

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
Τμήμα ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
"ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ"



Εργασία Μαθήματος ***Προηγμένος Προγραμματισμός Full Stack  
JavaScript Frameworks***

<b><i>Αριθμός εργασίας – Τίτλος εργασίας</i></b>	<b><i>Ατομική Απαλλακτική εργασία 8 μονάδων</i></b>
Όνομα φοιτητή	ΚΑΛΛΙΓΕΡΟΣ ΑΝΑΣΤΑΣΙΟΣ
Αρ. Μητρώου	mpsp2314
Ημερομηνία παράδοσης	7/2/2024



## Εκφώνηση εργασίας

### Θέμα:

Ο Δήμαρχος μιας πόλης, πλούσιας σε ιστορία και αξιοθέατα, αποφάσισε να εισέλθει στον κόσμο του Έξυπνου Τουρισμού. Με στόχο να προσφέρει στους κατοίκους και τους επισκέπτες μια ολοκληρωμένη εμπειρία, ανακοίνωσε την ανάπτυξη μιας ιστοσελίδας, μιας *mobile* εφαρμογής και μιας *desktop* εφαρμογής που θα επιτρέπει στους χρήστες (τουρίστες ή κατοίκους) να συμβάλλουν στον έξυπνο τουρισμό, αποθηκεύοντας σε μια βάση δεδομένων τα αγαπημένα τους μέρη και παρέχοντας πληροφορίες για τοπικά αξιοθέατα.

Στα πλαίσια αυτού του φιλόδοξου σχεδίου, καλούμαστε να αναπτύξουμε ένα API, το οποίο θα υποστηρίζει τις βασικές λειτουργίες διαχείρισης δεδομένων (*Create, Read, Update, Delete - CRUD*). Το API αυτό θα διασφαλίζει την ομαλή λειτουργία του εκάστοτε συστήματος, επιτρέποντας τη διαχείριση των χρηστών και των σημείων ενδιαφέροντος (*POI*).

### Λειτουργίες του API:

#### ▪ Εγγραφή Χρήστη:

Ένας νέος χρήστης εγγράφεται στο σύστημα, καταχωρώντας το όνομά του, την ηλικία και το email του. Το API ελέγχει αν υπάρχει ήδη χρήστης με το ίδιο email και αν ο χρήστης είναι κάτω των 18 ετών. Ο νέος χρήστης αποθηκεύεται στο collection "*Users*" της βάσης.

#### ▪ Προσθήκη Σημείου Ενδιαφέροντος (*POI*):



Υποστηρίζεται η αποθήκευση ενός νέου σημείου ενδιαφέροντος στη βάση, όπως ένα μουσείο, επιλέγοντας το όνομα, την κατηγορία, τη διεύθυνση και προαιρετικά μια φωτογραφία. Το POI αποθηκεύεται στο collection "POIs" της βάσης.

▪ **Ανάγνωση Πληροφοριών Χρήστη:**

Ο χρήστης του API μπορεί να διαβάσει τις πληροφορίες ενός χρήστη ή όλων. Με το API, μπορεί να αναζητήσει συγκεκριμένο χρήστη βάσει του email του και να βλέπει μόνο το όνομα του.

▪ **Ανακάλυψη Σημείων Ενδιαφέροντος (POI):**

Οι χρήστες μπορούν να δουν όλα τα σημεία ενδιαφέροντος στην πόλη. Με το API, μπορούν να αναζητήσουν ένα συγκεκριμένο POI βάσει παραμέτρων όπως η κατηγορία.

▪ **Ενημέρωση Πληροφοριών Χρήστη/ Σημείων ενδιαφέροντος**

▪ **Διαγραφή Χρήστη ή POI:**

Υποστηρίζεται η διαγραφή ενός χρήστη ή ενός POI

Πιο αναλυτικά, για την παρούσα εργασία, σας ανατίθεται η υλοποίηση ενός API που θα διαχειρίζεται δύο διαφορετικά collections σε μια MongoDB βάση δεδομένων.

**Collections:**

**1. Χρήστες (Users):**

ο Πεδία: Όνομα, Ηλικία, Email (υποχρεωτικά)

ο Το email αποθηκεύεται με μικρά γράμματα στη βάση, ανεξάρτητα από την είσοδο του χρήστη.

ο Έλεγχος για υπάρχον email κατά τη δημιουργία, με επιστροφή κατάλληλου status σε περίπτωση διπλοεγγραφής.

ο Έλεγχος ηλικίας <18 με χρήση middleware.

**2. Σημεία Ενδιαφέροντος (Points of Interest - POIs):**



ο Πεδία: Όνομα, Κατηγορία, Διεύθυνση, Γεωγραφικές Συντεταγμένες

ο Φωτογραφία (προαιρετικό)

ο Καμία περιοριστική απαίτηση ως προς τα πεδία.

### **Λειτουργίες API:**

#### **1. Create:**

ο Χρήστες: Αποθήκευση νέου χρήστη στη βάση με επιστροφή κατάλληλου status.

