**REST API**

**Εισαγωγή:**

Τι είναι ένα API?

Γενικά ένα API (Application Programming Interface) παρέχει ένα interface μεταξύ 2 συστημάτων/υπολογιστών για να μπορούν να αλληλοεπιδρούν μεταξύ τους.

Πότε ένα API θεωρείτε web Service?

Τα APIs που χρησιμοποιούν το πρωτόκολλο μεταφοράς HTTP ονομάζονται Web Services. Μερικά web services είναι τα REST και SOAP APIs.

Τι είναι ένα REST API?

Ένα REST (Representational State Transfer) API είναι ένα αρχιτεκτονικό πρότυπο (architectural pattern) το οποίο περιγράφει πώς τα κατανεμημένα συστήματα (distributed systems) μπορούν να σχεδιαστούν για να παρέχουν ένα συνεπές interface.

Ένα REST API επικεντρώνεται στα resources και στους τρόπους που έχει κάποιος πρόσβαση σε αυτά. H πρόσβαση των resources γίνετε μέσω προκαθορισμένων URLs συνοδεύοντας τα μαζί με ένα method.

Τα πιο συνηθισμένα methods είναι:

GET (για να πάρουμε ένα resource ή ένα collection από resources),

POST (για δημιουργήσουμε ένα resource),

PUT (για να κάνουμε update ένα resource),

DELETE (για να διαγράψουμε ένα resource)

**Σημείωση**: Αυτά τα methods υποστηρίζονται και από αυτό το REST API

**Status Codes του REST API:**

* **200 (Ok)**: Το request έγινε επιτυχώς.
* **201 (Created)**: Το request έγινε επιτυχώς και το resource δημιουργήθηκε.
* **400 (Bad Request)**: Το request δεν έγινε κατανοητό ή δεν συμπληρώθηκαν σωστά τα πεδία που έπρεπε.
* **401 (Unauthorized)**: Το authentication απέτυχε (συνήθως στο login)
* **403 (Forbidden)**: ο χρήστης δεν είχε τα απαραίτητα permissions για το operation που ζητάει.
* **404 (Not Found)**: Το resource δεν υπάρχει.
* **501 (Not implemented)**: Ο server δεν υποστηρίζει την λειτουργία που ζητείτε προκειμένου να ικανοποιηθεί το request.

**Methods που υποστηρίζονται από το REST API:**

Όπως ειπώθηκε και πιο πάνω τα Methods που υποστηρίζονται από το REST API είναι:

* GET (για να πάρουμε ένα resource ή ένα collection από resources),
* POST (για δημιουργήσουμε ένα resource),
* PUT (για να κάνουμε update ένα resource),
* DELETE (για να διαγράψουμε ένα resource)

**Περιγραφή των responses (server):**

Ο server αφού διαχειριστεί ένα request από τον client, στέλνει το κατάλληλο response πάντα σε μορφή json.

Η γενική μορφή του json είναι:

{

Status : “status”,

Msg : “message”,

Data : {obj\_data} ή [{obj0\_data} , {obj1\_data} , … , {objN\_data}],

ResourceID : “resource\_id”

}

Σημείωση: το ResourceID επιστρέφετε μόνο για κάποια νέα προσθήκη (POST) στην βάση δεδομένων.

Όπου:

Το status μπορεί να είναι «SUCCESS», «ERROR» ή «WARINING»

Το message περιέχει πληροφορίες σχετικές με τα data που επιστρέφονται στον client.

Τα data είναι ένα object ή ένα array από objects ανάλογα με τα resources που «ζητάει» ο client (data υπάρχουν και σε περιπτώσεις create και update για πιθανή επιβεβαίωση των στοιχείων).

Το ResourceID είναι το μοναδικό αναγνωριστικό. Δημιουργείτε στον server όταν γίνει κάποιο POST request για προσθήκη ενός νέου user, listing, bid κλπ.

**Resource: Users** ( "api/users/:UID")

Όταν ένας user(customer ή professional) κάνει εγγραφή, δημιουργείτε ένα καινούριο account με τα στοιχεία του, τα οποία μπορεί να τροποποιήσει ή να διαγράψει.

Customer Attributes:

|  |  |  |
| --- | --- | --- |
| name**\*** | String | First Name |
| surname\* | String | Last Name |
| username\* | String | Username |
| accountType\* | Enum | Account Type |
| gender\* | Enum | User Gender |
| bday\* | Date | Birthday |
| address\* | String | Address |
| email\* | String | Email |
| phoneNum | String | Phone Number |
| password\* | String | Password |

Δημιουργία Ενός Customer

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **201 (Created)** |
| URL: **/api/users** | Body: {  "status": "SUCCESS",  "msg": "**User Added**",  "data": {  "name": "customer name",  "surname": "customer surname",  "username": " customer username",  "accountType": "CUSTOMER",  "gender": "MALE",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953"  },  **"resourceID": " a985-82568e5b403d "**  } |
| Body: {  "name": "customer name",  "surname": "customer surname",  "username": " customer username",  "accountType": "CUSTOMER",  "gender": "MALE",  "bday": "1996-04-26",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953",  "password": " customerP@ssword!"  } |

Δημιουργεί ένα νέο customer με ένα μοναδικό UID(resourceID).

