

# ΕΦΑΡΜΟΓΗ ΕΠΑΓΓΕΛΜΑΤΙΚΗΣ ΔΙΚΤΥΩΣΗΣ (ΤΥΠΟΥ LINKEDIN)

API

## ΠΕΡΙΓΡΑΦΗ:

Σκοπό του παρόντος εγγράφου αποτελεί η περιγραφή του API της εφαρμογής επαγγελματικής δικτύωσης και των λειτουργιών της.

Η εφαρμογή επαγγελματικής δικτύωσης είναι διαδικτυακή εφαρμογή που επιτρέπει στους χρήστες να δημιουργήσουν ιδιωτικά προσωπικά προφίλ στα οποία μπορούν να διαχειρίζονται πληροφορίες σχετικά με αυτούς αλλά και να επικοινωνούν με άλλα άτομα αιτώντας συνδέσεις μαζί τους.

Ο κάθε χρήστης έχει τη δυνατότητα να δημιουργήσει ένα προφίλ στο οποίο θα αποθηκεύσει πληροφορίες σχετικά με την εκπαίδευσή του, της επαγγελματική του κατάσταση αλλά και τις ικανότητες του. Μέσω της πλατφόρμας αυτής, μπορεί ακόμα να αναρτήσει δημοσιεύσεις σχετικά με θέματα που τον αφορούν ή τον προβληματίζουν. Παράλληλα, έχει τη δυνατότητα να μοιραστεί τις πληροφορίες του με άλλους χρήστες της εφαρμογής που γνωρίζει, πραγματοποιώντας μια “σύνδεση” (Connection) με αυτούς. Ο κάθε χρήστης μπορεί να βλέπει όλες τις πληροφορίες όλων των συνδεδεμένων, με αυτόν, χρηστών αλλά και να επικοινωνεί με αυτούς στέλνοντας τους άμεσα μηνύματα. Έχει, ακόμα, πρόσβαση σε όλες τις αναρτήσεις τους και μπορεί, αν το επιθυμεί, να εκδηλώσει ενδιαφέρον σε κάποιες από αυτές ή να σχολιάσει μια προσωπική άποψη. Τέλος, κάθε χρήστης μπορεί να ανεβάσει αγγελίες για εργασία, οι οποίες φαίνονται σε όσους έχουν συνδεθεί με αυτόν, ή να ολοκληρώσει μια αίτηση για κάποια αγγελία κάποιας σύνδεσής του.

Η εφαρμογή έχει, από την εγκατάστασή της, έναν τουλάχιστον διαχειριστή ο οποίος έχει τη δυνατότητα να διαχειριστεί τους χρήστες και να εξάγει τα δεδομένα τους σε εξωτερικό αρχείο.

## Οι ρόλοι της εφαρμογής:

### → Διαχειριστής

- ◆ Διαχείριση χρηστών εφαρμογής
- ◆ Εξαγωγή δεδομένων χρηστών

### → Επαγγελματίας

- ◆ Διαχείριση πληροφοριών σχετικά με την εκπαίδευση και την επαγγελματική τους εμπειρία
- ◆ Αποστολή αιτημάτων σύνδεσης σε άλλους χρήστες
- ◆ Αποδοχή/Απόρριψη αιτημάτων σύνδεσης
- ◆ Επισκόπηση και διαχείριση δικτύου συνδέσεων
- ◆ Ανάρτηση άρθρων με εικόνες και βίντεο

- ◆ Ανάρτηση και απάντηση σε αγγελίες
- ◆ Προβολή στο χρονολόγιο άρθρων που ανάρτησαν οι συνδεδεμένοι με αυτούς επαγγελματίες και σχολιασμός σε αυτά
- ◆ Σημείωση ενδιαφέροντος για συγκεκριμένες αναρτήσεις
- ◆ Ειδοποίηση ενδιαφέροντος άλλων χρηστών στις αναρτήσεις τους
- ◆ Διεξαγωγή ιδιωτικών συζητήσεων με άλλους χρήστες στο δίκτυό τους
- ◆ Προβολή προφίλ άλλων χρηστών
- ◆ Διαχείριση ρυθμίσεων σύνδεσής τους

## ΥΛΟΠΟΙΗΣΗ:

### Σχήμα Βάσης:

Users	
<b>ID</b>	BIGINT(10)
FirstName	VARCHAR(30)
LastName	VARCHAR(30)
Email	VARCHAR(50)
Password	VARCHAR(30)
Phone	VARCHAR(15)
IsAdmin	BOOLEAN

