose = require('mongoose');
2
3  //schema
4  const landmarkSchema = new mongoose.Schema({
5      type:{
6          type: String,
7          required:[true, "A landmark must be landmark or musuem"]// specify error string
8      },
9      name: String,
10     description: String,
11     ratingsAverage: Number,
12     ratingsQuantity: Number
13 });
14 //model out of schema
15 const Landmark = mongoose.model("Landmark",landmarkSchema);
16
17 module.exports = Landmark;
```

```
JS landmarkModel.js X
models > JS landmarkModel.js > [?] <unknown>
1  const mongoose = require('mongoose');
2
3  //schema
4  const landmarkSchema = new mongoose.Schema({
5    type:{
6      type: String,
7      required:[true, "A landmark must be landmark or musuem"]// specify error string
8    },
9    name: String,
10   description: String,
11   ratingsAverage: Number,
12   ratingsQuantity: Number
13 });
14 //model out of schema
15 var Landmark = mongoose.model("Landmark",landmarkSchema);
16
17 module.exports = Landmark; //export model
```

import model

```
JS landmarkController.js X
controllers > JS landmarkController.js > getLandmarkById > getLandmarkById
1  const fs = require('fs');
2  const Landmark = require('../models/landmarkModel');
3
4  //landmarks routehandler
5  //we want to export ALL functions from this module
6  //we are going to put all these functions to the export object
7  exports.getAllLandmarks = (req,res)={
8
9    res.status(200).json({
10     status:"success"
11 });
12 }
13
14
15 exports.getLandmarkById = (req,res)={
16
17   //req.params stores all parameters that we define in the url
18   console.log(req.params)
19
20   //remember in js when we multiply a string that looks like a number
21   //with a number -> js converts string to number
22   var id = req.params.id*1;
23   res.status(200).json({
24     status:"success",
25     data: {
26       landmark
27     }
28 });
29
30 }
```

import exported functions (route handlers)

```
JS app.js > ...
1  //import express
2  const express= require('express');
3
4  const landmarkRouter= require('./routes/landmarkRoutes')
5  const userRouter= require('./routes/userRoutes')
6
7  // create app variable -> assigned result of calling express function
8  const app= express();
9
10 //MIDDLEWARES
11 app.use(express.json());
12
13 //here we specify the route(URL) for which we wish to use our middleware(landmarkRouter)
14 app.use('/api/v1/landmarks',landmarkRouter);
15 app.use('/api/v1/users',userRouter);
16
17 //export app to use it in the server.js file
18 module.exports = app;
```


import routers


```
landmarkRoutes.js X
app.js > JS landmarkRoutes.js > ...
1  //import express
2  const express= require('express');
3  //import landmarkController module
4  const landmarkController=require(`${__dirname}../controllers/landmarkController`);
5  //lets create a new router-> it is a middleware
6  const routes = express.Router();
7
8  //landmark Routes
9  routes.route('/')
10 .get(landmarkController.getAllLandmarks)
11 .post(landmarkController.addLandmark)
12
13
14 routes.route('/:id')
15 .get(landmarkController.getLandmarkById)
16 .patch(landmarkController.updateLandmarkById)
17 .delete(landmarkController.deleteLandmarkById)
```

Μέχρι τώρα...

- Άρα για κάθε resource έχουμε:

>  routes
resource Routes.js

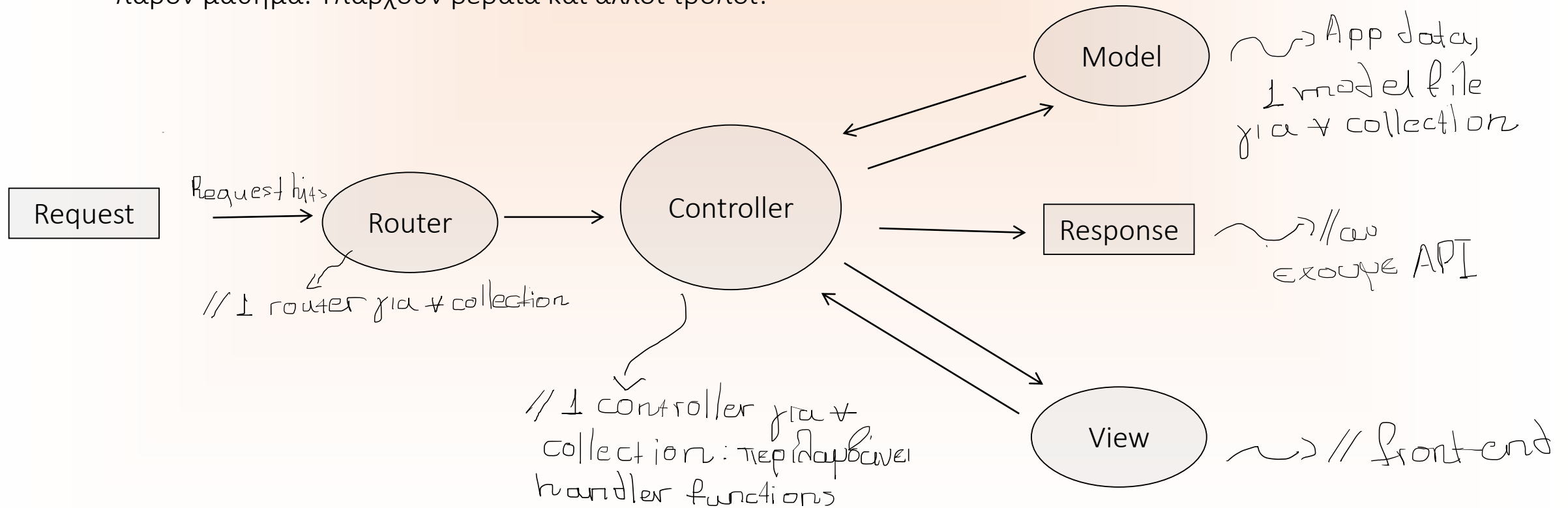
>  controllers
resource Controller.js

>  models
resource Model.js

Request → Response Cycle

MVC

Παρακάτω έχουμε έναν τρόπο υλοποίησης της αρχιτεκτονικής **mvc** που αξιοποιείται στα project στο παρόν μάθημα. Υπάρχουν βέβαια και άλλοι τρόποι!



To be continued...