Δημιουργίας Ενός Customer (**Error 400**: Bad Request)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: POST | Status code: **400 (****Bad Request)** |
| URL: /api/users | Body: {  "status": "ERROR",  "msg": "**Invalid field birthday**",  "data": {}  } |
| Body: {  "name": "customer name",  "surname": "customer surname",  "username": " customer username",  "accountType": "CUSTOMER",  "gender": "MALE",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953",  "password": " customerP@ssword!"  } |

**Προσοχή:** Στην εγκυρότητα των στοιχείων και στο ποια πεδία είναι required.

Προσθήκη ενός field σε ένα customer (**Error 501**: Not Implemented)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **501 (****Not Implemented)** |
| URL: /api/users/**a985-82568e5b403d** | Body: {  "status": "ERROR",  "msg": "**Server does not support the**  **functionality required to fulfill the**  **request**",  "data": {}  } |
| Body: {  "phoneNum": "99994953"  } |

**Προσοχή:** Στη σωστή χρήση των methods και URLs.

Get Customer

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL: **/api/users/ a985-82568e5b403d** | Body: {  "status": "SUCCESS",  "msg": "**User** **a985-82568e5b403d**",  "data": {  "name": " cust name",  "surname": " cust surname",  "username": " cust username",  "accountType": "CUSTOMER",  "gender": "MALE",  "UID": "a985-82568e5b403d",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953"  }  } |
| Body: {} |

Επιστέφει τον customer με UID(resourceID)="a985-82568e5b403d".

Get Customer (**Error 404**: Not Found)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: **404 (****Not Found)** |
| URL /api/users/**a985-82568e5b4-3453-345** | Body: {  "status": "ERROR",  "msg": "**User Does Not Exist**",  "data": { }  } |
| Body: {} |

**Προσοχή:** Πάντα ζητάμε users που υπάρχουν στο σύστημα.

Get Customer surname (**501**: Not Implemented)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: **501 (Not Implemented)** |
| URL: /api/users/ a985-82568e5b403d/**surname** | Body: {  "status": "ERROR",  "msg": "**Server does not support the**  **functionality required to fulfill the**  **request**",  "data": {}  } |
| Body: {} |

**Προσοχή:** Πάντα ζητάμε ολόκληρο το resource και όχι ένα υποσύνολο του.

Update Ενός Customer

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: PUT | Status code: **200 (ok)** |
| URL: /api/users/ **a985-82568e5b403d** | Body: {  "status": "SUCCESS",  "msg": "User Edited",  "data": {  "name": " cust name",  "surname": " cust surname",  "username": " cust username",  "accountType": "CUSTOMER",  "gender": "MALE",  "UID": "a985-82568e5b403d",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953"  }  } |
| Body: {  "name": "cust name",  "surname": " cust surname",  "username": " cust username",  "accountType": "CUSTOMER",  "gender": "MALE",  "bday": "1996-04-26",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953",  "password": " custP@ssword!"  } |

Κάνει update τα νέα στοιχεία του customer με UID(resourceID)="a985-82568e5b403d".

Update Ενός Customer (**Error 403**: Forbidden)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: PUT | Status code: **403 (****Forbidden)** |
| URL: /api/users/**a985-82568e5b4-234** | Body: {  "status": "ERROR",  "msg": "**No permission**",  "data": { }  } |
| Body: {  "name": "cust name",  "surname": " cust surname",  "username": " cust username",  "accountType": "CUSTOMER",  "gender": "MALE",  "bday": "1996-04-26",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953",  "password": " custP@ssword!"  } |

**Προσοχή**: Ποτέ δεν πρέπει κάποιος χρήστης να προσπαθεί να αλλάξει τα δεδομένα κάποιου άλλου χρήστη χωρίς να έχει εξουσιοδότηση.

Διαγραφή Ενός Customer

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **DELETE** | Status code: **200 (ok)** |
| URL: /api/users/**a985-82568e5b403d** | Body: {  "status": "SUCCESS",  "msg": "User Deleted",  "data": { }  } |
| Body: { } |

Διαγράφει τον customer με UID(resourceID)="a985-82568e5b403d".