Jobs	
<b>ID</b>	BIGINT(10)
Creator	BIGINT(10)
Title	VARCHAR(30)
Description	VARCHAR(200)
Date	DATE

Friends	
<b>User1</b>	BIGINT(10)
<b>User2</b>	BIGINT(10)

FriendRequest	
<b>Sender</b>	BIGINT(10)
<b>Receiver</b>	BIGINT(10)

Media	
<b>PostID</b>	BIGINT(10)
<b>Link</b>	VARCHAR(200)

Pictures	
<b>UserID</b>	BIGINT(10)
<b>Link</b>	VARCHAR(200)

Messages	
<b>UserFrom</b>	BIGINT(10)
<b>UserTo</b>	BIGINT(10)
Content	VARCHAR(200)
<b>Timestamp</b>	DATE

Likes	
<b>UserID</b>	BIGINT(10)
<b>PostID</b>	BIGINT(10)
Timestamp	TIMESTAMP

Comments	
<b>UserID</b>	BIGINT(10)
<b>PostID</b>	BIGINT(10)
Text	VARCHAR(50)
Timestamp	TIMESTAMP

**JobApplications**

<b>Applicant</b>	BIGINT(10)
<b>Job</b>	BIGINT(10)

**Notifications**

<b>ID</b>	BIGINT(10)
Timestamp	TIMESTAMP
Category	TINYINT(1)
UserFrom	BIGINT(10)
UserTo	BIGINT(10)
Post	BIGINT(10)

**Posts**

<b>ID</b>	BIGINT(10)
UserID	BIGINT(10)
Text	VARCHAR(200)
Date	DATE

**JobRequirements**

<b>JobID</b>	BIGINT(10)
<b>Skill</b>	VARCHAR(20)

**Experience**

UserID	BIGINT(10)
<b>ExperienceID</b>	BIGINT(10)
CompanyTitle	VARCHAR(30)
Position	VARCHAR(30)
DateFrom	DATE
DateTo	DATE

**Education**

UserID	BIGINT(10)
<b>EducationID</b>	BIGINT(10)
Degree	VARCHAR(30)
Institution	VARCHAR(30)
YearFrom	YEAR(4)
YearTo	YEAR(4)

**Skills**

UserID	BIGINT(10)
<b>SkillID</b>	BIGINT(10)
Skill	VARCHAR(20)

Βοηθητικές Κλάσεις:

<b>Article</b>	
postId	long
creator	long
text	String
likes	List<LikesEntity>
comments	List<CommentsEntity>

<b>JobSkillsAlike</b>	
jobId	long
skillsAlike	int

MyXMLObject	
userId	long
education	List<EducationEntity>
experience	List<ExperienceEntity>
connections	List<UserInfo>
posts	List<PostsEntity>
likes	List<LikesEntity>
comments	List<CommentsEntity>

UserInfo	
userId	long
firstName	String
lastName	String
email	String
image	String

## Κλήσεις στο API:

### Login (GET)

https://localhost:8443/api/rest/user/login/{email}/{password}	
Request	
email	String
password	String
Response	
userId	long
isAdmin	boolean

### *Περιγραφή:*

Κάθε φορά που ο χρήστης προσπαθεί να μπει στο λογαριασμό του, θα πρέπει πρώτα να επιβεβαιωθούν τα στοιχεία του. Η κλήση δέχεται το όνομα χρήστη και το κωδικό του χρήστη και επιστρέφει το μοναδικό αριθμό του μαζί με μια σημαία που υποδεικνύει αν είναι διαχειριστής ή όχι, αν ήταν πετυχημένη η είσοδος, αλλιώς επιστρέφει 0.

## Register (POST)

https://localhost:8443/api/rest/user/register	
Request	
password	String
passConfirm	String
firstName	String
lastName	String
email	String
phone	String
Response	
userId	long

### Περιγραφή:

Κάθε φορά που ένας νέος χρήστης θέλει να συνδεθεί στην εφαρμογή πρέπει πρώτα να δημιουργήσει λογαριασμό. Ο χρήστης εισάγει τα προσωπικά του στοιχεία, όπως το όνομα, το επώνυμο και τον κωδικό του και στέλνει το αίτημα. Αν η διαδικασία της εισαγωγής ολοκληρώθηκε δίχως προβλήματα τότε επιστρέφεται στην απάντηση ο μοναδικός αριθμός του χρήστη που δημιουργήθηκε. Αν υπάρχει ήδη χρήστης με το συγκεκριμένο email τότε επιστρέφεται 0. Αν ο κωδικός επιβεβαίωσης είναι διαφορετικός από τον κωδικό που εισήγαγε ο χρήστης, επιστρέφεται -1. Σε άλλη περίπτωση επιστρέφεται -2.

## UsersList (GET)

https://localhost:8443/api/rest/user/getusers	
Request	
Response	
usersList	List<Users>

### Περιγραφή:

Ο διαχειριστής της εφαρμογής έχει τη δυνατότητα να δει όλους τους χρήστες της εφαρμογής και να εξάγει τα στοιχεία τους. Για να δει τους χρήστες, στέλνει αίτημα στη παραπάνω διεύθυνση και επιστρέφεται μια λίστα με όλους τους χρήστες και τα στοιχεία τους.

## GetInfo (GET)

https://localhost:8443/api/rest/user/getinfo/{userId}	
Request	
userId	long
Response	
userInfo	UserInfo

### Περιγραφή:

Προκειμένου να μπορέσει ο χρήστης να δει τα στοιχεία του ίδιου είτε άλλων χρηστών θα πρέπει πρώτα να μετατρέψει τον μοναδικό αριθμό τους στα στοιχεία τους. Η κλήση δέχεται ως αίτημα τον μοναδικό αριθμό ενός χρήστη και επιστρέφει τα στοιχεία του, όπως όνομα, επίθετο και email.

## GetPosts (GET)

https://localhost:8443/api/rest/post/getposts/{userId}	
Request	
userId	long
Response	
articles	List<Articles>

### Περιγραφή:

Ο χρήστης, μέσω της εφαρμογής, έχει τη δυνατότητα να διαβάσει και να αντιδράσει (να σχολιάσει ή να εκδηλώσει ενδιαφέρον) σε άρθρα που ανεβάζουν άλλοι χρήστες στο δίκτυο. Η εμφάνιση των άρθρων είναι εξειδικευμένη ως προς τα ενδιαφέροντα και το δίκτυο του χρήστη, με σκοπό τη καλύτερη δυνατή προσωπική εμπειρία.

Αρχικά, εμφανίζονται άρθρα στα οποία έχουν αντιδράσει χρήστες με παρόμοια ενδιαφέροντα με τον ίδιο. Αυτό αυξάνει την πιθανότητα να ενδιαφέρει και τον ίδιο το συγκεκριμένο άρθρο. Έπειτα, εμφανίζονται όλα τα άρθρα που έχουν ανεβάσει οι,

συνδεδεμένοι με αυτόν, χρήστες, αλλά και αυτά στα οποία οι τελευταίοι έχουν εκδηλώσει ενδιαφέρον.

### **Βήματα:**

- Υλοποιήθηκε ένα διάνυσμα για όλες τις αντιδράσεις όλων των χρηστών σε κάθε δημοσίευση. Το διάνυσμα περιέχει τιμές από 0 έως 2, ανάλογα με το αν ο χρήστης έχει κάνει σχόλιο, έχει εκδηλώσει ενδιαφέρον, έχει κάνει και τα δύο ή τίποτα από τα δύο. Το διάνυσμα αυτό δημιουργείται στην αρχή της εκτέλεσης του προγράμματος και από εκεί και πέρα απλά ανανεώνεται.
- Με βάση το παραπάνω διάνυσμα, επιλέγονται οι 5 κοντινότεροι γείτονες, δηλαδή οι χρήστες που έχουν τα περισσότερα κοινά ενδιαφέροντα με το χρήστη.
- Για κάθε έναν από τους γείτονες αυτούς, επιλέγονται όλα τα άρθρα στα οποία έχουν εκδηλώσει αυτοί ενδιαφέρον ή έχουν σχολιάσει, στα οποία, ωστόσο, δεν έχει αντιδράσει ο αρχικός χρήστης.
- Από όλα αυτά τα άρθρα, εμφανίζονται μόνο όσα είναι από τις τελευταίες δύο μέρες.
- Έπειτα, εμφανίζονται με χρονολογική σειρά, όλα τα άρθρα των, συνδεδεμένων με αυτόν, χρηστών.

### **InsertPost (POST)**

https://localhost:8443/api/rest/post/insert	
Request	
userId	long
text	String
Response	
postId	long

### **Περιγραφή:**

Ο χρήστης έχει τη δυνατότητα να δημοσιεύσει ένα άρθρο ή ένα κείμενο, που να περιέχει τη γνώμη του πάνω σε συγκεκριμένα θέματα ή απλά ενημερώσεις στα άτομα με τα οποία έχει συνδεθεί. Οι δημοσιεύσεις περιέχουν μόνο κείμενο και η κλήση επιστρέφει τον μοναδικό κωδικό της δημοσίευσης.



## Like (POST)

https://localhost:8443/api/rest/post/like	
Request	
userId	long
postId	long
Response	
notificationId	long

### Περιγραφή:

Ο χρήστης μπορεί να εκδηλώσει ενδιαφέρον (από εδώ και στο εξής like) σε μια δημοσίευση άλλου χρήστη, δηλώνοντας ότι του άρεσε ή ότι του φάνηκε ενδιαφέρον. Από τη στιγμή που ένας χρήστης κάνει like, μετά δε μπορεί να ξανακάνει ή να το πάρει πίσω. Ο συνολικός αριθμός like μιας δημοσίευσης φαίνεται στη κεντρική σελίδα των δημοσιεύσεων. Μετά το like, στέλνεται αντίστοιχη ειδοποίηση στο χρήστη που είχε ανεβάσει τη δημοσίευση (ο μοναδικός αριθμός της οποίας επιστρέφεται ως απάντηση), ενώ ενημερώνονται κατάλληλα οι αντίστοιχες global δομές.

## Comment (POST)

https://localhost:8443/api/rest/post/comment	
Request	
userId	long
postId	long
text	String
Response	
notificationId	long

### Περιγραφή:

Ο χρήστης μπορεί να σχολιάσει, κάτω από μια δημοσίευση κάποιου άλλου χρήστη, τη γνώμη του πάνω στο ζήτημα που τίγεται. Το σχόλιο του χρήστη δεν μπορεί να αφαιρεθεί, μετά το σχολιασμό, και θα φαίνεται σε όλους, ανεξάρτητα από το αν είναι συνδεδεμένοι με αυτόν. Μετά το σχόλιο, στέλνεται αντίστοιχη ειδοποίηση στο χρήστη που είχε ανεβάσει τη δημοσίευση (ο μοναδικός αριθμός της οποίας επιστρέφεται ως απάντηση), ενώ ενημερώνονται κατάλληλα οι αντίστοιχες global δομές.

## GetConnections (GET)

https://localhost:8443/api/rest/user/getconnections/{userId}	
Request	
userId	long
Response	
users	List<UserInfo>

### Περιγραφή:

Ο κάθε χρήστης, έχει ανά πάσα στιγμή, τη δυνατότητα να δει όλα τα άτομα με τα οποία έχει συνδεθεί. Η παραπάνω κλήση υλοποιεί αυτή τη λειτουργία, επιστρέφοντας για κάθε χρήστη όλους τους συνδεδεμένους, με αυτόν, χρήστες.

## GetSearchResults (GET)

https://localhost:8443/api/rest/user/search/{query}	
Request	
searchQuery	String
Response	
results	List<UserInfo>

### Περιγραφή:

Η εφαρμογή δίνει τη δυνατότητα αναζήτησης κάποιου ατόμου ανάμεσα σε όλους τους χρήστες. Ο χρήστης γράφει ένα μέρος, ή όλο το όνομα του ατόμου που επιθυμεί να αναζητήσει και επιστρέφεται μια λίστα από τους χρήστες, τα ονόματα των οποίων ταιριάζουν στην αναζήτηση.

## GetJobs (GET)

https://localhost:8443/api/rest/job/getjobs/{userId}	
Request	
userId	long
Response	
fromFriends	List<Jobs>
alike	List<Jobs>
fromNeighbors	List<Jobs>

### Περιγραφή:

Ο χρήστης, μέσω της εφαρμογής, έχει τη δυνατότητα να δει και να κάνει αίτηση σε αγγελίες για εργασία. Οι αγγελίες που εμφανίζονται σε κάθε χρήστη είναι εξειδικευμένες και προκύπτουν τόσο από τον ίδιο και τις ικανότητές του, όσο και από το δίκτυο του. Αν κάνει αίτηση σε κάποια αγγελία τότε αυτή σταματάει να εμφανίζεται, και κάποια άλλη παίρνει τη θέση της.

Αρχικά, του εμφανίζονται όλες οι αγγελίες που έχουν δημοσιοποιηθεί από άτομα τα οποία έχουν συνδεθεί μαζί του. Η αγγελία θα πρέπει να έχει δημοσιοποιηθεί το πολύ τρεις μήνες πριν τη ημερομηνία προβολής της. Αυτό γίνεται για να φιλτράρονται τα αποτελέσματα και να μην επιβαρύνουν τη σελίδα ξεπερασμένες αγγελίες.

Στη συνέχεια, εμφανίζονται αγγελίες οι οποίες έχουν απαιτήσεις παρόμοιες με τις ικανότητες του χρήστη. Όσο πιο πολλά κοινά χαρακτηριστικά υπάρχουν, τόσο πιο ψηλά εμφανίζεται η αγγελία. Σημειώνεται ότι εμφανίζονται το πολύ πέντε αγγελίες αυτής της κατηγορίας κάθε φορά.

Τέλος, εμφανίζονται αγγελίες, στις οποίες έχουν δηλώσει άτομα με παρόμοια ενδιαφέροντα με το χρήστη. Το αποτέλεσμα αυτό προκύπτει μέσω του αλγορίθμου Κοντινότερου Γείτονα (5-Nearest Neighbor) στον οποίο κάθε χρήστης απεικονίζεται με βάση σε ποιες αγγελίες έχει κάνει αίτηση και σε ποιες όχι. Βρίσκονται οι πέντε χρήστες που έχουν κάνει τις πιο παρόμοιες αιτήσεις σε αγγελίες με το χρήστη και από αυτούς εμφανίζονται οι αγγελίες στις οποίες έχουν δηλώσει αυτοί αλλά όχι ο χρήστης.

### Βήματα:

(Αγγελιών μέσω συνδέσεων)

- Εύρεση και εμφάνιση όλων των αγγελιών που δημιουργήθηκαν από συνδεδεμένους χρήστες τους τελευταίους 3 μήνες.

(Αγγελιών μέσω κοινών χαρακτηριστικών)

- Ανάκτηση των χαρακτηριστικών του χρήστη.
- Για κάθε αγγελία συνολικά, εύρεση των απαιτήσεων της και εύρεση των κοινών χαρακτηριστικών που ζητάει η αγγελία με αυτά που έχει δηλώσει ο χρήστης.
- Επιλογή των πέντε αγγελιών με τα περισσότερα κοινά σημεία.
- Επιβεβαίωση ότι καμία από αυτές δεν έχει ήδη εμφανιστεί.

(Αγγελιών μέσω του αλγορίθμου κοντινότερου γείτονα)

- Εύρεση πέντε κοντινότερων γειτόνων μέσω του αλγορίθμου.
- Για κάθε έναν από αυτούς, εύρεση των αγγελιών στις οποίες έχουν κάνει αίτηση αυτοί αλλά όχι ο χρήστης.
- Επιβεβαίωση ότι καμία από αυτές δεν έχει ήδη εμφανιστεί.

Αναγνωρίζεται ότι λόγω μεγάλου πλήθους δεδομένων, σε μεταγενέστερο στάδιο, ο αλγόριθμος κοντινότερου γείτονα θα είναι αρκετά πιο αργός απ' ό τι άλλοι παρόμοιοι αλγόριθμοι και θα προκαλεί πλέον αισθητές καθυστερήσεις. Για τους σκοπούς της εφαρμογής, τώρα, είναι ωστόσο ικανοποιητικός.

## JobApplication (POST)

https://localhost:8443/api/rest/job/apply	
Request	
userId	long
jobId	long
Response	
success	boolean

### Περιγραφή:

Ο χρήστης μπορεί να στείλει μια αίτηση για μια από τις αναγραφόμενες δουλειές. Από τη στιγμή που πατήσει την αποστολή της αίτησης, τα στοιχεία του αποστέλλονται στον δημιουργό της αίτησης. Η διαδικασία είναι μη αναστρέψιμη.

**InsertJob (POST)**

https://localhost:8443/api/rest/job/insert	
Request	
userId	long
jobTitle	String
jobDescription	String
skills	List<String>
Response	
jobId	long

*Περιγραφή:*

Κάθε χρήστης μπορεί να ανεβάσει μια αγγελία για μια εργασία. Στη αγγελία αυτή τοποθετεί τον τίτλο της θέσης, μια περιγραφή και όλα τα στοιχεία που πρέπει να έχει ο υποψήφιος για να πραγματοποιήσει αίτηση. Η αγγελία αποθηκεύεται και εμφανίζεται πλέον σε όλους τους ενδιαφερόμενους. Μετά την δημιουργία της, ανανεώνονται και οι global δομές.

**GetMyApplicants (GET)**

https://localhost:8443/api/rest/job/getapplicants/{jobId}	
Request	
jobId	long
Response	
users	List<UserInfo>

*Περιγραφή:*

Ο κάθε δημιουργός μιας αγγελίας μπορεί να δει τα στοιχεία των ατόμων που έχουν κάνει αίτηση σε κάποια αγγελία του. Μέσω αυτής της κλήσης, που δέχεται τον μοναδικό αριθμό της κάθε αγγελίας, επιστρέφεται μια λίστα με όλους τους χρήστες που έχουν κάνει αίτηση.

## GetMyApplications (GET)

https://localhost:8443/api/rest/job/getmyapplications/{userId}	
Request	
userId	long
Response	
jobs	List<Jobs>

### Περιγραφή:

Όταν κάποιος χρήστης στέλνει μια αίτηση για μια θέση εργασίας, τότε δε μπορεί να ξαναστείλει. Η συγκεκριμένη κλήση επιστρέφει όλες της αγγελίες στις οποίες έχει κάνει αίτηση ο χρήστης ώστε να μη μπορεί να ξανακάνει σε κάποια που έχει ήδη κάνει.

## GetMyJobs (GET)

https://localhost:8443/api/rest/job/getmyjobs/{userId}	
Request	
userId	long
Response	
jobs	List<Jobs>

### Περιγραφή:

Ο χρήστης μπορεί να δει όλες τις αγγελίες τις οποίες ο ίδιος είχε ανεβάσει, ώστε να μπορεί να εξάγει τα στοιχεία των ατόμων που έκαναν αίτηση είτε να αλλάξει τα στοιχεία της αγγελίας.

## EditJob (POST)

https://localhost:8443/api/rest/job/edit	
Request	
jobId	long
jobTitle	String

jobDescription	String
<i>Response</i>	
success	boolean

*Περιγραφή:*

Ο κάθε δημιουργός μιας αγγελία μπορεί να αλλάξει τα στοιχεία της αγγελίας, όπως τον τίτλο ή τη περιγραφή της θέσης. Το σύνολο των απαιτούμενων χαρακτηριστικών δεν αλλάζει.

**SendConnectionRequest (POST)**

<b>https://localhost:8443/api/rest/connection/sendrequest</b>	
<i>Request</i>	
sender	long
receiver	long
<i>Response</i>	
notificationId	long

*Περιγραφή:*

Προκειμένου δύο άτομα να συνδεθούν (connection) θα πρέπει ένα από τα δύο να στείλει πρώτα ένα αίτημα σύνδεσης στο άλλο. Μαζί με το αίτημα, αποστέλνεται μια ειδοποίηση στον παραλήπτη, ο οποίος μπορεί να επισκεφθεί το προφίλ του χρήστη που του έστειλε το αίτημα και να αποφασίσει αν θα το δεχτεί ή όχι.

**GetConnectionRequests (GET)**

<b>https://localhost:8443/api/rest/connection/getrequests/{userId}</b>	
<i>Request</i>	
userId	long
<i>Response</i>	

senders	List<UserInfo>
---------	----------------

### Περιγραφή:

Προκειμένου να δει το προφίλ ενός χρήστη, ο χρήστης της εφαρμογής θα πρέπει να γνωρίζει αν είναι συνδεδεμένοι, και αν δεν είναι, αν του έχει αποστείλει αίτημα σύνδεσης. Σε περίπτωση που ο χρήστης του έχει στείλει αίτημα σύνδεσης, τότε έχει την επιλογή ανάμεσα στην αποδοχή και την απόρριψή του. Η επιστροφή της κλήσης είναι μια λίστα με όλους τους χρήστες που έχουν στείλει αίτημα στο χρήστη.

### AcceptConnection (POST)

https://localhost:8443/api/rest/connection/accept	
Request	
userIdFrom	long
userIdTo	long
Response	
notificationId	long

### Περιγραφή:

Όταν κάποιος χρήστης στέλνει αίτημα σύνδεσης σε έναν άλλον, εφόσον ο τελευταίος το επιθυμεί μπορεί να το δεχτεί και να συνδεθούν οι δύο χρήστες μεταξύ τους. Μαζί με την αποδοχή, αποστέλνεται και μια ειδοποίηση στον χρήστη (ο μοναδικός αριθμός της οποίας στέλνεται ως απάντηση) που έστειλε το αίτημα, πως το αίτημα του έγινε δεκτό.

### RejectConnection (POST)

https://localhost:8443/api/rest/connection/reject	
Request	
userIdFrom	long
userIdTo	long



<i>Response</i>	
success	boolean

### Περιγραφή:

Όταν κάποιος χρήστης στέλνει αίτημα σύνδεσης σε έναν άλλον, εφόσον ο τελευταίος δεν τον γνωρίζει, μπορεί να το απορρίψει και να μη συνδεθούν οι δύο χρήστες μεταξύ τους.

## GetPersonalInfo (GET)

<b>https://localhost:8443/api/rest/user/getpersonalinfo/{userId}</b>	
<i>Request</i>	
userId	long
<i>Response</i>	
experience	List<Experience>
education	List<Education>
skills	List<Skills>

### Περιγραφή:

Ο κάθε χρήστης μπορεί να δει για τον ίδιο, ή για όλους τους συνδεδεμένους με αυτόν χρήστες, όλα τα στοιχεία του βιογραφικού του. Δηλαδή έχει πρόσβαση στα στοιχεία που έχει δηλώσει ως εκπαίδευση, επαγγελματική εμπειρία ή δεξιότητες, τα οποία επιστρέφονται ως μια λίστα το καθένα.

## PostExperience (POST)

<b>https://localhost:8443/api/rest/experience/post</b>	
<i>Request</i>	
userId	long
companyTitle	String
position	String

dateFrom	java.sql.Date
dateTo	java.sql.Date
<i>Response</i>	
experienceId	long

### Περιγραφή:

Ο χρήστης μπορεί να δημοσιεύσει στο προσωπικό του προφίλ την επαγγελματική του εμπειρία, δηλαδή για κάθε θέση, το τίτλο, την εταιρεία και τις ημερομηνίες που εργάστηκε εκεί. Η κάθε προσθήκη έχει μοναδικό κωδικό, ο οποίος επιστρέφεται και ως απάντηση.

### PostEducation (POST)

<b>https://localhost:8443/api/rest/education/post</b>	
<i>Request</i>	
userId	long
institution	String
degree	String
yearFrom	String
yearTo	String
<i>Response</i>	
educationId	long

### Περιγραφή:

Ο χρήστης μπορεί να δημοσιεύσει στο προσωπικό του προφίλ την εκπαιδευτική του σταδιοδρομία, δηλαδή για κάθε πτυχίο, το είδος του, το ινστιτούτο από όπου το απέσπασε (σχολείο, πανεπιστήμιο κλπ) και τις χρονιές που σπούδασε εκεί. Η κάθε προσθήκη έχει μοναδικό κωδικό, ο οποίος επιστρέφεται και ως απάντηση.

## PostSkill (POST)

https://localhost:8443/api/rest/skill/post	
Request	
userId	long
skill	String
Response	
skillId	long

### Περιγραφή:

Ο χρήστης μπορεί να δημοσιεύσει στο προσωπικό του προφίλ τις προσωπικές του δεξιότητες, δηλαδή τις ικανότητες του σε τομείς που θα μπορέσουν να τον οδηγήσουν στη πρόσληψή του. Η κάθε προσθήκη έχει μοναδικό κωδικό, ο οποίος επιστρέφεται και ως απάντηση.

## UpdateExperience (POST)

https://localhost:8443/api/rest/experience/update	
Request	
experienceId	long
companyTitle	String
position	String
dateFrom	java.sql.Date
dateTo	java.sql.Date
Response	
success	boolean

### Περιγραφή:

Ο χρήστης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία που συντελούν το βιογραφικό του στην εφαρμογή. Πιο συγκεκριμένα μπορεί να επεξεργαστεί κάποια δουλειά με την οποία ασχολήθηκε, αλλάζοντας τη διάρκεια ή τα στοιχεία της θέσης.

## UpdateEducation (POST)

https://localhost:8443/api/rest/education/update	
Request	
educationId	long
institution	String
degree	String
yearFrom	String
yearTo	String
Response	
success	boolean

### Περιγραφή:

Ο χρήστης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία που συντελούν το βιογραφικό του στην εφαρμογή. Πιο συγκεκριμένα μπορεί να επεξεργαστεί την εκπαίδευσή του, αν το θεωρήσει απαραίτητο, αλλάζοντας τη διάρκεια ή την ονομασία του ινστιτούτου.

## UpdateSkill (POST)

https://localhost:8443/api/rest/skill/update	
Request	
skillId	long
skill	String
Response	
success	boolean

### Περιγραφή:

Ο χρήστης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία που συντελούν το βιογραφικό του στην εφαρμογή. Πιο συγκεκριμένα μπορεί να επεξεργαστεί κάποια από τις δεξιότητες του, σε περίπτωση που το κρίνει αναγκαίο.

## SendMessage (POST)

https://localhost:8443/api/rest/message/send	
Request	
userFrom	long
userTo	long
message	String
Response	
success	boolean

### Περιγραφή:

Η εφαρμογή δίνει τη δυνατότητα σε συνδεδεμένους, μεταξύ τους, χρήστες να επικοινωνήσουν μεταξύ τους με άμεσα μηνύματα. Τα μηνύματα αυτά έχουν ένα αποστολέα και έναν αποδέκτη. Η συγκεκριμένη κλήση στέλνει το μήνυμα με το περιεχόμενό του και την ώρα αποστολής στον παραλήπτη.

## GetMessages (GET)

https://localhost:8443/api/rest/message/get/{userId}/{friendId}	
Request	
userId	long
friendId	long
Response	
messages	List<Message>

### Περιγραφή:

Ο χρήστης μπορεί να δει τα μηνύματα που του έχουν στείλει άλλοι χρήστες και να απαντήσει, είτε να τα αφήσει ασχολίαστα. Για την εμφάνιση των μηνυμάτων, καλείται η παραπάνω κλήση, η οποία επιστρέφει μια λίστα αντικειμένων της κλάσης Message τα οποία περιέχουν τη συζήτηση των δύο ατόμων.

## GetNotifications (GET)

https://localhost:8443/api/rest/notifications/get/{userId}	
Request	
userId	long
Response	
notifications	List<Notifications>

### Περιγραφή:

Σε κάθε αλληλεπίδραση άλλων χρηστών με τον χρήστη, του αποστέλνεται μια ειδοποίηση, ένα μικρό μήνυμα, πρακτικά, που περιέχει περιορισμένες πληροφορίες σχετικά με το είδος της ειδοποίησης, το χρήστη που αλληλοεπίδρασε με τον χρήστη και αν ήταν like ή σχόλιο, τη δημοσίευση στην οποία πραγματοποιήθηκε η εκδήλωση.

## GetConversations (GET)

https://localhost:8443/api/rest/user/getconversations/{userId}	
Request	
userId	long
Response	
users	List<UserInfo>

### Περιγραφή:

Προτού δει τα μηνύματα που έχει με κάποιον χρήστη, θα πρέπει ο χρήστης να μπορεί να δει όλα τα άτομα με τα οποία επικοινωνεί τη παρούσα χρονική στιγμή. Η παραπάνω κλήση επιστρέφει μια λίστα από όλα τα άτομα με τα οποία ο χρήστης έχει ανταλλάξει έστω και ένα μήνυμα.

## UpdateEmail (POST)

https://localhost:8443/api/rest/user/email/update	
Request	
userId	long
email	String
password	String
Response	
userId	long

### Περιγραφή:

Στα πλαίσια της επεξεργασίας των δεδομένων του χρήστη, μπορεί να αλλάξει και να ανανεώσει την ηλεκτρονική του διεύθυνση. Μαζί με τη νέα διεύθυνση, απαιτείται και ο κωδικός του χρήστη για πρόσθετη ασφάλεια.

## UpdatePassword (POST)

https://localhost:8443/api/rest/user/password/update	
Request	
userId	long
oldPassword	String
newPassword	String
Response	

### Περιγραφή:

Στα πλαίσια της επεξεργασίας των δεδομένων του χρήστη, μπορεί να αλλάξει και να ανανεώσει τον κωδικό του λογαριασμού του. Μαζί με το νέο κωδικό, απαιτείται και ο παλιός κωδικός του χρήστη για πρόσθετη ασφάλεια.

## GetPost (GET)

<b>https://localhost:8443/api/rest/post/get/{postId}</b>	
<i>Request</i>	
postId	long
<i>Response</i>	
post	Article

### *Περιγραφή:*

Σε περίπτωση που σε ένα χρήστη έρθει ειδοποίηση για μια δημοσίευσή του, τότε μπορεί να δει την δημοσίευση αυτή μόνη της και να δει τις αντιδράσεις σε αυτήν. Η παραπάνω κλήση επιστρέφει για ένα μοναδικό αριθμό ειδοποίησης, τα στοιχεία της.

### Λοιπές επεξηγηματικές λεπτομέρειες:

Οι μοναδικοί αριθμοί των οντοτήτων της εφαρμογής βρίσκονται με αύξουσα σειρά στα παρακάτω όρια:

- Χρήστες: 1000000001 – 1999999999
- Δημοσιεύσεις: 2000000001 – 2999999999
- