



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

ομάδα 72 παραδοτέο 2

REST API INSTRUCTOR

AUTHRESTCONTROLLER

POST : /api/auth/signup:

Είναι υπεύθυνο για την εγγραφή (registration) νέων χρηστών.

Δέχεται ένα αίτημα με τα στοιχεία του νέου χρήστη (όνομα χρήστη, κωδικό πρόσβασης, ρόλο, κ.λπ.) και δημιουργεί έναν νέο χρήστη στη βάση δεδομένων.

Επιστρέφει ένα μήνυμα επιτυχίας ή αποτυχίας.

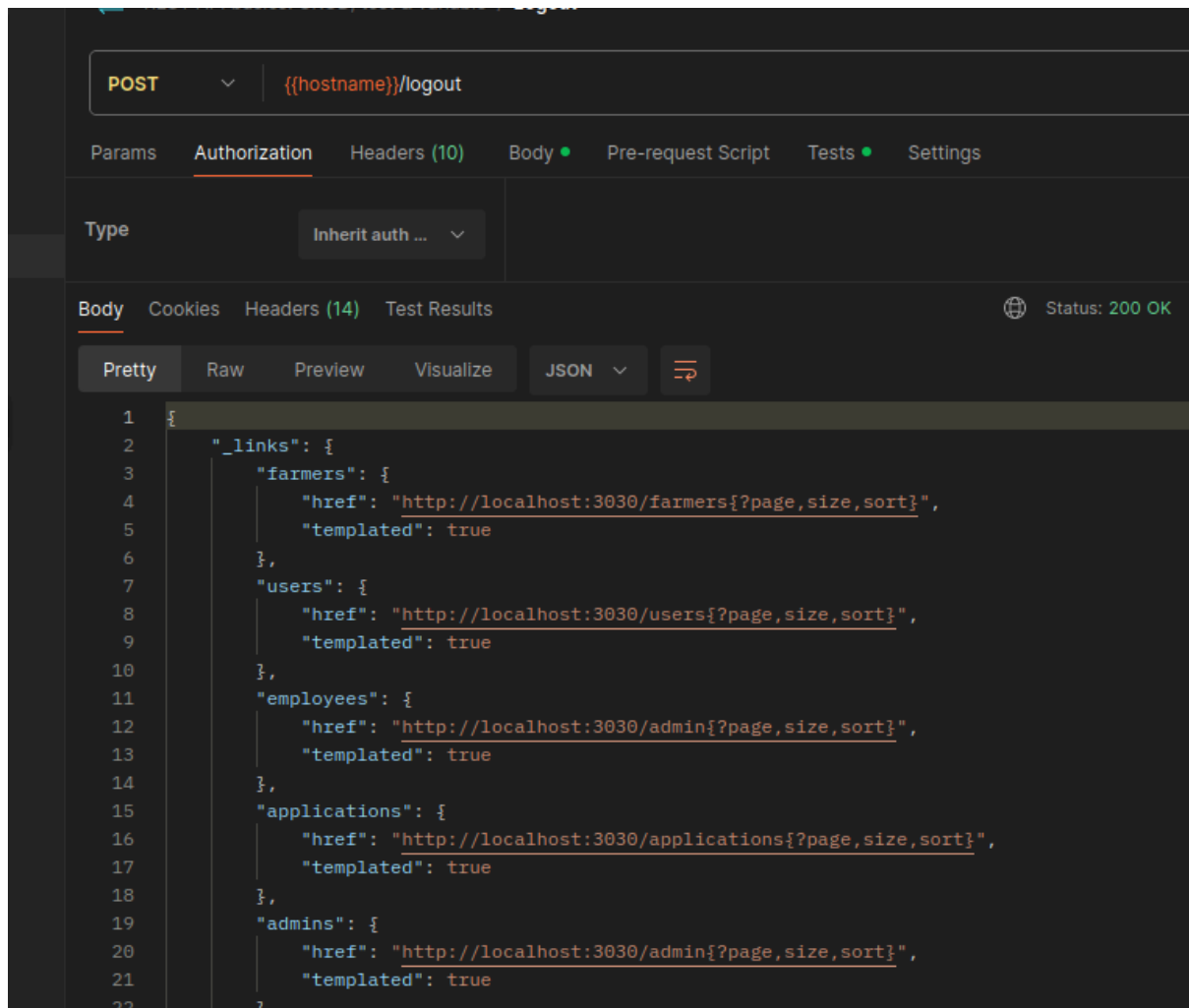
```
POST {{hostname}}/api/auth/signup

Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

8 {
9 }
10 // Mpampas Admin
11 {
12   "username": "PetrMan",
13   "name": "Petroulas Manolis",
14   "password": "PetrMan",
15   "birthDate": "1969-08-29",
16   "role": "ROLE_ADMIN"
17 }
18 }
19 // Mama Employee
20 {
21   "username": "LatsElen",
22   "name": "Latsi Eleni",
23   "password": "LatsElen",
24   "birthDate": "1972-11-03",
25   "role": "ROLE_EMPLOYEE"
26 }
27 // Giorgos Farmer
28 /**{
29   "username": "PetrGeorge",
30   "name": "Petroulas Giorgos",
31   "password": "PetrGeorge",
32   "birthDate": "2000-05-25",
33   "role": "ROLE_FARMER"
34 }
35 */
36 {
37   "username": "username",
38   "name": "UserName",
39   "password": "username",
40   "birthDate": "1995-05-25",
41   "role": "ROLE_USER"
42 }
```

