

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ - ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ  
ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

ADVANCED PROGRAMMING IN FULL STACK JAVASCRIPT FRAMEWORKS

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ API

---

Ομάδα ανάπτυξης:  
Ζάρτηλας Βασίλειος – ΜΠΣΠ21015

*Πειραιάς, Φεβρουάριος 2022*

# Περιεχόμενα

1 Εισαγωγή	3
2 Points	4
2.1 Create	4
2.2 Read	5
Με το όνομα (χρήση της getPointByName)	5
Με το ID (χρήση της getPointById)	5
Όλα τα σημεία (χρήση της getAllPoints)	6
2.3 Update	6
Με το όνομα (χρήση της updatePointByName)	6
Με το ID (χρήση της updatePointById)	7
2.4 Delete	7
Με το ID (χρήση της deletePoint)	7
3 Users	9
3.1 Create	9
3.2 Read	11
Με το email (χρήση της getUserByEmail)	11
Με το Id (χρήση της getUserById)	12
Όλοι οι χρήστες (χρήση της getAllUsers)	12
3.3 Update	13
Με το email (χρήση της updateUserByEmail)	13
3.4 Delete	13
Με το ID (χρήση της deleteUser)	13
4 Βιβλιοθήκες - Εργαλεία	15
5 Βιβλιογραφία - Πηγές	16

# 1 Εισαγωγή

---

- MVC: Το project είναι οργανωμένο σε mvc αρχιτεκτονική αλλά λείπει ο φάκελος view εφόσον δεν έχει φτιαχτεί User Interface.
- Η εφαρμογή περιέχει το αρχείο package.json και ορίζεται ότι η εφαρμογή θα ξεκινάει από το index.js:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "start": "nodemon index.js"  
},
```

- Στο terminal εκτελώντας την εντολή npm start ξεκινάει η εφαρμογή και ακούσει στο port:8080:

```
zartilas@zartilas:~/Documents/Github/fullstack/Points-of-interest$ npm start  
> points-of-interest@1.0.0 start  
> nodemon index.js  
  
[nodemon] 2.0.15  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,json  
[nodemon] starting `node index.js`  
(node:91919) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency  
(Use `node --trace-warnings ...` to show where the warning was created)  
info: App is listening on url http://localhost:8080  
info: MongoDB connected successfully...
```

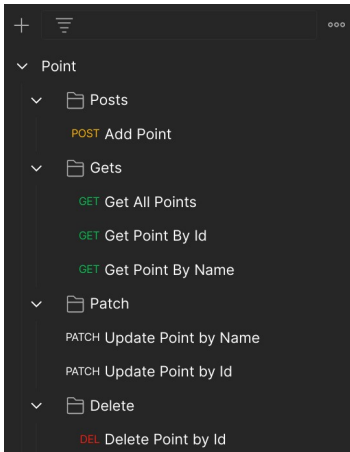
- Στην εφαρμογή περιέχονται μεταβλητές περιβάλλοντος (.env):