Διαγραφή Ενός Customer (**Error 403**: Forbidden)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: DELETE | Status code: **403 (Forbidden)** |
| URL: /api/users/**a985-82568e5-234** | Body: {  "status": "ERROR",  "msg": "**No permission**",  "data": { }  } |
| Body: { } |

**Προσοχή**: Ποτέ δεν πρέπει κάποιος χρήστης να προσπαθεί να διαγράψει κάποιον άλλο χρήστη χωρίς να έχει εξουσιοδότηση.

Professional Attributes:

|  |  |  |
| --- | --- | --- |
| name\* | String | First Name |
| surname\* | String | Last Name |
| username\* | String | Username |
| accountType\* | Enum | Account Type |
| gender\* | Enum | User Gender |
| bday\* | Date | Birthday |
| address\* | String | Address |
| email\* | String | Email |
| phoneNum\* | String | Phone Number |
| password\* | String | Password |
| jobs\* | Enum | Jobs |
| workExperience | Double | Work Experience In Years |
| sLocations\* | Enum | Locations Served |
| aboutMe | String | About Me Description |

Δημιουργία Ενός Professional

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **201 (Created)** |
| URL: **/api/users** | Body: {  "status": "SUCCESS",  "msg": "**User Added**",  "data": {  "jobs": "",  "workExperience": 4,  "aboutMe": "aboutMe",  "name": "Prof name",  "surname": " Prof surname",  "username": " Prof username",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": " Prof address",  "email": " Prof@gmail.com",  "phoneNum": "99994953"  }  **"resourceID": "343e0e5a-e6e0"**,  } |
| Body: {  "name": "Prof name",  "surname": " Prof surname",  "username": " Prof username",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "bday": "1996-04-26",  "address": " Prof address",  "email": " Prof@gmail.com",  "phoneNum": "99994953",  "password": " ProfP@ssword!",  "job": "student",  "workExperience": "4",  "sLocations": "NICOSIA",  "aboutMe": "aboutMe"  } |

Δημιουργεί ένα νέο professional με ένα μοναδικό UID(resourceID).

Δημιουργία Ενός Professional (**Error 400**: Bad Request)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: POST | Status code: **400 (Bad Request)** |
| URL: /api/users | Body: {  "status": "ERROR",  "msg": "**Invalid field username**",  "data": {}  } |
| Body: {  "name": "Prof name",  "surname": " Prof surname",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "bday": "1996-04-26",  "address": " Prof address",  "email": " Prof@gmail.com",  "phoneNum": "99994953",  "password": " ProfP@ssword!",  "job": "student",  "sLocations": "NICOSIA"  } |

**Προσοχή**: Στην εγκυρότητα των στοιχείων και στο ποια πεδία είναι required. Get Professional

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: 200 (ok) |
| URL: /api/users/**343e0e5a-e6e0** | Body: {  "status": "SUCCESS",  "msg": "User 343e0e5a-e6e0",  "data": {  "jobs": "",  "workExperience": 4,  "aboutMe": "aboutMe",  "name": "Prof name",  "surname": " Prof surname",  "username": " Prof username",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "UID": "343e0e5a-e6e0",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": " Prof address",  "email": " Prof@gmail.com",  "phoneNum": "99994953"  }  } |
| Body: { } |

Επιστέφει τον professional με UID(resourceID)="343e0e5a-e6e0".

Get Professional (**Error 404**: Not Found)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: **404 (Not Found)** |
| URL: /api/users/**343e0e5a-e6e0** | Body: {  "status": "ERROR",  "msg": "**User Does Not Exist**",  "data": { }  } |
| Body: { } |

**Προσοχή**: Πάντα ζητάμε users που υπάρχουν στο σύστημα.

Update Ενός Professional

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: PUT | Status code: **200 (ok)** |
| URL: /api/users/**343e0e5a-e6e0** | Body: {  "status": "SUCCESS",  "msg": "**User Edited**",  "data": {  "jobs": "",  "workExperience": 4,  "aboutMe": "aboutMe",  "name": "Pname",  "surname": " Psurname",  "username": " Pusername",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "UID": "343e0e5a-e6e0",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": " Paddress",  "email": " Prof@gmail.com",  "phoneNum": "99994953"  }  } |
| Body: {  "name": "Pname",  "surname": " Psurname",  "username": " Pusername",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "bday": "1996-04-26",  "address": " Paddress",  "email": " Prof@gmail.com",  "phoneNum": "99994953",  "password": " ProfP@ssword!",  "job": "student",  "workExperience": "4",  "sLocations": "NICOSIA",  "aboutMe": "aboutMe"  } |

Κάνει update τα νέα στοιχεία του professional με UID(resourceID)="343e0e5a-e6e0".