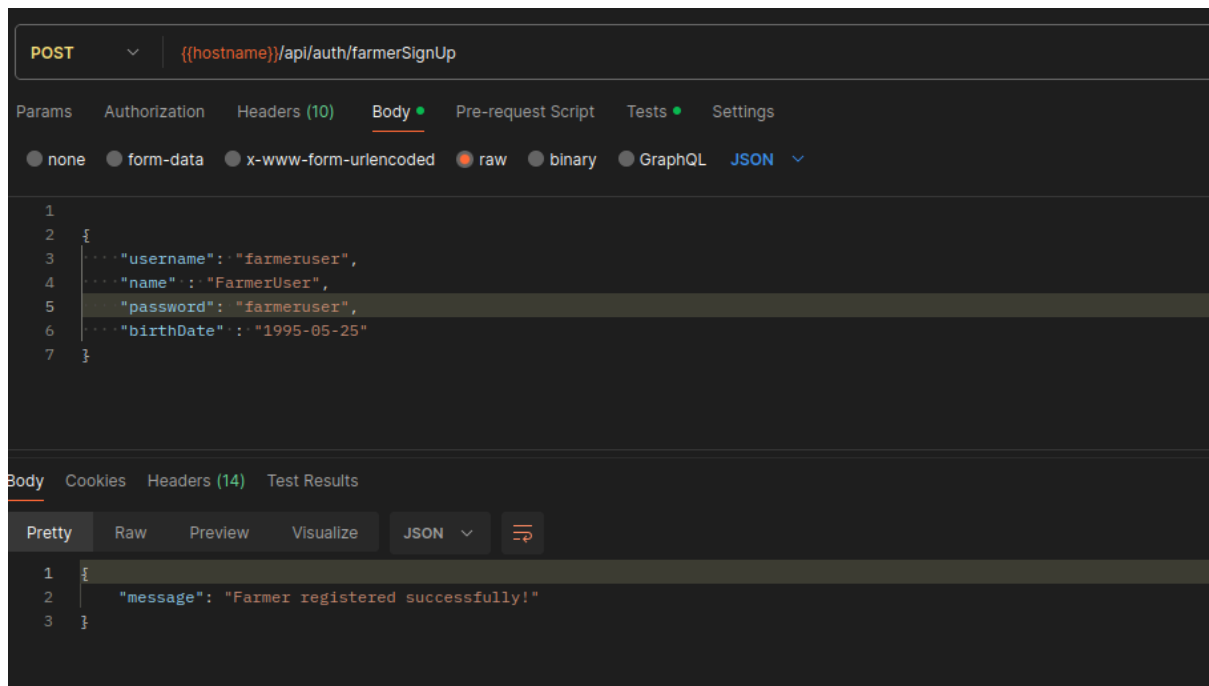
POST /api/auth/logout:

Είναι υπεύθυνο για την αποσύνδεση ενός χρήστη.
Δέχεται ένα αίτημα με το JWT που πρέπει να ακυρωθεί.
Το JWT προστίθεται σε μια μαύρη λίστα (blacklist) για ασφαλή αποσύνδεση.



POST /api/auth/farmerSignUp:

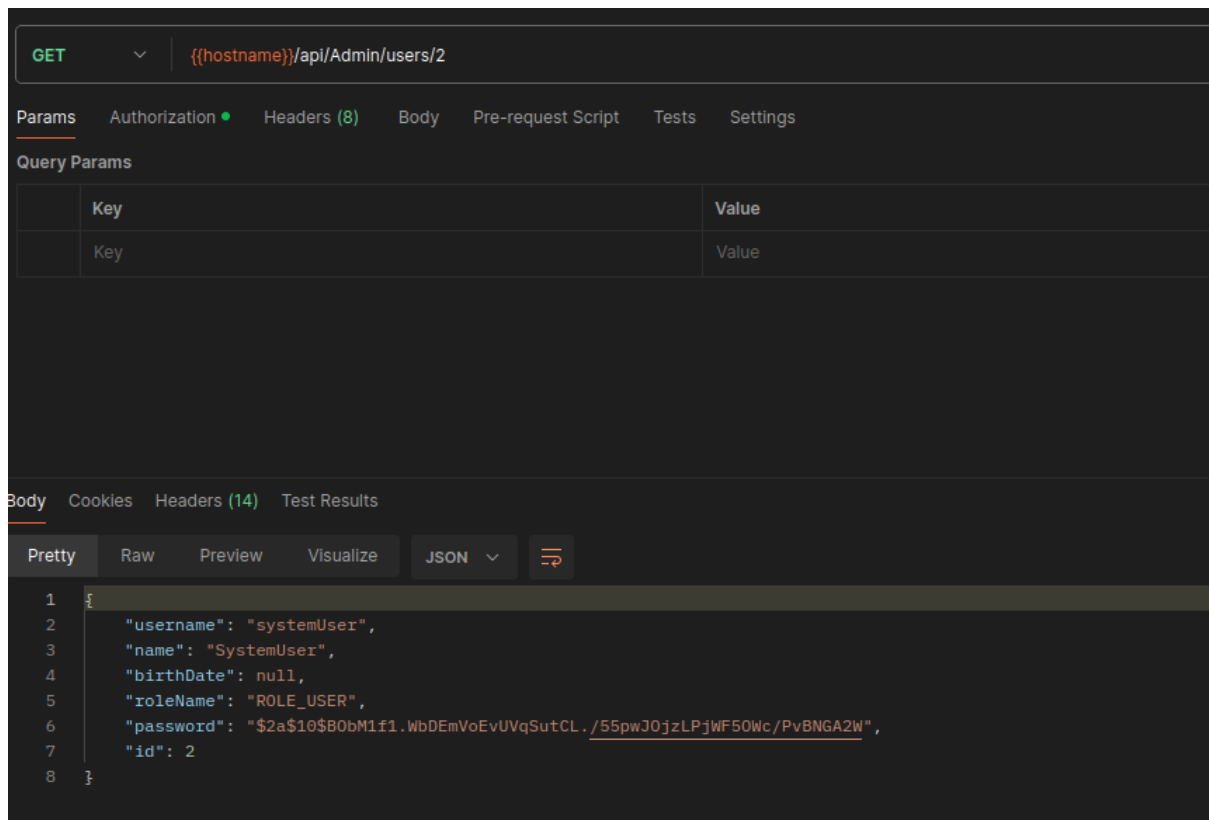
Είναι υπεύθυνο για την εγγραφή (registration) νέων χρηστών με ρόλο "Καλλιεργητής" (Farmer). Παρόμοιο με το `/api/auth/signup` αλλά εφαρμόζεται για καλλιεργητές.



ADMINRESTCONTROLLER

GET `api/Admin/users/{userId}`:

Επιστρέφει ένα αντικείμενο UserDTO για έναν συγκεκριμένο χρήστη, χρησιμοποιώντας το userDao. Η @GetMapping αναλαμβάνει το μονοπάτι `/users/{userId}`, όπου `{userId}` είναι η μεταβλητή στο μονοπάτι που αναπαριστά τον αριθμό αναγνώρισης του χρήστη. Το @ResponseBody δηλώνει ότι η τιμή που επιστρέφεται από τη μέθοδο πρέπει να συμπεριληφθεί απευθείας στο σώμα της απόκρισης HTTP.



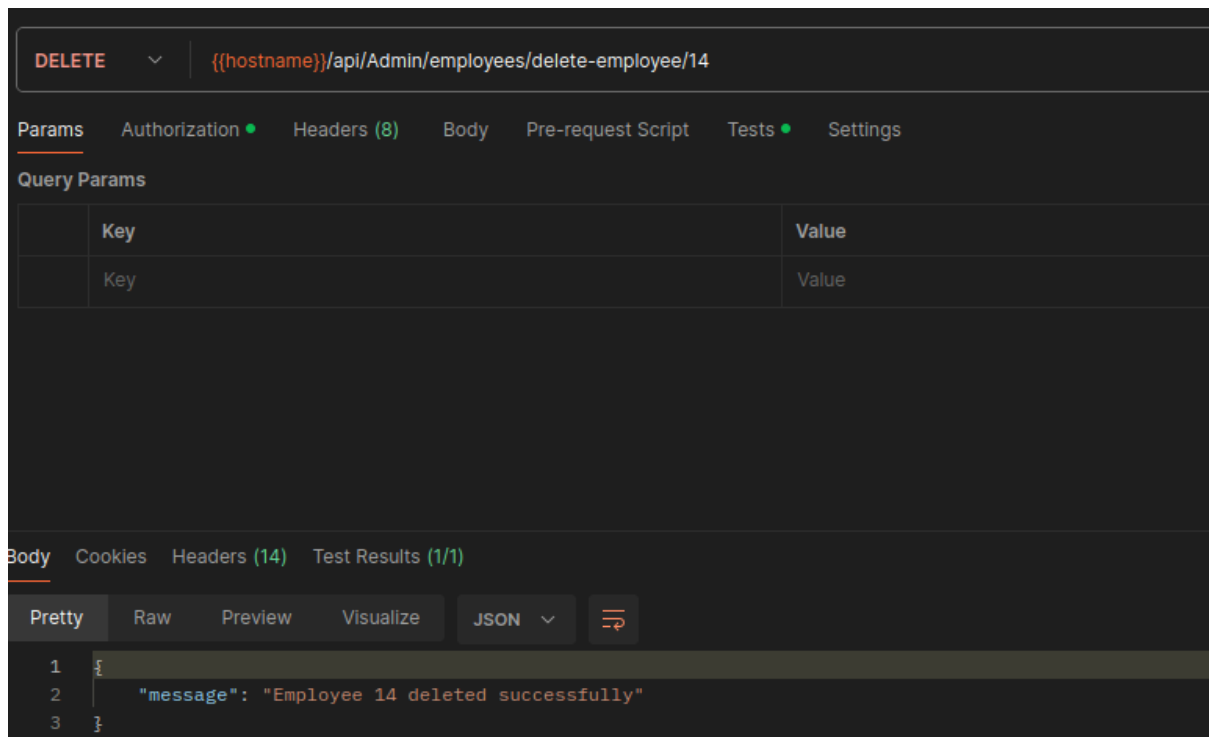
PUT api/Admin/users/update-user/{userId}:

Ενημερώνει τα στοιχεία ενός συγκεκριμένου χρήστη, χρησιμοποιώντας το userDao.

Η @PostMapping αναλαμβάνει το μονοπάτι `/users/update-user/{userId}`, όπου `{userId}` είναι ο αριθμός αναγνώρισης του χρήστη προς ενημέρωση.

Χρησιμοποιεί τα δεδομένα που παρέχονται στο σώμα του αιτήματος για να ενημερώσει τον χρήστη, όπως το όνομα χρήστη, το όνομα και την ημερομηνία γέννησης.

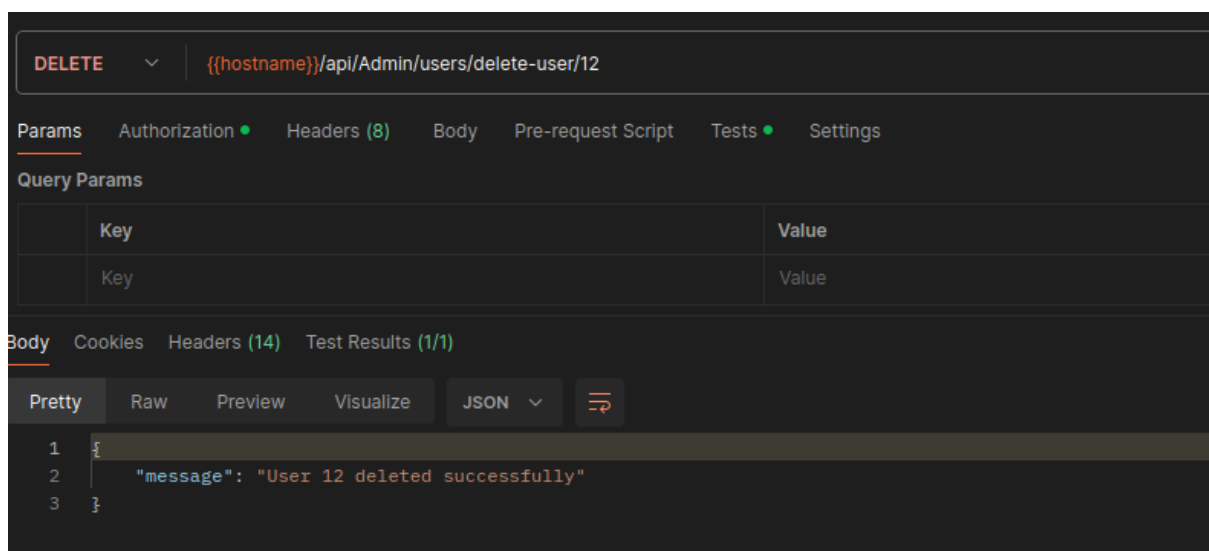
Επιπλέον, μπορεί να ενημερώσει τον ρόλο του χρήστη αν δοθεί ένας νέος ρόλος.



DELETE `api/Admin/users/delete-user/{userId}`:

Διαγράφει έναν συγκεκριμένο χρήστη, χρησιμοποιώντας το `userDao`.

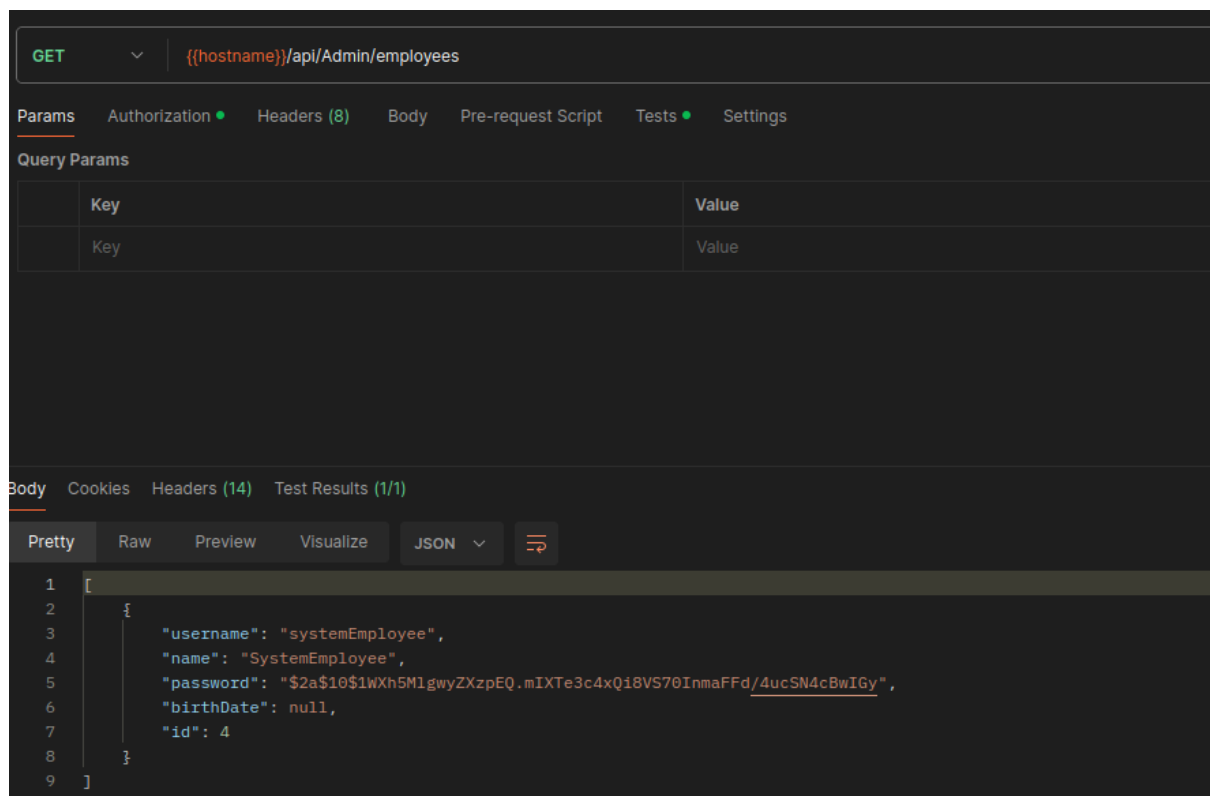
Η `@DeleteMapping` αναλαμβάνει το μονοπάτι `/users/delete-user/{userId}`, όπου `{userId}` είναι ο αριθμός αναγνώρισης του χρήστη προς διαγραφή.



Για τους Υπαλλήλους (Employees):

GET api/Admin/employees:

Επιστρέφει μια λίστα με όλους τους υπαλλήλους, χρησιμοποιώντας τον employeeDAO.



GET api/Adim/employees/{employeeId}:

Επιστρέφει έναν συγκεκριμένο υπάλληλο βάσει του `employeeId`, χρησιμοποιώντας τον `employeeDAO`.

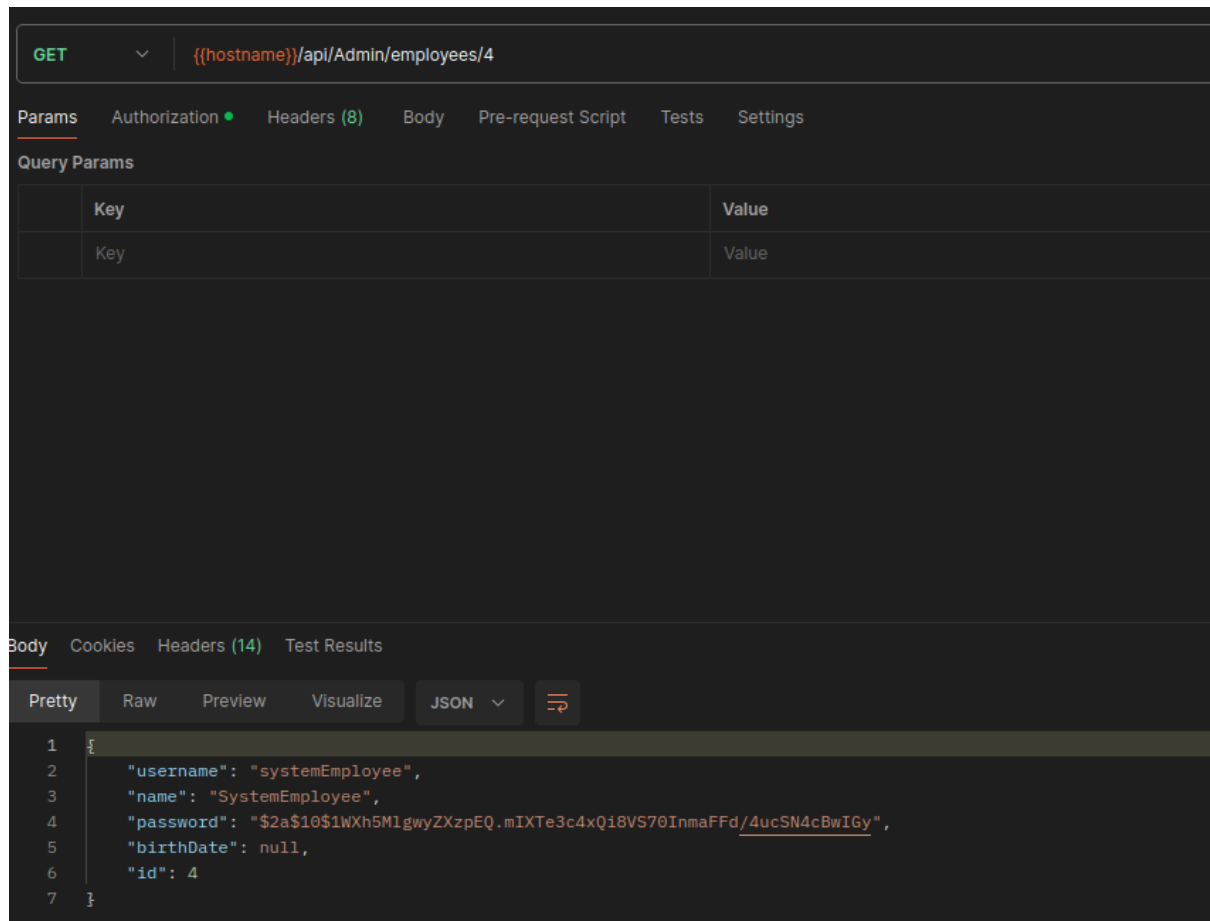
POST /employees/update-employee:

Ενημερώνει τα στοιχεία ενός υπαλλήλου, χρησιμοποιώντας τον employeeDAO.

Χρησιμοποιεί τα δεδομένα που παρέχονται στο σώμα του αιτήματος (με τη χρήση της UpdateRequest) για να ενημερώσει τον υπάλληλο, όπως το όνομα χρήστη, το όνομα και την ημερομηνία γέννησης.

Το αναγνωριστικό του υπαλλήλου (employeeId) προσδιορίζεται στο αντικείμενο UpdateRequest.

Επιστρέφει απάντηση HTTP με κατάλληλο μήνυμα.



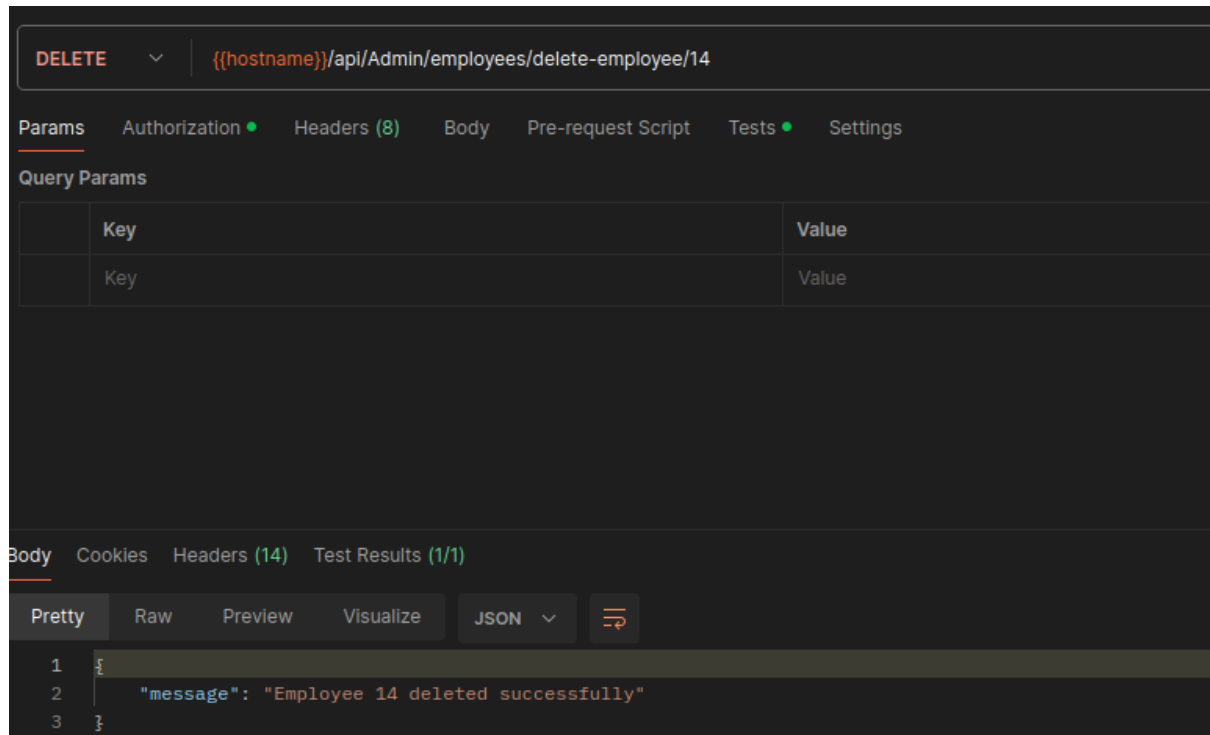
DELETE

api/Admin/employees/delete-employee/{employeeId}:

Διαγράφει έναν συγκεκριμένο υπάλληλο, χρησιμοποιώντας τον employeeDAO.

Το employeeId προσδιορίζεται στο μονοπάτι.

Επιστρέφει μήνυμα επιτυχίας ή αποτυχίας.



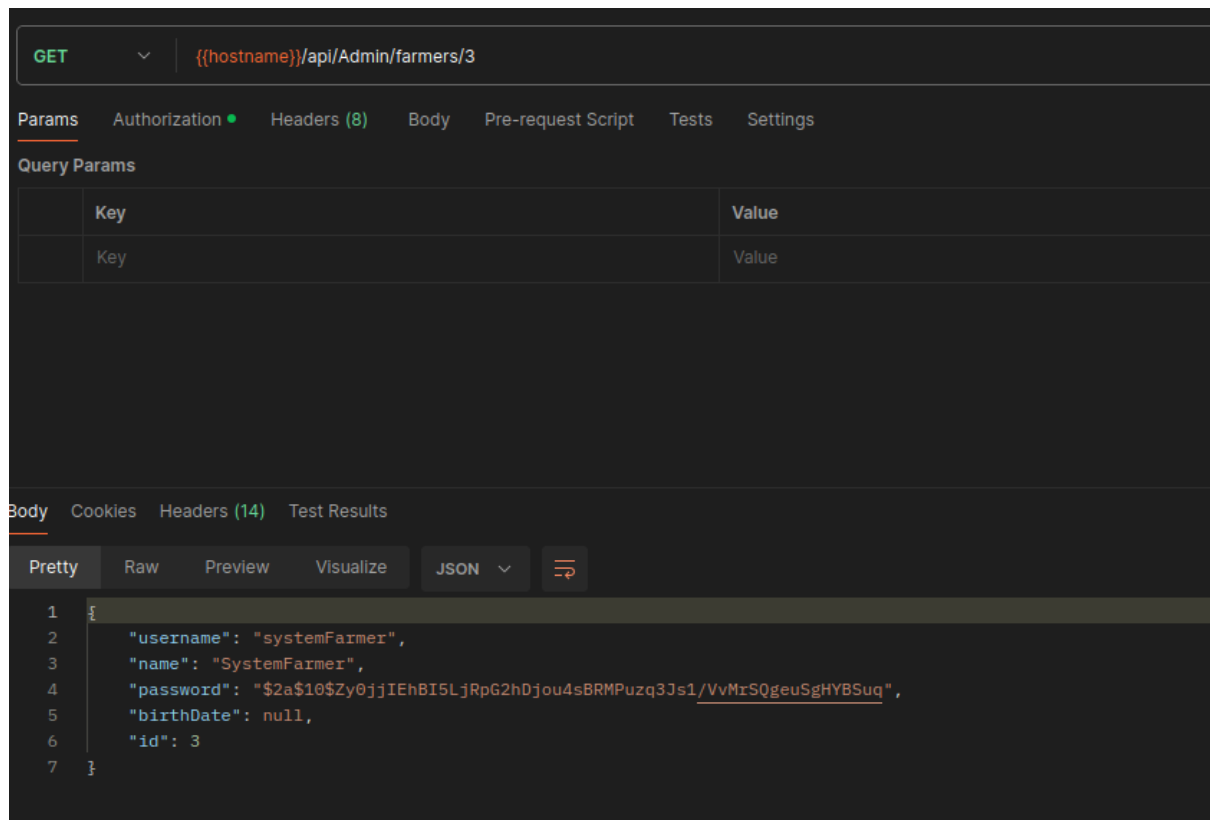
[Για τους Καλλιεργητές \(Farmers\):](#)

GET api/Admin/farmers:

Επιστρέφει μια λίστα με όλους τους καλλιεργητές, χρησιμοποιώντας τον farmerDao.

GET /farmers/{farmerId}:

Επιστρέφει έναν συγκεκριμένο καλλιεργητή βάσει του farmerId, χρησιμοποιώντας τον farmerDao.



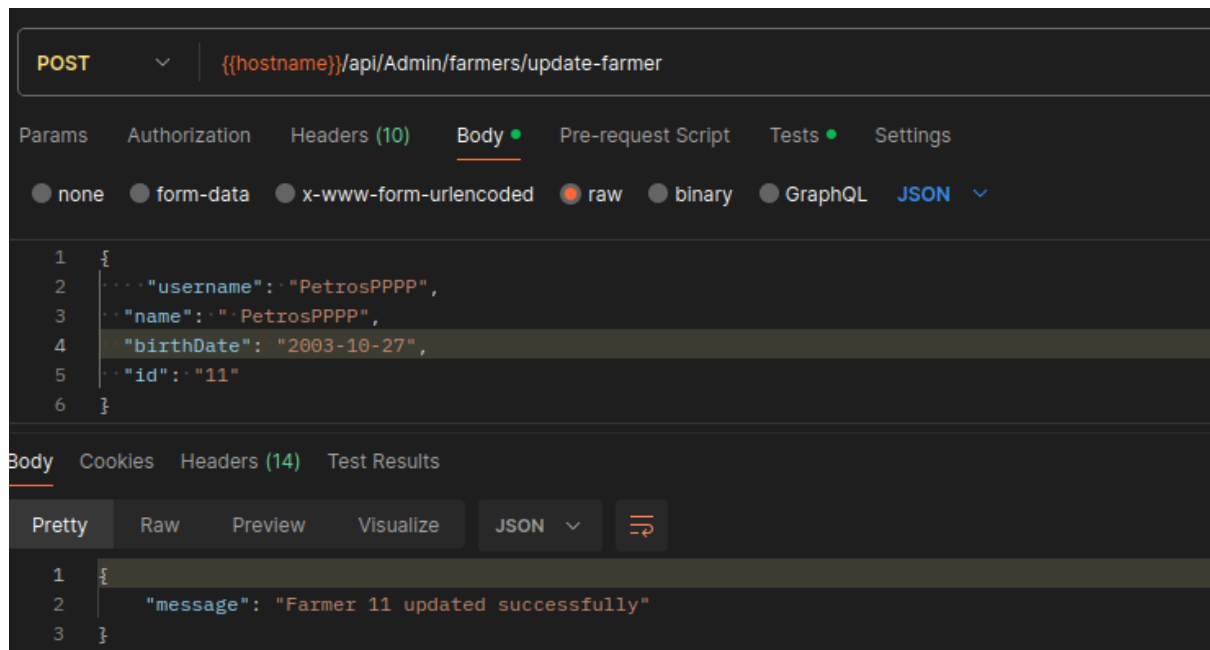
PUT /farmers/update-farmer:

Ενημερώνει τα στοιχεία ενός καλλιεργητή, χρησιμοποιώντας τον `farmerDao`.

Χρησιμοποιεί τα δεδομένα που παρέχονται στο σώμα του αιτήματος (με τη χρήση της `UpdateRequest`) για να ενημερώσει τον καλλιεργητή, όπως το όνομα χρήστη, το όνομα και την ημερομηνία γέννησης.

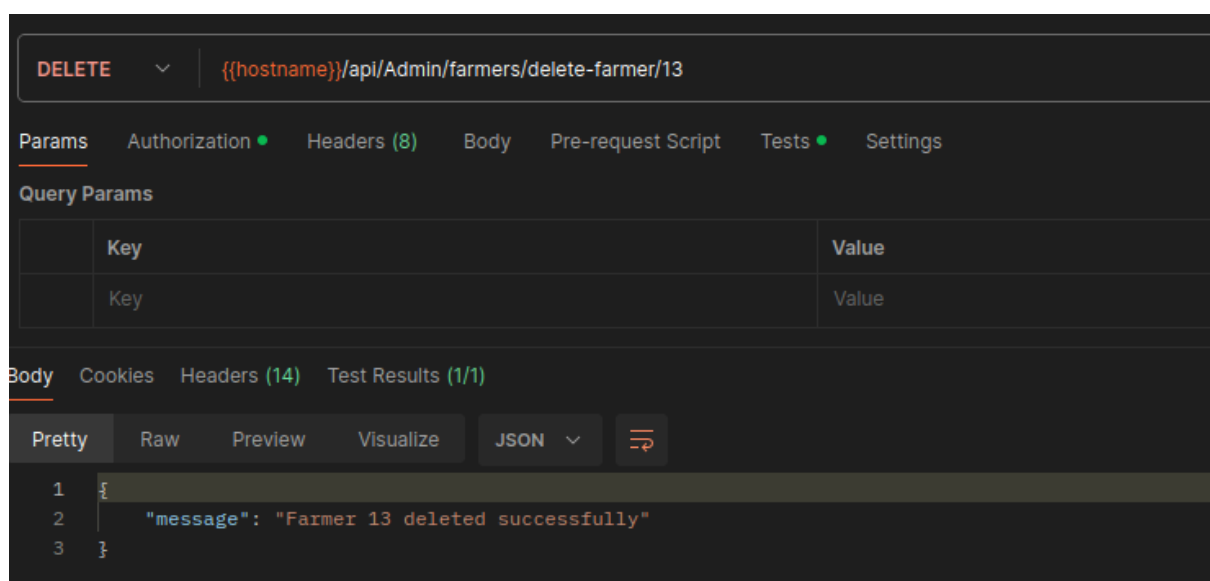
Το αναγνωριστικό του καλλιεργητή (`farmerId`) προσδιορίζεται στο αντικείμενο `UpdateRequest`.

Επιστρέφει απάντηση HTTP με κατάλληλο μήνυμα.



DELETE /farmers/delete-farmer/{farmerId}:

Διαγράφει έναν συγκεκριμένο καλλιεργητή, χρησιμοποιώντας τον `farmerDao`. Το `farmerId` προσδιορίζεται στο μονοπάτι. Επιστρέφει μήνυμα επιτυχίας ή αποτυχίας.



FARMERRESTCONTROLLER

POST /api/Farmer/submitApplication:

Αυτή η μέθοδος αναλαμβάνει την υποβολή αίτησης για έναν καλλιεργητή.

Λαμβάνει ένα αίτημα (SubmitRequest) που περιέχει τα στοιχεία της αίτησης (όνομα αγροτικής συνεταιριστικής, μέλη, προϊόντα, εμβαδόν καλλιέργειας, όνομα καλλιεργητή).

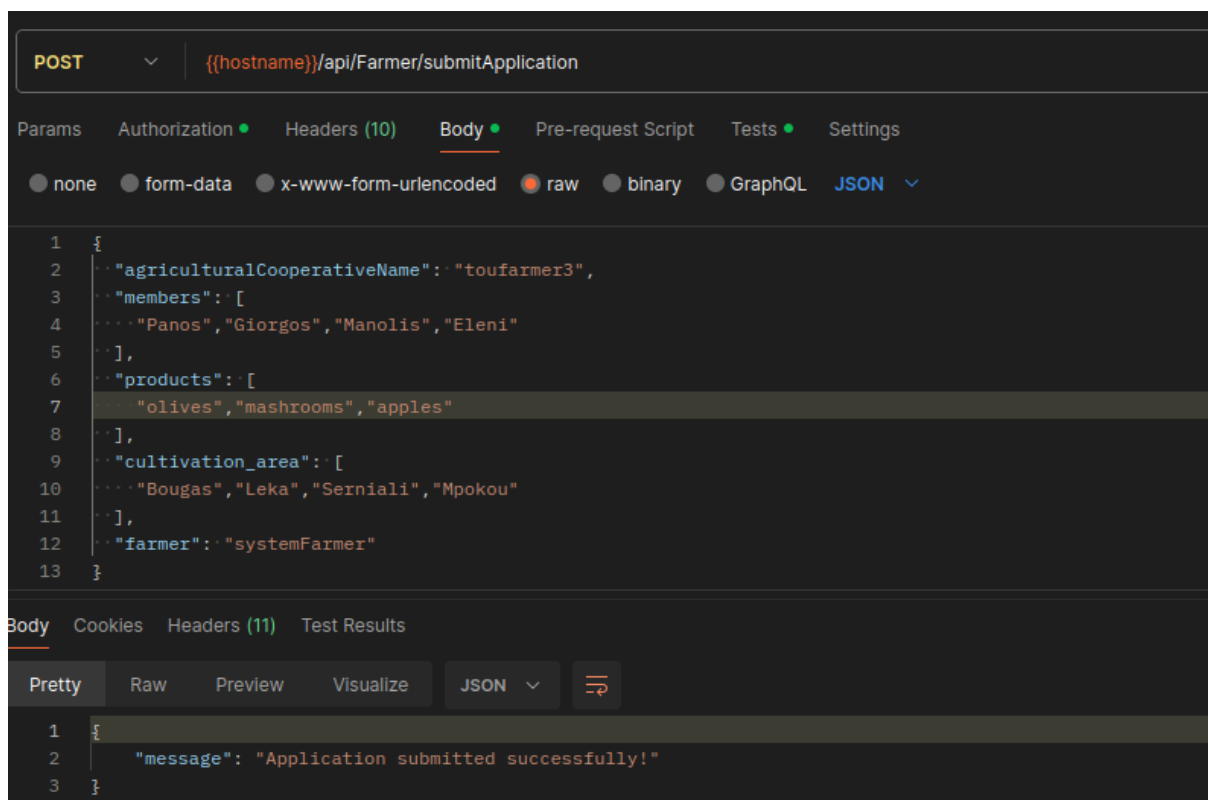
Ελέγχει αν υπάρχει ήδη αίτηση με το ίδιο όνομα αγροτικής συνεταιριστικής και, αν ναι, επιστρέφει μήνυμα λάθους.

Δημιουργεί ένα νέο αντικείμενο Application με τα στοιχεία της αίτησης και αποθηκεύει στη βάση δεδομένων.

Βρίσκει τον καλλιεργητή με βάση το όνομα που περιέχεται στο αίτημα και τον συσχετίζει με την αίτηση.

Τελικά, αποθηκεύει ξανά την αίτηση με τον συσχετισμένο καλλιεργητή.

Επιστρέφει μήνυμα επιτυχίας.



GET /api/Farmer/get/not:

Αυτή η μέθοδος επιστρέφει τις ειδοποιήσεις (αιτήσεις) που αφορούν τον συγκεκριμένο καλλιεργητή.

Λαμβάνει το όνομα του καλλιεργητή από το πεδίο αυθεντικοποίησης.

Χρησιμοποιεί το `farmerRepository` για να βρει τον καλλιεργητή.

Στη συνέχεια, χρησιμοποιεί το `applicationRepository` για να βρει όλες τις αιτήσεις που αφορούν αυτόν τον καλλιεργητή.
Επιστρέφει τη λίστα των αιτήσεων.

EMPLOYEE REST CONTROLLER

GET /api/Employee/applications:

Επιστρέφει όλες τις αιτήσεις που υπάρχουν στη βάση δεδομένων.
Χρησιμοποιεί το `applicationRepository` για να εκτελέσει μια ερώτηση που επιστρέφει όλες τις αιτήσεις.

```
GET {{hostname}}/api/Employee/applications

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

Body  Cookies  Headers (11)  Test Results

Pretty  Raw  Preview  Visualize  JSON  ↕

1  [
2    {
3      "id": 2,
4      "agriculturalCooperativeName": "toufarmer",
5      "comments": null,
6      "status": "UNSEEN",
7      "members": [
8        "Panos",
9        "Giorgos",
10       "Manolis",
11       "Eleni"
12     ],
13     "products": [
14       "string"
15     ],
16     "farmer": {
17       "username": "systemFarmer",
18       "name": "SystemFarmer",
19       "password": "$2a$10$Zy0jjIEhBI5LjRpG2hDjou4sBRMPuzq3JJs1/VvMrSQgeuSgHYBSuq",
20       "birthDate": null,
21       "id": 3
22     },
23     "employee": null,
24     "cultivationArea": [
25       "Bougas",
26       "Leka",
27       "Serniali",
28       "Mpokou"
29     ]
30   },
31   {
32     "id": 1,
33     "agriculturalCooperativeName": "touGiorgou",
34     "comments": null,
35     "status": "UNSEEN",
```

GET /api/Employee/application/{applicationId}:

Επιστρέφει μια συγκεκριμένη αίτηση βάσει του applicationId.

Χρησιμοποιεί το applicationRepository για να επιστρέψει την αίτηση που αντιστοιχεί στο δοθέν applicationId.

The screenshot shows a REST client interface with a GET request to `{{hostname}}/api/Employee/application/2`. The response is displayed in JSON format, showing details for an application with ID 2, including agricultural cooperative name, status, members, products, farmer information, and cultivation area.

```
1 {
2   "id": 2,
3   "agriculturalCooperativeName": "toufarmer",
4   "comments": null,
5   "status": "UNSEEN",
6   "members": [
7     "Panos",
8     "Giorgos",
9     "Manolis",
10    "Eleni"
11  ],
12  "products": [
13    "string"
14  ],
15  "farmer": {
16    "username": "systemFarmer",
17    "name": "SystemFarmer",
18    "password": "$2a$10$Zy0jjIEh8I5LjRpG2hDjou4sBRMPuzq3Js1/VvMrSQgeuSgHYBSuq",
19    "birthDate": null,
20    "id": 3
21  },
22  "employee": null,
23  "cultivationArea": [
24    "Bougas",
25    "Leka",
26    "Serniali",
27    "Mpokou"
28  ]
29 }
```

PUT /api/Employee/application/approval/{applicationId}:

Εξετάζει μια αίτηση και ενημερώνει τη βάση δεδομένων με την απόφαση του υπαλλήλου. Οι παράμετροι είναι το `applicationId` (το οποίο παίρνει από το path) και το `checkRequest` (το οποίο παίρνει από το body του αιτήματος). Χρησιμοποιεί το `applicationRepository` για να ανακτήσει την αίτηση βάσει του `applicationId`. Ελέγχει αν η αίτηση υπάρχει. Αν όχι, επιστρέφει ένα μήνυμα λάθους. Χρησιμοποιεί το `employeeRepository` για να βρει τον υπάλληλο βάσει του ονόματος του υπαλλήλου στο `checkRequest`. Ελέγχει αν ο υπάλληλος υπάρχει. Αν όχι, επιστρέφει ένα μήνυμα λάθους. Ενημερώνει την αίτηση με τα στοιχεία από το `checkRequest`. Αποθηκεύει τις αλλαγές στο `applicationRepository`. Επιστρέφει ένα μήνυμα επιτυχίας.