```
1  NODE_ENV=development  
2  
3  #express server config  
4  
5  PORT=8080  
6  HOST=localhost  
7  HOST_URL=http://localhost:8080
```

- Στην αρχή του project δημιουργήθηκε μία απλή σελίδα για την καλύτερη κατανόηση του μαθήματος αλλά μετά διαγράφηκε. Υπάρχει σχετικό βίντεο στα αρχεία.
- Το project υπάρχει και στο github: <https://github.com/zartilas/Points-of-interest>
- Postman: <https://www.postman.com/zartilas/workspace/points-of-interest>

## 2 Points

Στην πιο κάτω φωτογραφία βλέπουμε τα post, get, patch και delete στα collections του postman:

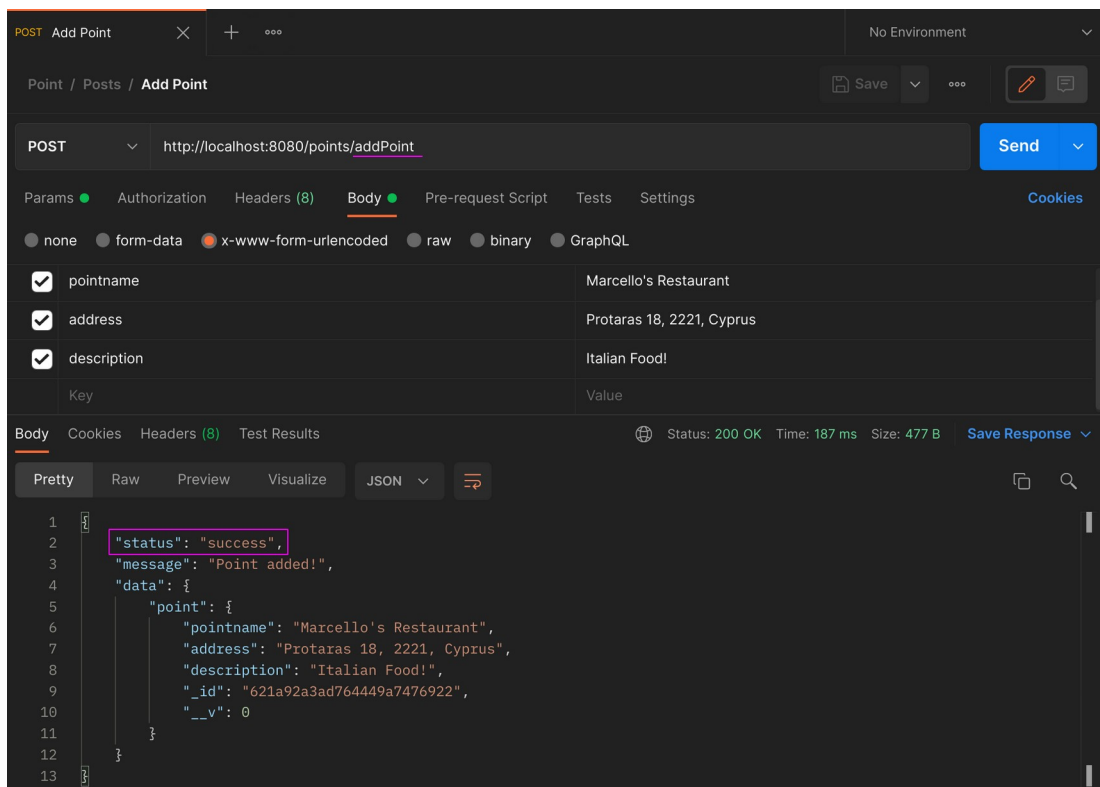


### 2.1 Create

Για τα σημεία δημιουργούνται τα εξής πεδία:

1. Όνομα.
2. Διεύθυνση.
3. Περιγραφή.

Η δημιουργία του σημείου γίνεται μέσω της **addPoint**:

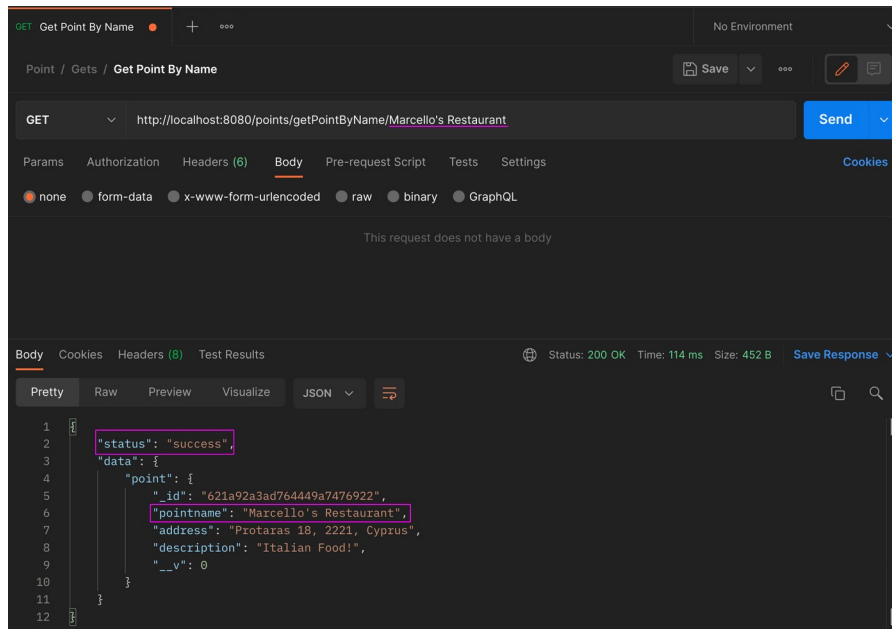


## 2.2 Read

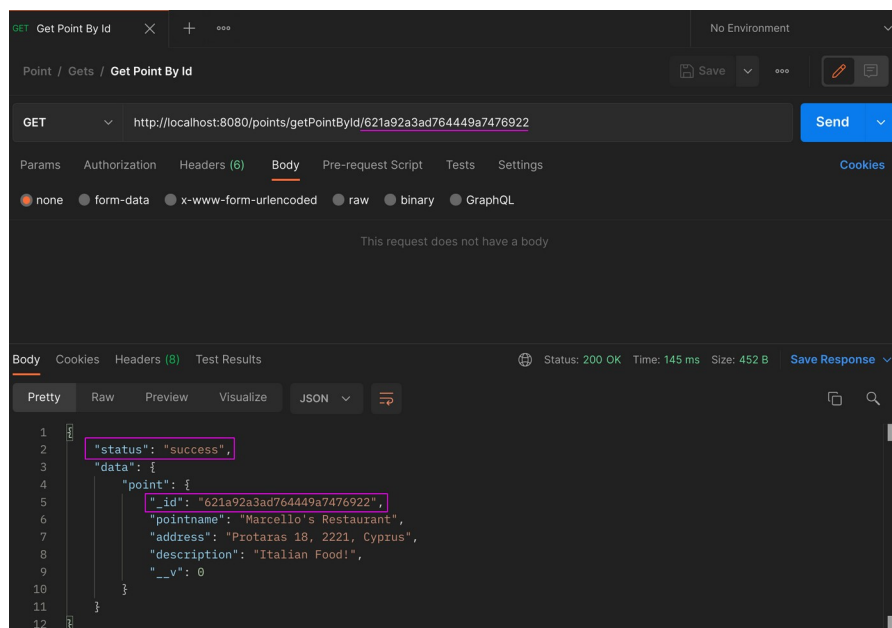
Τα σημεία ενδιαφέροντος διαβάζονται με τους εξής τρόπους:

1. Με το όνομα.
2. Με το ID.
3. Όλα τα σημεία μαζί.

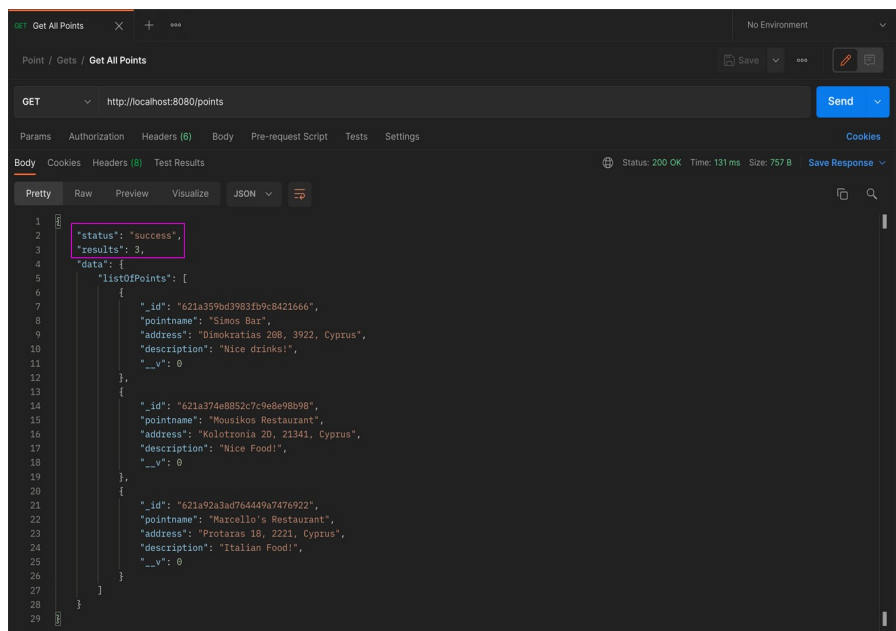
### Με το όνομα (χρήση της getPointByName)



### Με το ID (χρήση της getPointById)



## Όλα τα σημεία (χρήση της getAllPoints)

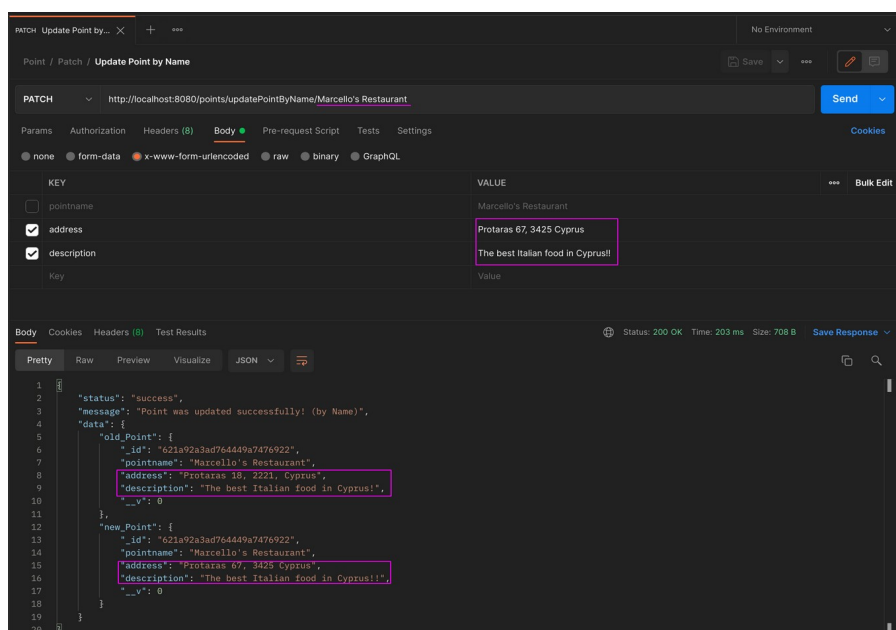


## 2.3 Update

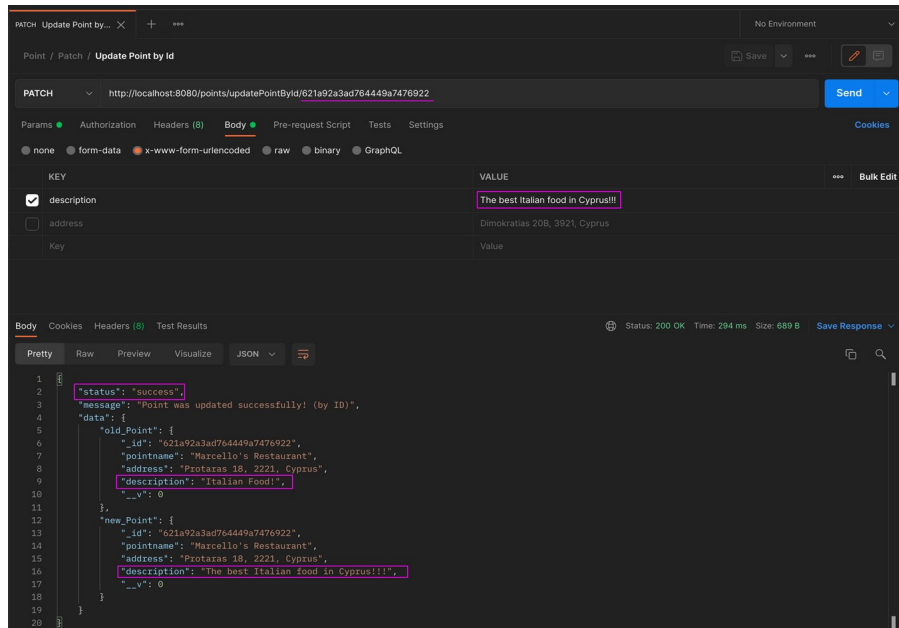
Τα σημεία ενδιαφέροντος ενημερώνονται με τους εξής τρόπους:

1. Με το όνομα.
2. Με το ID.

## Με το όνομα (χρήση της updatePointByName)



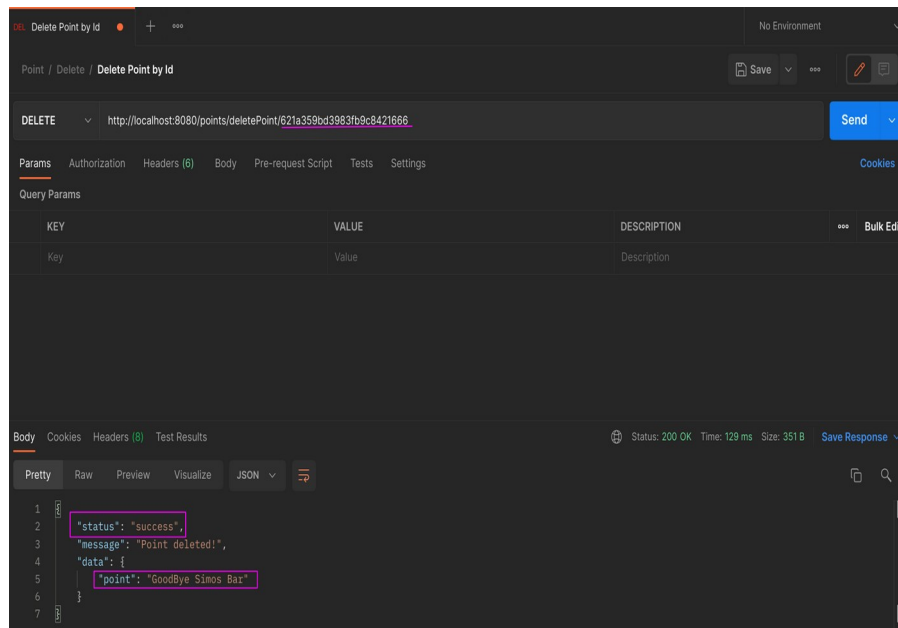
## Με το ID (χρήση της updatePointById)



## 2.4 Delete

Τα σημεία ενδιαφέροντος διαγράφονται μέσω του ID.

## Με το ID (χρήση της deletePoint)



Όταν ξανά γίνει `getAllPoints` βλέπουμε τις διαφορές και το διαγραφμένο σημείο δεν υπάρχει πλέον:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://localhost:8080/points`
- Body Type:** x-www-form-urlencoded
- Status:** 200 OK
- Time:** 154 ms
- Size:** 638 B

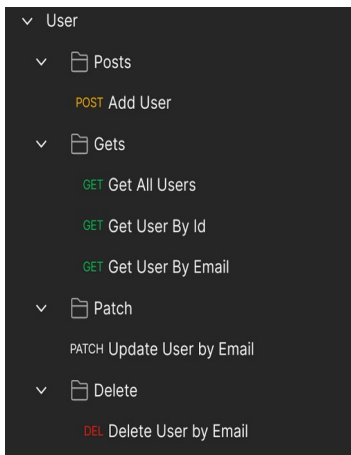
The response body is displayed in JSON format:

```
{
  "status": "success",
  "results": 2,
  "data": {
    "listOfPoints": [
      {
        "_id": "621a374e8852c7c9e8e98b98",
        "pointname": "Mousikos Restaurant",
        "address": "Kolotronia 2D, 21341, Cyprus",
        "description": "Nice Food!",
        "__v": 0
      },
      {
        "_id": "621a92a3ad764449a7476922",
        "pointname": "Marcello's Restaurant",
        "address": "Protaras 67, 3425 Cyprus",
        "description": "The best Italian food in Cyprus!!",
        "__v": 0
      }
    ]
  }
}
```



## 3 Users

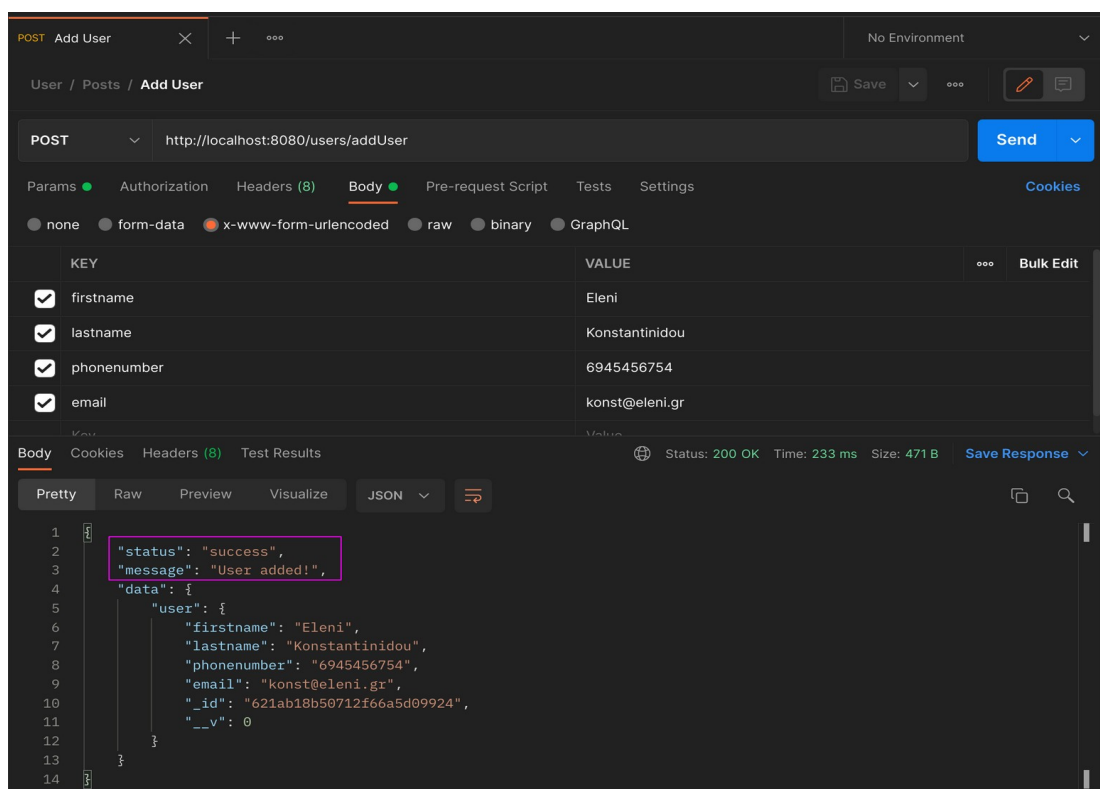
Στην πιο κάτω φωτογραφία βλέπουμε τα post, get, patch και delete στα collections του postman:



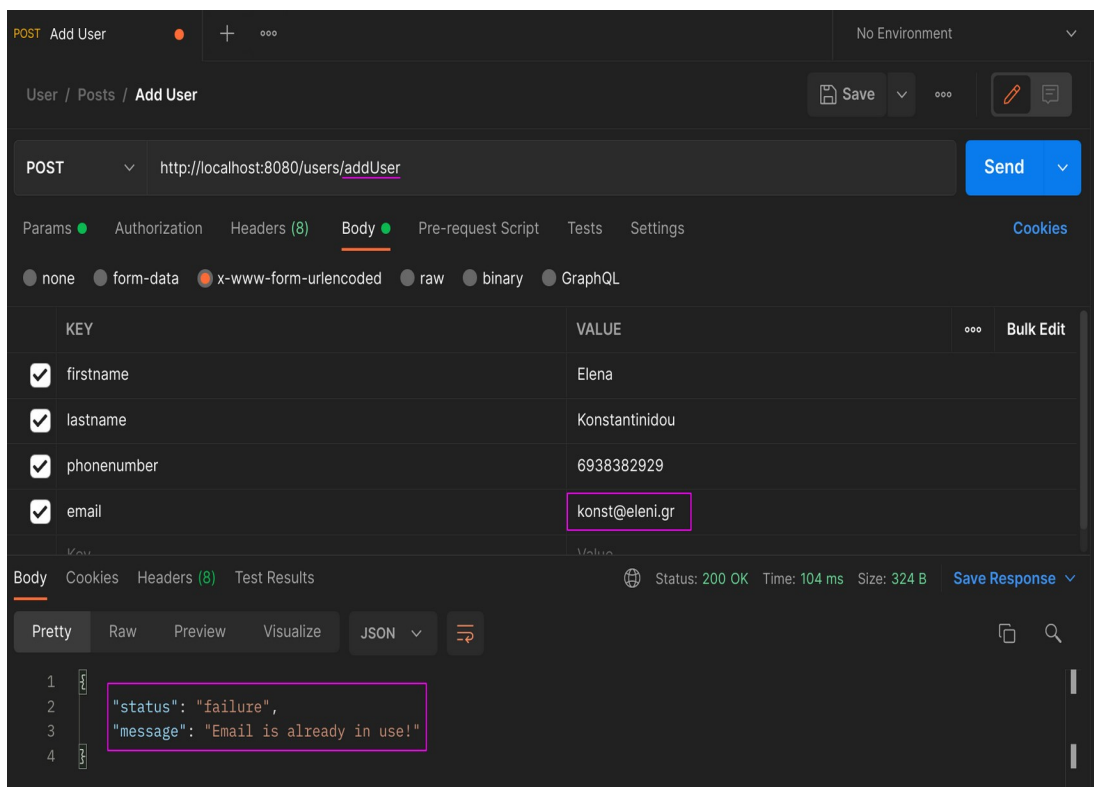
### 3.1 Create

Για τον χρήστη δημιουργούνται τα εξής πεδία:

1. Όνομα.
2. Επίθετο.
3. Αριθμός τηλεφώνου.
4. Ηλεκτρονικό ταχυδρομείο.



Στην πιο κάτω φωτογραφία βλέπουμε ότι αν επιθυμούμε να δημιουργήσουμε την πρώτη ξαδέρφη της Ελένης την Έλενα Κωνσταντινίδου κι εφόσον αλλάξουμε το όνομα και το τηλέφωνο και κρατήσουμε το ίδιο email τότε δεν μπορούμε να την καταχωρήσουμε!!



**Σημείωση:** Ο έλεγχος για την διπλότυπη εμφάνιση του email γίνεται μέσω της Mongoose στο Model του user και όχι μέσω κάποιου middleware ( έχει γραφτεί κώδικας για τον έλεγχο του email μέσω middleware αλλά δεν χρησιμοποιήθηκε (./middleware/checkEmail.js) ). Μέσα στο **userSchema** (./models/user.js) το email έχει δηλωθεί ως **unique** έτσι με αυτήν την δήλωση δεν επιτρέπει η mongoose να γραφτεί email που υπάρχει ήδη στην βάση.

```
email: {  
  type: String,  
  unique: true,  
  minlength: 5,  
  maxlength: 50,  
  lowercase: true  
};
```

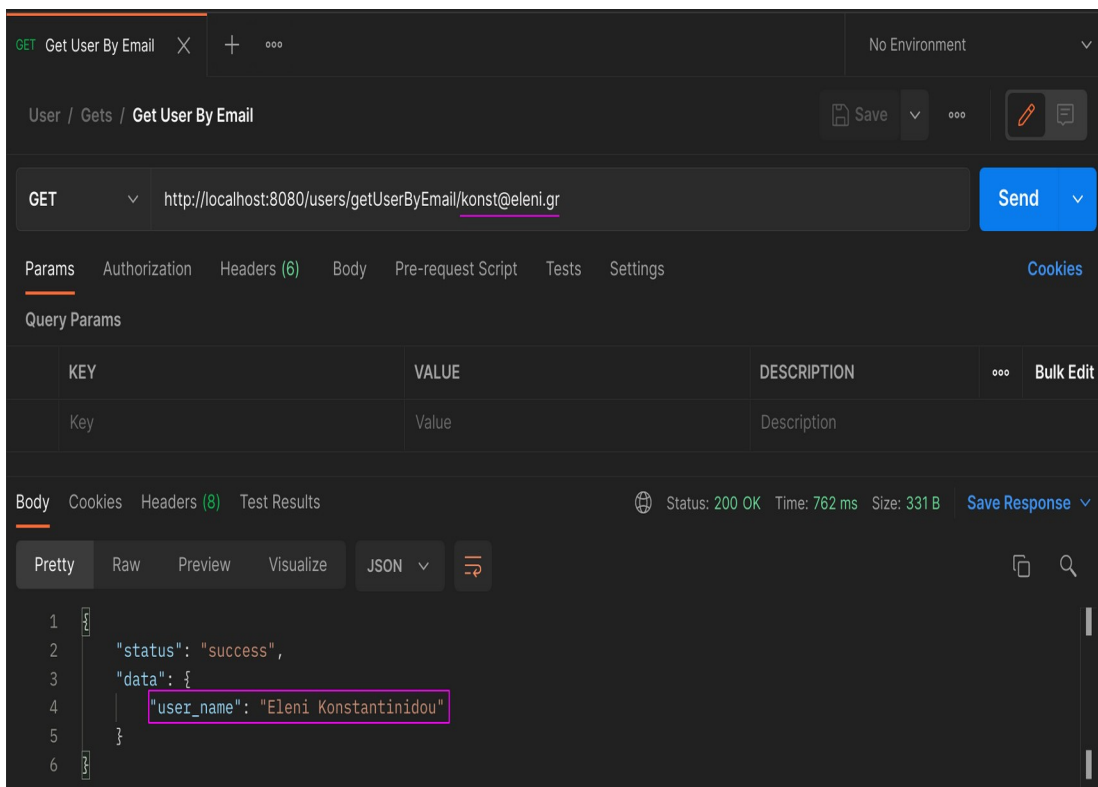
## 3.2 Read

Τα σημεία ενδιαφέροντος διαβάζονται με τους εξής τρόπους:

1. Με το email.
2. Με το ID.
3. Όλα οι χρήστες μαζί.

### Με το email (χρήση της getUserByEmail)

Στην πιο κάτω φωτογραφία βλέπουμε ότι με την αναζήτηση του χρήστη μέσω του email επιστρέφει μόνο το όνομα του χρήστη:



## Με το Id (χρήση της getUserById)

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/users/getUserById/621ab18b50712f66a5d09924`. The response status is 200 OK. The response body is a JSON object with the following structure:

```
{
  "status": "success",
  "data": {
    "user": {
      "_id": "621ab18b50712f66a5d09924",
      "firstname": "Eleni",
      "lastname": "Konstantinidou",
      "phonenumber": "6945456754",
      "email": "konst@eleni.gr",
      "__v": 0
    }
  }
}
```

## Όλοι οι χρήστες (χρήση της getAllUsers)

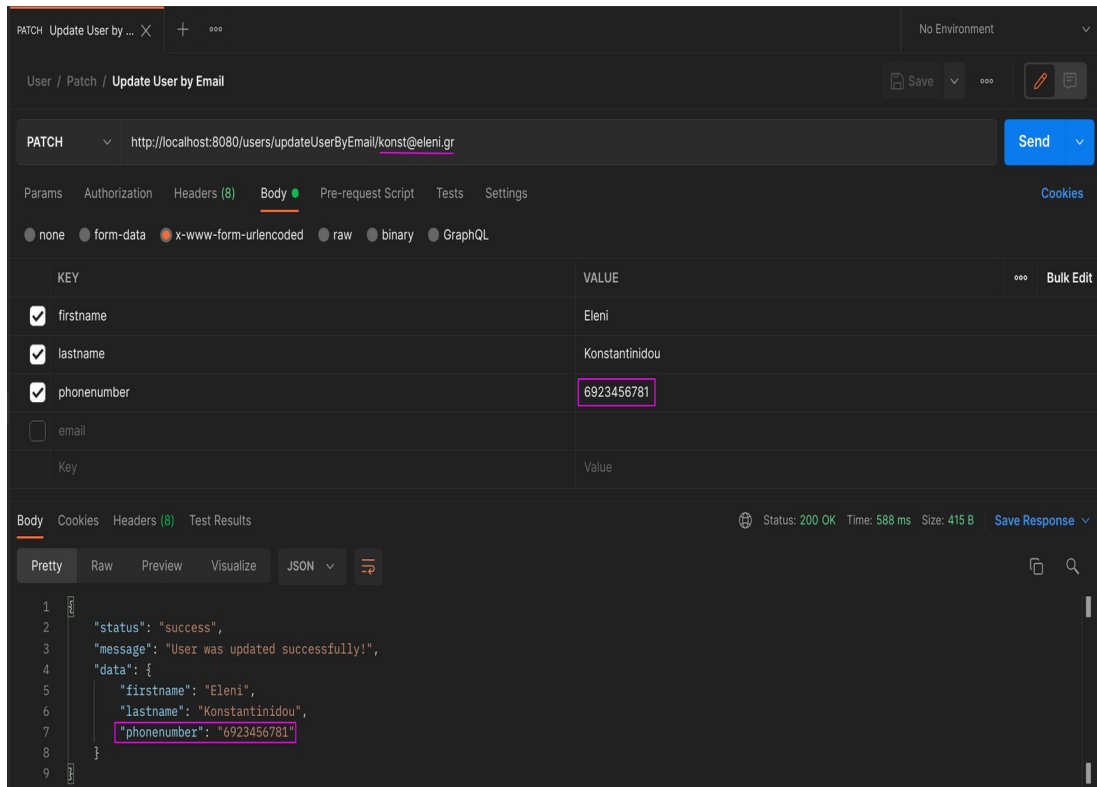
The screenshot shows a Postman interface for a GET request to `http://localhost:8080/users`. The response status is 200 OK. The response body is a JSON object with the following structure:

```
{
  "status": "success",
  "results": 3,
  "data": {
    "userlist": [
      {
        "_id": "621943d84113011d66e4481c",
        "firstname": "Vasilis",
        "lastname": "Zartilas",
        "phonenumber": "6946373838",
        "email": "vzartilas@pm.me",
        "__v": 0
      },
      {
        "_id": "621a58a7437bbfddb5994072",
        "firstname": "Michaela",
        "lastname": "Matsagkou",
        "phonenumber": "6988756452",
        "email": "michaela@matsa.com",
        "__v": 0
      },
      {
        "_id": "621ab18b50712f66a5d09924",
        "firstname": "Eleni",
        "lastname": "Konstantinidou",
        "phonenumber": "6945456754",
        "email": "konst@eleni.gr",
        "__v": 0
      }
    ]
  }
}
```

## 3.3 Update

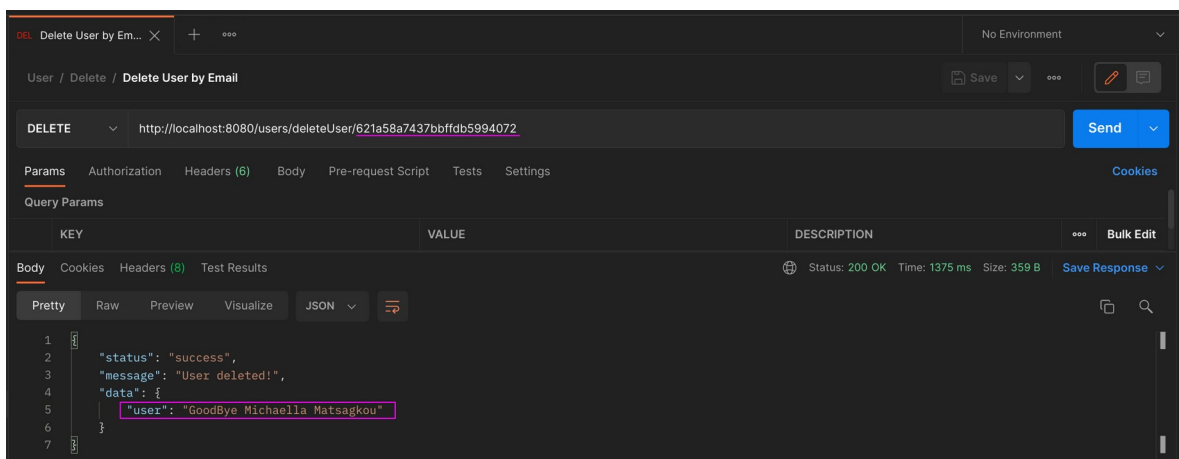
Οι χρήστες ενημερώνονται μέσω του email:

**Με το email (χρήση της updateUserByEmail)**

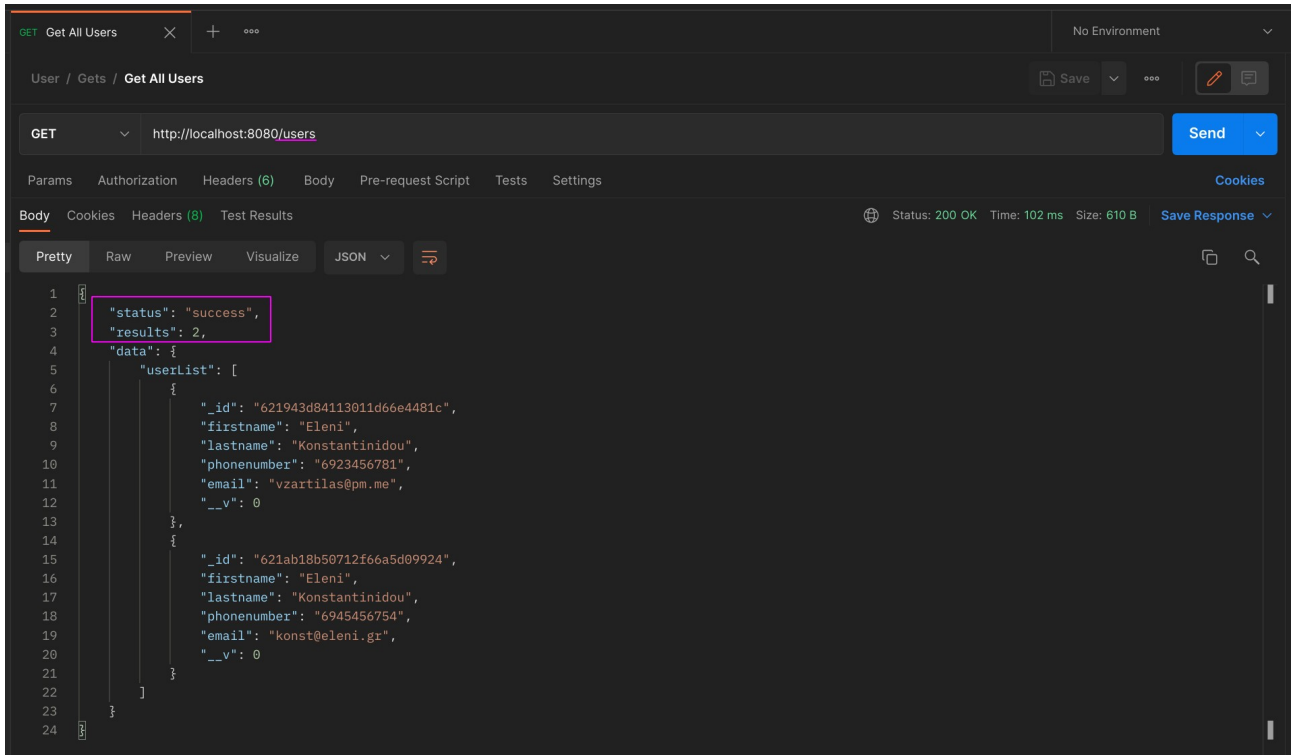


## 3.4 Delete

**Με το ID (χρήση της deleteUser)**



**Όταν ξανά γίνει `getAllUsers` βλέπουμε τις διαφορές και ο διαγραφμένος χρήστης δεν υπάρχει πλέον:**



## 4 Βιβλιοθήκες - Εργαλεία

---

Όνομα	Έκδοση
Body parser	1.19.2
Cors	2.8.5
Dotenv	16.0.0
Express	4.17.3
Express async errors	3.1.1
Joi	13.1.0
Joi-objectid	2.0.0
Mongoose	6.2.2
Winston	2.4.0
Nodemon	2.0.15
Postman	9.13.2
MongoDB compass	1.30.1
MongoDB	5.0.6

## 5 Βιβλιογραφία - Πηγές

---

1. Σημειώσεις και κώδικας του μαθήματος
2. <https://dev.to/dalalrohit/how-to-connect-to-mongodb-atlas-using-node-js-k9i>
3. <https://www.youtube.com/watch?v=RU9Vzu9uDYU>
4. <https://mongoosejs.com/docs/guide.html>
5. <https://stackoverflow.com/a/16883572/13297070>
6. <https://nodejs.org/en/docs/>