ο POI: Αποθήκευση νέου σημείου ενδιαφέροντος με επιστροφή κατάλληλου status.

#### **2. Read:**

ο Χρήστες:

- Όλων των εγγράφων των χρηστών.
- Ενός χρήστη βάσει του email του (εμφάνιση μόνο του ονόματος).

ο POI:

- Όλων των σημείων ενδιαφέροντος.
- Ενός σημείου ενδιαφέροντος βάσει παραμέτρου στο request.

#### **3. Update:**

ο Ενημέρωση ενός χρήστη βάσει του email του.

ο Ενημέρωση ενός σημείου ενδιαφέροντος βάσει του id του.

#### **4. Delete:**

ο Διαγραφή ενός χρήστη βάσει του email του.

ο Διαγραφή ενός σημείου ενδιαφέροντος βάσει του id του.



**Σημείωση:** Πέρα από τα υποχρεωτικά πεδία, μπορείτε να προσθέσετε ότι άλλο κρίνετε χρήσιμο για να αποθηκεύεται στη βάση στο κάθε *collections*.

**Επιπλέον:**

- Για τη χρήση του παρόντος API, **δεν** απαιτείται αυθεντικοποίηση (*authentication*)
- **Δεν** απαιτείται να δημιουργηθεί *front end*.
- Για τη βάση καλείστε να χρησιμοποιήσετε την **MongoDB** και τη **mongoose**
- Το *project* σας θα πρέπει να είναι οργανωμένο σε **MVC αρχιτεκτονική**. Μπορείτε να επιλέξετε ή τον τρόπο που είδαμε στα μαθήματα, ή οποιονδήποτε άλλο.
- Η εφαρμογή θα πρέπει να περιέχει **ένα *package.json*** αρχείο. Εκεί θα ορίζεται από που γίνεται η εκτέλεση της εφαρμογής. Στο *terminal* εκτελώντας την εντολή ***npm start*** θα πρέπει να γίνεται εκκίνηση της εφαρμογής.
- Να χρησιμοποιούνται **μεταβλητές περιβάλλοντος** στην εφαρμογή σας.
- Μπορείτε να αποστείλετε το *postman collection* (προαιρετικό)
- Μαζί με τον κώδικα καλείστε να παραδώσετε και ένα **έγγραφο** στο οποίο θα περιλαμβάνονται ως απαντήσεις στα παραπάνω ζητούμενα, τα *requests* που κάνετε, και τα *responses* που λαμβάνετε.

**Σημαντικό:**

- ✓ Σε περίπτωση που δεν καταφέρετε να υλοποιήσετε ολόκληρη την εργασία, μπορείτε να στείλετε



---

μέρος της (το οποίο βέβαια να είναι λειτουργικό) και θα βαθμολογηθείτε για αυτό!

- ✓ Το παραδοτέο σας θα είναι ένα αρχείο .zip το οποίο θα περιλαμβάνει τον κώδικα, το έγγραφο

και ότι άλλο κρίνετε απαραίτητο (πχ *postman collection*) το οποίο πρέπει να το ανεβάσετε στο

χώρο των εργασιών του *e-class* εντός της προθεσμίας.

Καλή επιτυχία!



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Επίδειξη της λύσης .....	8
---	--------------------------	---