Update Ενός Professional (**Error 403**: Forbidden)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: PUT | Status code: **403 (****Forbidden)** |
| URL: /api/users/**343e0e5a-e6e0** | Body: {  "status": "ERROR",  "msg": " **No permission**",  "data": { }  } |
| Body: {  "name": "Pname",  "surname": " Psurname",  "username": " Pusername",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "bday": "1996-04-26",  "address": " Paddress",  "email": " Prof@gmail.com",  "phoneNum": "99994953",  "password": " ProfP@ssword!",  "job": "student",  "sLocations": "NICOSIA"  } |

**Προσοχή**: Ποτέ δεν πρέπει κάποιος χρήστης να προσπαθεί να αλλάξει τα δεδομένα κάποιου άλλου χρήστη χωρίς να έχει εξουσιοδότηση.

Διαγραφή Ενός Professional

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: DELETE | Status code: **200 (ok)** |
| URL: /api/users/**343e0e5a-e6e0** | Body: {  "status": "SUCCESS",  "msg": "User Deleted",  "data": { }  } |
| Body: { } |

Διαγράφει τον professional με UID(resourceID )="343e0e5a-e6e0".

Διαγραφή Ενός Professional (**Error 403**: Forbidden)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: DELETE | Status code: **403 (Forbidden)** |
| URL: /api/users/**343e0e5a-e6e0** | Body: {  "status": "ERROR",  "msg": "**No permission**",  "data": { }  } |
| Body: { } |

**Προσοχή**: Ποτέ δεν πρέπει κάποιος χρήστης να προσπαθεί να διαγράψει κάποιον άλλο χρήστη χωρίς να έχει εξουσιοδότηση.

Get All Users

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL: **/api/users** | Body: {  "status": "SUCCESS",  "msg": "Get All Users",  "data": [{  "jobs": "",  "workExperience": 4,  "aboutMe": "aboutMe",  "name": "Prof name",  "surname": " Prof surname",  "username": " Prof username",  "accountType": "PROFESSIONAL",  "gender": "MALE",  "UID": "343e0e5a-e6e0",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": " Prof address",  "email": " Prof@gmail.com",  "phoneNum": "99994953"  },  {  "name": " cust name",  "surname": " cust surname",  "username": " cust username",  "accountType": "CUSTOMER",  "gender": "MALE",  "UID": "a985-82568e5b403d",  "bday": "Apr 26, 1996 12:00:00 AM",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953",  }]  } |
| Body: { } |

Επιστέφει ένα πίνακα με όλους τους χρήστες που υπάρχουν στο σύστημα.

login

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: POST | Status code: 200 (ok) |
| URL: /api/login | Header: { "token": "token" }  Body: {  "status": "SUCCESS",  "msg": " login Successfully",  "data": { }  } |
| Body: {  email: "[emailtest@gmail.com](mailto:emailtest@gmail.com)"  password: " custP@ssword!"  } |

Επιστρέφει το token στο header

login (**Error 401**: Unauthorized)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: POST | Status code: **401 (Unauthorized)** |
| URL: /api/login | Body: {  "status": "ERROR",  "msg": "**Unauthorized User**",  "data": { }  } |
| Body: {  email: "[emailtest@gmail.com](mailto:emailtest@gmail.com)"  password: " custP@ssword!"  } |

Επιστρέφει αυτό το response όταν το email ή το password είναι λάθος.

logout

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: 200 (ok) |
| URL: /api/logout | Body: {  "status": "SUCCESS",  "msg": " logout Successfully",  "data": { }  } |
| Body: { } |

Αποσύνδεση του χρήστη από το σύστημα.

Πίνακας με URLs και διαθέσιμα query params:

|  |  |  |
| --- | --- | --- |
| **URL:** | **Query Params:** | **Comments:** |
| /api/users | ?gender=male | Returns all male users |
| ?created-after=10-2-2019 | Returns users created after 10-2-2019 |
| ?created-before=10-3-2019 | Returns users created before 10-3-2019 |
| ?account-type=customer | Returns all customers |
| ?count=true | Returns the count of all users |
| ?check=email | checks email availability (POST req) |
| ?check=username | checks username availability (POST req) |
| ?rating=4 | Returns users with specified rating |
| ?served-location=Nicosia, Heraklion, … | Returns users who served at least one of specified locations |
| ?jobs=job1,job2, … | Returns users with at least one of specified jobs |
| /api/users/:UID | ?rating=true | get user rating |

**Σημείωση:** Μπορούν να συνδυαστούν περισσότερα από ένα query params για κάθε request με την χρήση του συμβόλου(&).

Check Email Availability

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: POST | Status code: 200 (ok) |
| URL: /api/users?check=email | Body: {  "status": "SUCCESS",  "msg": " Email Is Available",  "data": {  "isAvailable": true  }  } |
| Body: {  email: "emailtest@gmail.com"  } |

Επιστέφει αν είναι available το συγκεκριμένο email (χρησιμοποιείτε κυρίως κατά το signup).

Check Username Availability

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: POST | Status code: 200 (ok) |
| URL: /api/users?check=username | Body: {  "status": "SUCCESS",  "msg": " Username Is Available",  "data": {  "isAvailable": true  }  } |
| Body: {  username: "testUsername"  } |

Επιστέφει αν είναι available το συγκεκριμένο username (χρησιμοποιείτε κυρίως κατά το signup).

Get users (όπου το account type τους είναι customer και το gender είναι male)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: 200 (ok) |
| URL: /api/users?account-type=customer&  gender=male | Body: {  "status": "SUCCESS",  "msg": "Get Users",  "data": [{  "name": " cust name",  "surname": " cust surname",  "username": " cust username",  "accountType": "CUSTOMER",  "gender": "MALE",  "UID": "a985-82568e5b403d",  "bday": "Apr 02, 1999 12:00:00 AM",  "address": "customer address",  "email": "customer email",  "phoneNum": "99994953"  }]  } |
| Body: { } |

Επιστέφει ένα array με όλους τους users που ικανοποιούν τις 2 συνθήκες(δηλαδή να είναι male customers).

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: GET | Status code: 200 (ok) |
| URL: /api/users?account-type=professionals&  count=true | Body: {  "status": "SUCCESS",  "msg": "Count Users",  "data": {  "sum": "10"  }  } |
| Body: { } |

Επιστέφει πόσοι είναι οι professionals στο σύστημα. Χρησιμοποιείτε όταν θέλουμε να ξέρουμε το σύνολο ενός collection χωρίς να χρειαστεί να πάρουμε όλη την αναπαράσταση του resource(πχ για στατιστικά της σελίδας).

**Resource: Listings** ("api/users/:UID/listings/:LID" ή "api/listings/:LID")

Όταν ένας user(customer ή professional) κάνει post ένα listing, δημιουργείτε ένα καινούριο listing με τα πεδία που συμπλήρωσε, τα οποία μπορεί να τροποποιήσει ή να διαγράψει.

Listing Attributes:

|  |  |  |
| --- | --- | --- |
| title\* | String | Title |
| description\* | String | Description |
| pics | Base64 Array | Pictures |
| available\_from | Date | Start Time |
| available\_until\* | Date | End Time |
| location\* | Enum | Location |
| jobCategory\* | Enum | Category |
| max\_price\* | Double | Max Price |

Δημιουργία Ενός Listing

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **201 (Created)** |
| URL: **/api/users/:UID/listings** | Body: {  "status": "SUCCESS",  "msg": "**Listing Added**",  "data": {  "UID": "UserID",  "title": "Listing title",  "description": "Listing description",  "pics": [  "base64str0",  "base64str1"  ],  "available\_from": "Apr 5, 2019 12:00:00AM",  "available\_until": "Apr 6, 2019 12:00:00 AM",  "location": "HERAKLION",  "jobCategory": "Listing Category",  "max\_price": 10  },  **"resourceID": "5dda19a57b57"**  } |
| Body: {  "UID": "UserID",  "title": "Listing title",  "description": "Listing description",  "pics": ["base64str0","base64str1"],  "available\_from": "2019-04-05",  "available\_until": "2019-04-06",  "location": "HERAKLION",  "jobCategory": "Listing Category",  "max\_price": 10.0  } |

Δημιουργεί ένα νέο listing με ένα μοναδικό LID(resourceID).

Δημιουργία Ενός Listing (**Error 400**: Bad Request)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **400 (Bad Request)** |
| URL: **/api/users/:UID/listings** | Body: {  "status": "ERROR",  "msg": "**Invalid field location**",  "data": {}  } |
| Body: {  "UID": "UserID",  "title": "Listing title",  "description": "Listing description",  "pics": ["base64str0","base64str1"],  "available\_from": "2019-04-05",  "available\_until": "2019-04-06",  **"location": "invalid-location",**  "jobCategory": "Listing Category",  "max\_price": 10.0  } |

**Προσοχή:** Στην εγκυρότητα των στοιχείων και στο ποια πεδία είναι required.

Προσθήκη ενός field σε ένα Listing (**Error 501**: Not Implemented)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **501 (Not Implemented)** |
| URL: **/api/users/:UID/listings** | Body: {  "status": "ERROR",  "msg": "**Server does not support the**  **functionality required to fulfill the**  **request**",  "data": {}  } |
| Body: {  "pics": ["base64str0","base64str1"]  } |

**Προσοχή:** Στη σωστή χρήση των methods και URLs.

Get Listing

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL: /api/listings/**5dda19a57b57**  ( ή /api/users/ :UID/listings/**5dda19a57b57)** | Body: {  "status": "SUCCESS",  "msg": "Listing With LID: 5dda19a57b57",  "data": {  "UID": "UserID",  "LID": "5dda19a57b57",  "title": "Listing title",  "description": "Listing description",  "pics": [  "base64str0",  "base64str1"  ],  "available\_from": "Apr 5, 2019 12:00:00AM",  "available\_until": "Apr 6, 2019 12:00:00 AM",  "location": "HERAKLION",  "jobCategory": "Listing Category",  "max\_price": 10  }  } |
| Body: {} |

Επιστέφει το listing με LID(resourceID)=" 5dda19a57b57".

Get Listing (**Error 404**: Not Found)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **404 (Not Found)** |
| URL: /api/listings/**5dda57b57-34**  ( ή /api/users/ :UID/listings/**5dda57b57-34)** | Body: {  "status": "ERROR",  "msg": "**Listing Does Not Exist**",  "data": { }  } |
| Body: {} |

**Προσοχή:** Πάντα ζητάμε listings που υπάρχουν στο σύστημα.

Update Ενός Listing

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **PUT** | Status code: **200 (ok)** |
| URL: /api/listings/**5dda19a57b57**  ( ή /api/users/ :UID/listings/**5dda19a57b57)** | Body: {  "status": "SUCCESS",  "msg": "**Listing Edited**",  "data": {  "UID": "UserID",  "LID": "5dda19a57b57",  "title": "changed title",  "description": "changed description",  "pics": [  "base64str0",  "base64str1"  ],  "available\_from": "Apr 5, 2019 12:00:00AM",  "available\_until": "Apr 6, 2019 12:00:00 AM",  "location": "HERAKLION",  "jobCategory": "Listing Category",  "max\_price": 10  }  } |
| Body: {  "title": "changed title",  "description": "changed description",  "pics": [  "base64str0",  "base64str1"  ],  "available\_from": "2019-04-05",  "available\_until": "2019-04-06",  "location": "HERAKLION",  "jobCategory": "Listing Category",  "max\_price": 10  } |

Κάνει update τα νέα στοιχεία του listing με LID(resourceID)=" 5dda19a57b57".

Update Ενός Listing (**Error 403**: Forbidden)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **PUT** | Status code: **403 (Forbidden)** |
| URL: /api/listings/**5dda19a57b57**  ( ή /api/users/ :UID/listings/**5dda19a57b57)** | Body: {  "status": "ERROR",  "msg": "**No permission**",  "data": { }  } |
| Body: {  "title": "changed title",  "description": "changed description",  "pics": [  "base64str0",  "base64str1"  ],  "available\_from": "2019-04-05",  "available\_until": "2019-04-06",  "location": "HERAKLION",  "jobCategory": "Listing Category",  "max\_price": 10  } |

**Προσοχή**: Mόνο ο χρήστης που έκανε post το listing μπορεί να το τροποποιήσει.

Διαγραφή Ενός Listing

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **DELETE** | Status code: **200 (ok)** |
| URL: /api/listings/**5dda19a57b57**  ( ή /api/users/:UID/lsitings/ **5dda19a57b57)** | Body: {  "status": "SUCCESS",  "msg": "Listing Deleted",  "data": { }  } |
| Body: { } |

Διαγράφει το listing με LID="5dda19a57b57".

Διαγραφή Ενός Listing (**Error 403**: Forbidden)

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **DELETE** | Status code: **403 (Forbidden)** |
| URL: /api/listings/**5dda19a57b57**  ( ή /api/users/:UID/lsitings/ **5dda19a57b57)** | Body: {  "status": "ERROR",  "msg": "**No permission**",  "data": { }  } |

Πίνακας με URLs και διαθέσιμα query params:

|  |  |  |
| --- | --- | --- |
| **URL:** | **Query Params:** | **Comments:** |
| /api/users/:UID/listings | ?count=true | Returns the count of all listings |
| ?active=true | Returns all active listings |
| ?with-bids=true | Returns all listings with bids |
| ?location=Nicosia | Returns listings created for this location |
| ?min-price=10 | Returns listings with minimum price 10 |
| ?max-price=100 | Returns listings with maximum price 100 |
| ?category = categ1, categ2, … , categN | Returns listings in these categories |
| ?created-after=10-4-2019 | Returns listings created after  10-4-2019 |
| ?created-before=10-5-2019 | Return listings created before  10-5-2019 |
| ?search=keyword1, keyword2, … | Return listings that matches the keywords |

**Σημείωση:** Μπορούν να συνδυαστούν περισσότερα από ένα query params για κάθε request με την χρήση του συμβόλου(&).

Get all active listings with keywords “wall” & “painting”

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL: /api/listings**?active=true&**  **keywords=wall,painting** | Body: {  "status": "SUCCESS",  "msg": "Listing With LID: 5dda19a57b57",  "data": [{  "UID": "UserID",  "LID": "5dda19a57b57",  "title": "Listing title",  "description": "Listing description",  "pics": [  "base64str0",  "base64str1"  ],  "available\_from": "Apr 5, 2019 12:00:00AM",  "available\_until": "Apr 6, 2020 12:00:00 AM",  "location": "HERAKLION",  "jobCategory": "wall painting",  "max\_price": 10  }]  } |
| Body: {} |

Επιστρέφει ένα πίνακα με όλα τα listings που είναι active και περιέχουν τα keywords wall και painting

**Resource: Bids** ("api/users/:UID/listings/:LID/bids/:BID" ή "api/listings/:LID/bids/:BID " ή "api/bids/:BID ")

Ο κάθε professional έχει την δυνατότητα να κάνει μέχρι ένα bid σε κάθε listing από τα listings που τον αφορούνε, επίσης μπορεί να τροποποιήσει ή να διαγράψει το bid του όποτε επιθυμεί.

Bid Attributes:

|  |  |  |
| --- | --- | --- |
| solution\_decription\* | String | Description |
| price\* | Double | Bid price |
| time\_to\_fix\* | Double | Time Needed |
| When\* | Date | When the event will take place |

Δημιουργία Ενός Bid

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **201 (Created)** |
| URL: **/api/users/:UID/listings/:LID/bids** | Body: {  "status": "SUCCESS",  "msg": "**Bid Added**",  "data": {  "UID": "UserID",  "BID": "293a7fea-b771",  "LID": "ListingID",  "solution\_decription":  "price": 10,  "time\_to\_fix": 20,  "when": "Apr 5, 2019 12:00:00 AM",  "selected": false  }  **"resourceID": "293a7fea-b771"**  } |
| Body: {  "UID": "UserID",  "LID": "ListingID",  "solution\_decription":"solution\_decription",  "price": 10,  "time\_to\_fix": 20.0,  "when" : "2019-04-05",  "selected": "false"  } |

Δημιουργεί ένα νέο bid με ένα μοναδικό BID(resourceID).

Get Bid

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL:  /api/users/ :UID/listings/:LID/bids/**5dda19a57b57**  ( ή /api/bids/**5dda19a57b57**) | Body: {  "status": "SUCCESS",  "msg": "Bid With BID = 293a7fea-b771",  "data": {  "UID": "UserID",  "BID": "293a7fea-b771",  "LID": "ListingID",  "solution\_decription": "solution\_decription",  "price": 10,  "time\_to\_fix": 20,  "when": "Apr 5, 2019",  "selected": false  }  } |
| Body: {} |

Επιστέφει το listing με BID(resourceID)=" 293a7fea-b771".

Update Ενός bid

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **PUT** | Status code: **200 (ok)** |
| URL:  /api/users/:UID/listings/:LID/bids/**293a7fea-b771**  ( ή /api/bids/**293a7fea-b771**) | Body: {  "status": "SUCCESS",  "msg": "**Bid Edited**",  "data": {  "UID": "UserID",  "BID": "293a7fea-b771",  "LID": "ListingID",  "solution\_decription":"solution\_decription",  **"price": 15,**  **"time\_to\_fix": 30,**  "when": "Apr 5, 2019 12:00:00 AM",  "selected": false  }  } |
| Body: {  "UID": "UserID",  "LID": "ListingID",  "solution\_decription":"solution\_decription",  **"price": 15,**  **"time\_to\_fix": 30.0,**  "when" : "2019-04-05",  "selected": "false"  } |

Κάνει update τα νέα στοιχεία του listing με BID(resourceID)=" 293a7fea-b771".

Διαγραφή Ενός Bid

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **DELETE** | Status code: **200 (ok)** |
| URL:  /api/users/:UID/lsitings/ :LID/bids/**293a7fea-b771** | Body: {  "status": "SUCCESS",  "msg": "**Bid Deleted**",  "data": { }  } |
| Body: { } |

Διαγράφει το listing με BID="293a7fea-b771".

Πίνακας με URLs και διαθέσιμα query params:

|  |  |  |
| --- | --- | --- |
| **URL:** | **Query Params:** | **Comments:** |
| /api/users/:UID/listings/:LID/bids | ?count=true | Returns the count of all bids |
| ?selected=true | Returns all selected bids |

**Resource: Reviews** ("api/users/:UID/reviews/:RID " ή "api/reviews/:RID")

Μετά το τέλος μίας εργασίας, μπορούν οι χρήστες να βαθμολογήσουν ο ένας τον άλλον, και να εκφράσουν την γνώμη τους για την συνεργασία που είχαν.

Review Attributes:

|  |  |  |
| --- | --- | --- |
| RID\* | String | Unique Identifier |
| UID\* | String | Review from user |
| TO\_UID\* | String | Review to user |
| LID\* | String | Listing id |
| rating\* | Double | Rating from user (UID) |
| comments\* | String | Comments from user (UID) |

Δημιουργία Ενός Review

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **POST** | Status code: **201 (Created)** |
| URL: **/api/users/:UID/reviews** | Body: {  "status": "SUCCESS",  "msg": "**Review Added**",  "data": {  "UID": "UserID",  "TO\_UID": "TO\_UserID",  "LID": "ListingID",  "RID": "ef3ac4d2-7344",  "rating": 5,  "comments": "excellent job"  }  **"resourceID": " ef3ac4d2-7344"**  } |
| Body: {  "UID": "UserID",  "TO\_UID": "TO\_UserID",  "LID": "ListingID",  "RID": "ef3ac4d2-7344",  "rating": 5,  "comments": "excellent job"  } |

Δημιουργεί ένα νέο review με ένα μοναδικό RID(resourceID).

Get Review

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL:  /api/users/ :UID/reviews/**ef3ac4d2-7344**  ( ή /api/reviews/**ef3ac4d2-7344**) | Body: {  "status": "SUCCESS",  "msg": "Review With RID = ef3ac4d2-7344",  "data": {  "UID": "UserID",  "TO\_UID": "TO\_UserID",  "LID": "ListingID",  "RID": "ef3ac4d2-7344",  "rating": 5,  "comments": "excellent job"  }  } |
| Body: {} |

Επιστέφει το review με RID(resourceID)="ef3ac4d2-7344".

Update Ενός bid

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **PUT** | Status code: **200 (ok)** |
| URL: /api/users/:UID/reviews/**ef3ac4d2-7344**  ( ή /api/reviews/**ef3ac4d2-7344**) | Body: {  "status": "SUCCESS",  "msg": "**Review Edited**",  "data": {  "UID": "UserID",  "TO\_UID": "TO\_UserID",  "LID": "ListingID",  "RID": "ef3ac4d2-7344 ",  **"rating": 4.5,**  "comments": "excellent job"  }  } |
| Body: {  "UID": "UserID",  "TO\_UID": "TO\_UserID",  "LID": "ListingID",  "RID": "ef3ac4d2-7344 ",  **"rating": 4.5,**  "comments": "excellent job"  } |

Κάνει update τα νέα στοιχεία του reviewμε RID(resourceID)="ef3ac4d2-7344".

Διαγραφή Ενός Review

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **DELETE** | Status code: **200 (ok)** |
| URL:  /api/users/:UID/reviews/**ef3ac4d2-7344** | Body: {  "status": "SUCCESS",  "msg": "**Review Deleted**",  "data": { }  } |
| Body: { } |

Διαγράφει το reviewμε RID="ef3ac4d2-7344".

**Resource: Events** ("api/events/:UID")

Για τους professionals φτιάχνετε δυναμικά το πρόγραμμα τους με βάση τα bids που έκαναν και επιλέγηκαν από τους customers. Δεν αποθηκεύετε κάτι στην βάση δεδομένων, τα δεδομένα εξάγονται δυναμικά με βάση τα υπάρχοντα.

Get Events

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL:  /api/events/ :UID | Body: {  "status": "SUCCESS",  "msg": "User events",  "data": [{  "UID": "UserID"  "date": "Dec 31, 2022"  "LID": "ca390966-d29b"  "duration": 54  "wonBID": "23dc44b1-3658-4a9a"  "title": "listing-title"  "descr": "listing-descr"  }]  } |
| Body: {} |

Επιστέφει όλα τα events του χρήστη με UID = " UserID ".

**Resource: Statistics** ("api/statistics/:UID" ή "api/users/:UID/ statistics")

Για τους professionals φτιάχνονται δυναμικά στατιστικά στοιχεία που αφορούνε τα bids και τα reviews που έκαναν, τα listings που επιλέγηκαν, καθώς και τα λεφτά που κέρδισαν. Τα στατιστικά είναι για τον τελευταίο χρόνο, τελευταίο μήνα, τελευταία βδομάδα ή για όλο το διάστημα που ο professional είναι εγγεγραμμένος στην σελίδα. By default, αν γίνει κάποιο request στο γενικό url επιστρέφονται τα στατιστικά του τελευταίου χρόνου.

Get Statistics

|  |  |
| --- | --- |
| **Request:** | **Response:** |
| Method: **GET** | Status code: **200 (ok)** |
| URL:  /api/statistics/ :UID | Body: {  "status": "SUCCESS",  "msg": "User Year statistics",  "data": {  "Money": {  "January ": 95,  "February": 150  },  "Reviews": {  "January ": 1  },  "Listings": {},  "Bids": {  "January ": 3,  "February": 10  }  }  } |
| Body: {} |

Επιστέφει τα στατιστικά του χρήστη για τον τελευταίο χρόνο.