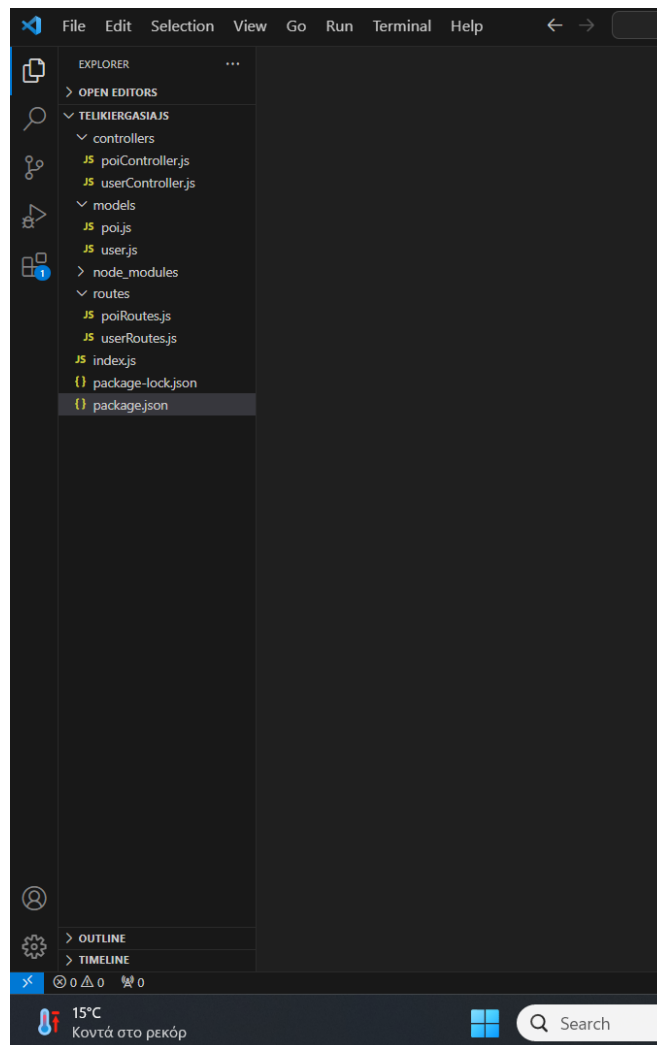
## 1 Επίδειξη της λύσης

Αρχικά το πρόγραμμα οργανώθηκε με **αρχιτεκτονική MVC**. Ο βασικός folder ονομάστηκε TELIKIERGASIAJS. Στην συνέχεια δημιουργήθηκαν οι controllers (poiController,userController), τα models(user.js,poi.js) ,routes (userRoutes.js,poiRoutes.js) και η index.js. Επιπλέον ως περιβάλλον ανάπτυξης χρησιμοποιήθηκε το Visual Studio Code. Επιπρόσθετα αξιοποιήθηκε το postman app και ως βάση έγινε χρήση της MongoDB και της mongoose. Δεν δημιουργήθηκε frond end όπως ακόμα περιέχει ένα package.json αρχείο. Εκεί ορίζεται από που γίνεται η εκτέλεση της εφαρμογής. Στο terminal εκτελώντας την εντολή npm start γίνεται η εκκίνηση της εφαρμογής. Ακολουθούν χαρακτηριστικά στιγμιότυπα (screenshots) από τα ζητούμενα, τα requests, και τα responses.





## TELIKIERGASIAJS:





## NPM START:

```
TERMINAL  PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE
PS C:\Users\tasso\Desktop\TelikiErgasiaJs> npm start

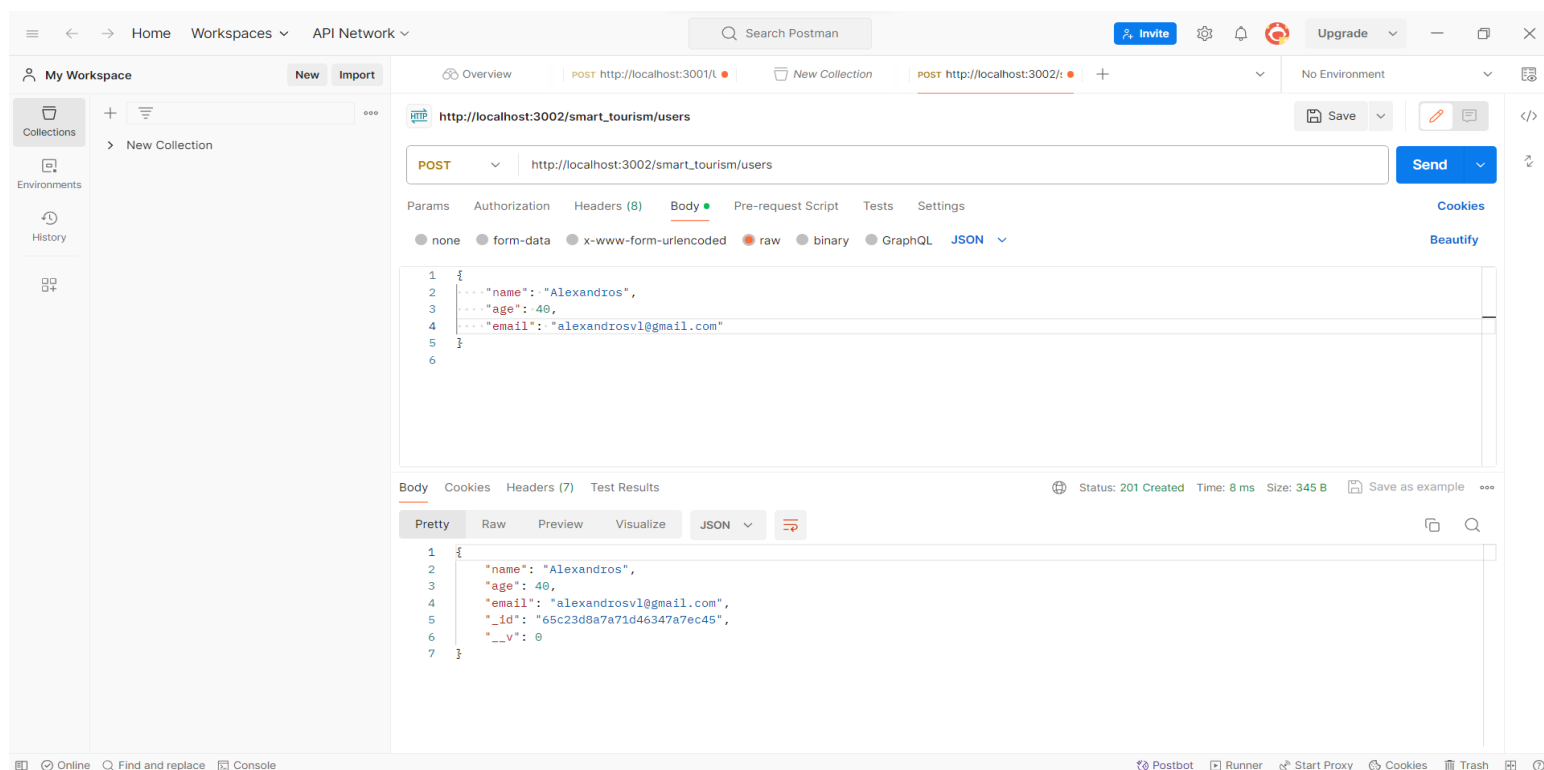
> telikiErgasiaJs@1.0.0 start
> node index.js

(node:12628) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:12628) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on port 3002
Connected to database
[]
```

## Λειτουργίες του API:

### Εγγραφή Χρήστη:

## ΠΡΟΣΘΗΚΗ ΧΡΗΣΤΗ





smart\_tourism.users

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{
  "name": "Giannis",
  "age": 25,
  "email": "giannisk@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId('65c23ad37a71d46347a7ec39'),
  "name": "Kostas",
  "age": 28,
  "email": "kostasla@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId('65c23cee7a71d46347a7ec42'),
  "name": "Panagiotis",
  "age": 35,
  "email": "panagiotism@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId('65c23d8a7a71d46347a7ec45'),
  "name": "Alexandros",
  "age": 48,
  "email": "alexandrosvl@gmail.com",
  "__v": 0
}
```

## ΕΛΕΓΧΟΣ ΥΠΑΡΞΗΣ EMAIL

POST http://localhost:3002/smart\_tourism/users

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
  "name": "Alexandros",
  "age": 47,
  "email": "alexandrosvl@gmail.com"
}
```

Status: 400 Bad Request Time: 24 ms Size: 305 B Save as example

```
{
  "error": "Το email χρησιμοποιείται ήδη."
}
```



## ΕΛΕΓΧΟΣ ΗΛΙΚΙΑΣ

The screenshot shows the Postman interface with a POST request to `http://localhost:3002/smart_tourism/users`. The request body is a JSON object:

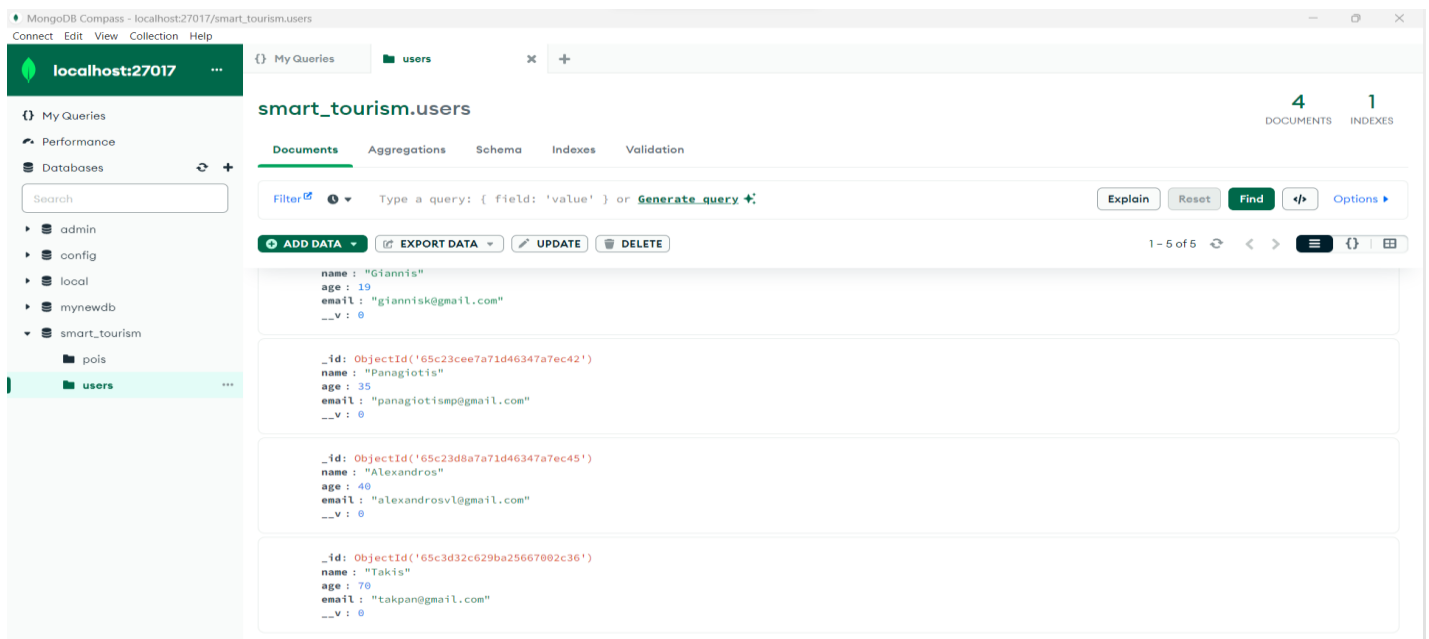
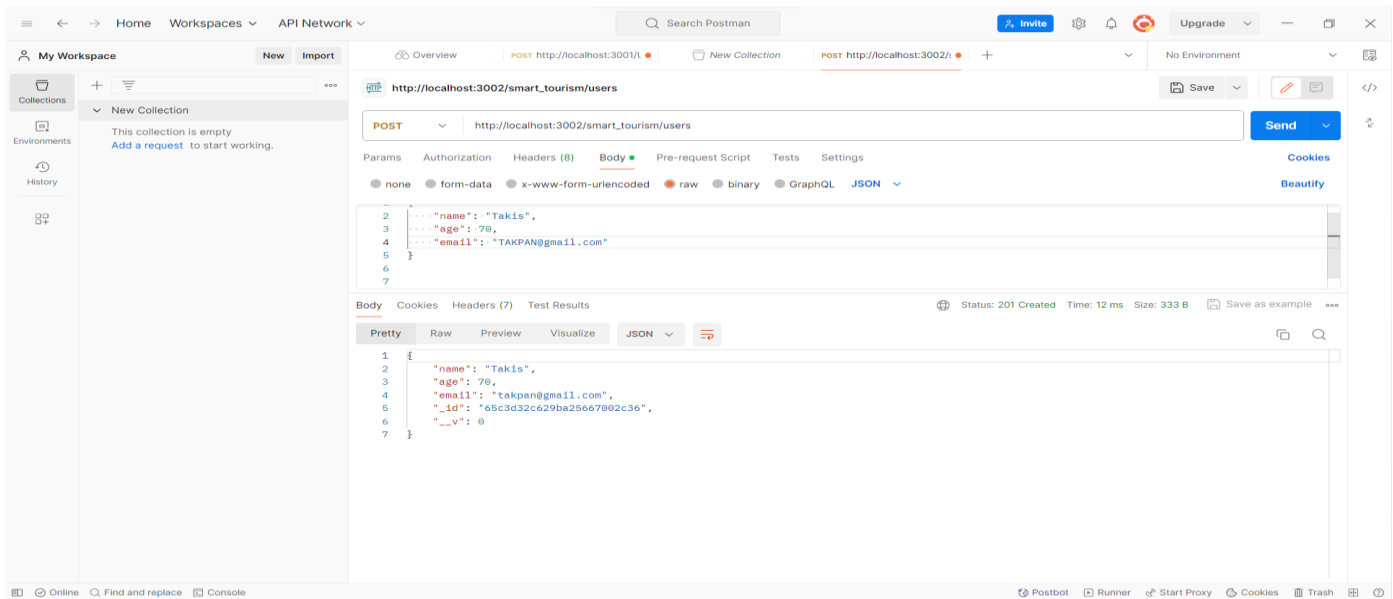
```
1 {
2   "name": "nikoss",
3   "age": 17,
4   "email": "nikosts@gmail.com"
5 }
6
```

The response status is **400 Bad Request** with a time of 10 ms and size of 345 B. The response body is:

```
1 {
2   "error": "Πρέπει να είστε άνω των 18 ετών για να εγγραφείτε."
3 }
```



# ΤΟ EMAIL ΝΑ ΑΠΟΘΗΚΕΥΕΤΑΙ ΜΕ ΜΙΚΡΑ ΓΡΑΜΜΑΤΑ ΣΤΗ ΒΑΣΗ





## ΠΡΟΣΘΗΚΗ ΣΗΜΕΙΟΥ ΕΝΔΙΑΦΕΡΟΝΤΟΣ (POI)

Postman interface showing a POST request to `http://localhost:3002/smart_tourism/pois`. The request body is a JSON object:

```
1 {
2   "name": "stratos",
3   "category": "palaska",
4   "address": "Αθήνα, Ελλάδα",
5   "coordinates": [37.9756, 23.6867],
6   "photo": "https://straos.com/photos/eidikesdynamis.jpg"
7 }
8
```

The response status is 201 Created. The response body is a JSON object:

```
1 {
2   "name": "stratos",
3   "category": "palaska",
4   "address": "Αθήνα, Ελλάδα",
5   "coordinates": [
6     37.9756,
7     23.6867
8   ],
9   "photo": "https://straos.com/photos/eidikesdynamis.jpg",
10   "_id": "65c247237a71d46347a7ec52",
11   "__v": 0
12 }
```

MongoDB Compass interface showing the `smart_tourism.pois` collection. The collection contains three documents:

```
category: "Αρχαιολογικός Χώρος"
address: "Αθήνα, Ελλάδα"
coordinates: Array (2)
photo: "https://arxaia.com/photos/akropoli.jpg"
__v: 0

_id: ObjectId('65c246d27a71d46347a7ec5b')
name: "Στάδιο Ολυμπιακού Συγκροτήματος Αθηνών"
category: "Αθλητικός Χώρος"
address: "Αθήνα, Ελλάδα"
coordinates: Array (2)
photo: "https://olympics.com/photos/olympic_stadium.jpg"
__v: 0

_id: ObjectId('65c247237a71d46347a7ec52')
name: "stratos"
category: "palaska"
address: "Αθήνα, Ελλάδα"
coordinates: Array (2)
photo: "https://straos.com/photos/eidikesdynamis.jpg"
__v: 0
```



## ΑΝΑΓΝΩΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΧΡΗΣΤΗ

Postman interface showing a GET request to `http://localhost:3002/smart_tourism/users`. The response is a JSON array of 4 user objects, each with `_id` and `name` fields.

```
1 [
2   {
3     "_id": "65c23a5a7a71d46347a7ec33",
4     "name": "Anastasis"
5   },
6   {
7     "_id": "65c23a997a71d46347a7ec36",
8     "name": "Giannis"
9   },
10  {
11    "_id": "65c23cee7a71d46347a7ec42",
12    "name": "Panagiotis"
13  },
14  {
15    "_id": "65c23d8a7a71d46347a7ec45",
16    "name": "Alexandros"
17  }
18 ]
```

Postman interface showing a GET request to `http://localhost:3002/smart_tourism/users/anastasiskal@gmail.com`. The response is a JSON object for the user with the specified email.

```
1 {
2   "_id": "65c23a5a7a71d46347a7ec33",
3   "name": "Anastasis"
4 }
```



## ΑΝΑΓΝΩΣΗ ΣΗΜΕΙΩΝ ΕΝΔΙΑΦΕΡΟΝΤΟΣ (POI)

Postman interface showing a GET request to `http://localhost:3002/smart_tourism/pois`. The response is a JSON array of two POI objects.

```
1 [
2   {
3     "_id": "65c2439e7a71d46347a7ec48",
4     "name": "Στρατός",
5     "category": "Στρατός",
6     "address": "Αθήνα, Ελλάδα",
7     "coordinates": [
8       37.9755,
9       23.734
10    ],
11    "photo": "https://polemikonautiko.com/photos/nautiko.jpg",
12    "__v": 0
13  },
14  {
15    "_id": "65c243d07a71d46347a7ec4a",
16    "name": "Παρθενόν",
17    "category": "Αρχαιολογικός Χώρος",
18    "address": "Αθήνα, Ελλάδα",
19    "coordinates": [
20      37.9715,
21      23.7251
22    ],
23    "photo": "https://istoria.com/photos/parthenonas.jpg",
24    "__v": 0
25  },
26  {
27    "_id": "65c244bd7a71d46347a7ec4c",
```

Postman interface showing a GET request to `http://localhost:3002/smart_tourism/pois/Στρατός`. The response is a JSON object for the 'Στρατός' POI.

```
1 {
2   "_id": "65c2439e7a71d46347a7ec48",
3   "name": "Ναυτικό",
4   "category": "Στρατός",
5   "address": "Αθήνα, Ελλάδα",
6   "coordinates": [
7     37.9755,
8     23.734
9   ],
10  "photo": "https://polemikonautiko.com/photos/nautiko.jpg",
11  "__v": 0
12 }
```





# ΕΝΗΜΕΡΩΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΝΟΣ ΧΡΗΣΤΗ ΒΑΣΕΙ EMAIL

MongoDB Compass - localhost:27017/smart\_tourism.users

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases Search

admin config local mynewdb smart\_tourism

smart\_tourism.users

3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{
  "name": "Giannis",
  "age": 25,
  "email": "giannisk@gmail.com",
  "_id": "65c23d37a71d46347a7ec39"
}
```

```
{
  "name": "Kostas",
  "age": 28,
  "email": "kostasla@gmail.com",
  "_id": "65c23d37a71d46347a7ec42"
}
```

```
{
  "name": "Panagiotis",
  "age": 35,
  "email": "panagiotismp@gmail.com",
  "_id": "65c23d37a71d46347a7ec45"
}
```

```
{
  "name": "Alexandros",
  "age": 40,
  "email": "alexandrosvl@gmail.com",
  "_id": "65c23d37a71d46347a7ec46"
}
```

Home Workspaces API Network

Search Postman

My Workspace

New Import

Overview POST http://localhost:3001/ New Collection PUT http://localhost:3002/sr

http://localhost:3002/smart\_tourism/users/giannisk@gmail.com

PUT http://localhost:3002/smart\_tourism/users/giannisk@gmail.com

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Giannis",
3   "age": 19,
4   "email": "giannisk@gmail.com"
5 }
6
```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 20 ms Size: 332 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "65c23a997a71d46347a7ec36",
3   "name": "Giannis",
4   "age": 19,
5   "email": "giannisk@gmail.com",
6   "__v": 0
7 }
```



MongoDB Compass - localhost:27017/smart\_tourism.users

Connect Edit View Collection Help

localhost:27017

My Queries

Performance

Databases

Search

admin

config

local

mynewdb

smart\_tourism

pois

users

smart\_tourism.users

3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{
  "_id": ObjectId("65c23a5a7a71d46347a7ec33"),
  "name": "Anastasis",
  "age": 22,
  "email": "anastasiskal@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c23a997a71d46347a7ec36"),
  "name": "Giannis",
  "age": 19,
  "email": "giannisk@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c23ad37a71d46347a7ec39"),
  "name": "Kostas",
  "age": 28,
  "email": "kostasla@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c23cee7a71d46347a7ec42"),
  "name": "Panagiotis",
  "age": 35,
  "email": "panagiotismp@gmail.com",
  "v": 0
}
```



# ΕΝΗΜΕΡΩΣΗ ΕΝΟΣ ΣΗΜΕΙΟΥ ΕΝΔΙΑΦΕΡΟΝΤΟΣ ΒΑΣΕΙ ΤΟΥ ID ΤΟΥ

The screenshot shows the MongoDB Compass interface for the 'smart\_tourism.pois' collection. The left sidebar lists the databases and collections. The main area displays three documents in the 'Documents' tab. Each document contains fields for \_id, name, category, address, coordinates, and photo.

```
{ "_id": ObjectId('65c244bd7a71d46347a7ec4c'), "name": "Μουσείο Λούβρου", "category": "Μουσείο", "address": "Παρίσι, Γαλλία", "coordinates": Array (2), "photo": "https://galia.com/photos/louvre.jpg", "__v": 0 }
```

```
{ "_id": ObjectId('65c245057a71d46347a7ec4e'), "name": "Ακρόπολη", "category": "Αρχαιολογικός Χώρος", "address": "Αθήνα, Ελλάδα", "coordinates": Array (2), "photo": "https://arxaiia.com/photos/akropolis.jpg", "__v": 0 }
```

```
{ "_id": ObjectId('65c247237a71d46347a7ec52'), "name": "stratos", "category": "palaska", "address": "Αθήνα, Ελλάδα", "coordinates": Array (2), "photo": "https://straos.com/photos/eidikesnameis.jpg", "__v": 0 }
```

The screenshot shows the Postman interface with a PUT request to update a document in the 'smart\_tourism.pois' collection. The request body is a JSON object with the following fields: name, category, address, coordinates, and photo. The response status is 200 OK.

```
PUT http://localhost:3002/smart_tourism/pois/65c243d07a71d46347a7ec4a
```

```
{ "name": "Παρθενώνας", "category": "Αρχαιολογικός Χώρος", "address": "Αθήνα, Ελλάδα", "coordinates": [ 37.9715, 23.7251 ], "photo": "https://history.com/photos/paithenon.jpg" }
```

```
{ "_id": "65c243d07a71d46347a7ec4a", "name": "Παρθενώνας", "category": "Αρχαιολογικός Χώρος", "address": "Αθήνα, Ελλάδα", "coordinates": [ 37.9715, 23.7251 ], "photo": "https://history.com/photos/paithenon.jpg", "__v": 0 }
```



MongoDB Compass - localhost:27017/smart\_tourism.pois

Connect Edit View Collection Help

localhost:27017

My Queries

pois

smart\_tourism.pois

5 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN RESET FIND Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{
  "_id": ObjectId("65c2439e7a71d46347a7ec48"),
  "name": "Ναυτικό",
  "category": "Στρατός",
  "address": "Αθήνα, Ελλάδα",
  "coordinates": Array(2),
  "photo": "https://polemikonautiko.com/photos/nautiko.jpg",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c243d07a71d46347a7ec4a"),
  "name": "Παρθενώνας",
  "category": "Αρχαιολογικός Χώρος",
  "address": "Αθήνα, Ελλάδα",
  "coordinates": Array(2),
  "photo": "https://history.com/photos/parthenon.jpg",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c244bd7a71d46347a7ec4c"),
  "name": "Μουσείο Λούβρου",
  "category": "Μουσείο",
  "address": "Παρίσι, Γαλλία",
  "coordinates": Array(2),
  "photo": "https://gallia.com/photos/louvre.jpg",
  "__v": 0
}
```



## ΔΙΑΓΡΑΦΗ ΧΡΗΣΤΗ ΒΑΣΕΙ ΤΟΥ EMAIL

MongoDB Compass - localhost:27017/smart\_tourism.users

Connect Edit View Collection Help

localhost:27017

My Queries users

smart\_tourism.users

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{
  "name": "Giannis",
  "age": 25,
  "email": "giannisk@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId('65c23ad37a71d46347a7ec39'),
  "name": "Kostas",
  "age": 28,
  "email": "kostasla@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId('65c23cee7a71d46347a7ec42'),
  "name": "Panagiotis",
  "age": 35,
  "email": "panagiotismp@gmail.com",
  "__v": 0
}
```

```
{
  "_id": ObjectId('65c23d8a7a71d46347a7ec45'),
  "name": "Alexandros",
  "age": 40,
  "email": "alexandrosvl@gmail.com",
  "__v": 0
}
```

Home Workspaces API Network

Search Postman

My Workspace

Overview POST http://localhost:3001/l DEL http://localhost:3002/sr

http://localhost:3002/smart\_tourism/users/kostasla@gmail.com

DELETE http://localhost:3002/smart\_tourism/users/kostasla@gmail.com

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Παρθενώνας",
3   "category": "Αρχαιολογικός Χώρος",
4   "address": "Αθήνα, Ελλάδα",
5   "coordinates": [37.9715, 23.7251],
6   "photo": "https://istoria.com/photos/parthenonas.jpg"
}
```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 10 ms Size: 275 B Save as example

```
1 {
2   "message": "User deleted successfully."
}
```



MongoDB Compass - localhost:27017/smart\_tourism.users

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases

Search

admin config local mynewdb smart\_tourism

pois users

smart\_tourism.users

3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 4 of 4

```
{ "_id": ObjectId("65c23a5a7a71d46347a7ec33"),  
  "name": "Anastasis",  
  "age": 22,  
  "email": "anastasiskal@gmail.com",  
  "__v": 0 }  
  
{ "_id": ObjectId("65c23a997a71d46347a7ec36"),  
  "name": "Giannis",  
  "age": 19,  
  "email": "giannisk@gmail.com",  
  "__v": 0 }  
  
{ "_id": ObjectId("65c23cee7a71d46347a7ec42"),  
  "name": "Panagiotis",  
  "age": 35,  
  "email": "panagiotismp@gmail.com",  
  "__v": 0 }  
  
{ "_id": ObjectId("65c23d8a7a71d46347a7ec45"),  
  "name": "Alexandros",  
  "age": 40,  
  "email": "alexandrosvl@gmail.com",  
  "__v": 0 }
```



## ΔΙΑΓΡΑΦΗ ΡΟΙ ΒΑΣΕΙ ΤΟΥ ID ΤΟΥ

The screenshot shows the MongoDB Compass interface for the 'smart\_tourism.pois' collection. The left sidebar shows the database structure with 'pois' selected. The main area displays 5 documents in the 'Documents' tab. The documents are:

- `{ "_id": "65c244bd7a71d46347a7ec4c", "name": "Μουσείο Λούβρου", "category": "Μουσείο", "address": "Παρίσι, Γαλλία", "coordinates": [37.568, 2.312], "photo": "https://gallia.com/photos/louvre.jpg" }`
- `{ "_id": "65c245957a71d46347a7ec4e", "name": "Ακρόπολη", "category": "Αρχαιολογικός Χώρος", "address": "Αθήνα, Ελλάδα", "coordinates": [37.971, 23.725], "photo": "https://arxaiia.com/photos/akropolis.jpg" }`
- `{ "_id": "65c247237a71d46347a7ec52", "name": "Stratos", "category": "παλaska", "address": "Αθήνα, Ελλάδα", "coordinates": [37.971, 23.725], "photo": "https://straos.com/photos/eidikesdynameis.jpg" }`

The screenshot shows a Postman interface with a DELETE request to `http://localhost:3002/smart_tourism/pois/65c246d27a71d46347a7ec50`. The request body is empty. The response is a JSON object:

```
{  "message": "POI deleted successfully."}
```



MongoDB Compass - localhost:27017/smart\_tourism.pois

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases

Search

admin config local mynewdb smart\_tourism

pois users

smart\_tourism.pois

1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 6 of 6

```
{
  "photo": "https://polemikonautiko.com/photos/nautiko.jpg",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c243d07a71d46347a7ec4a"),
  "name": "Παρθενώνας",
  "category": "Αρχαιολογικός Χώρος",
  "address": "Αθήνα, Ελλάδα",
  "coordinates": Array(2),
  "photo": "https://istoria.com/photos/parthenonas.jpg",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c244bd7a71d46347a7ec4c"),
  "name": "Μουσείο Λούβρου",
  "category": "Μουσείο",
  "address": "Παρίσι, Γαλλία",
  "coordinates": Array(2),
  "photo": "https://gallia.com/photos/louvre.jpg",
  "__v": 0
}
```

```
{
  "_id": ObjectId("65c245057a71d46347a7ec4e"),
  "name": "Ακρόπολη",
  "category": "Αρχαιολογικός Χώρος",
  "address": "Αθήνα, Ελλάδα",
  "coordinates": Array(2),
  "__v": 0
}
```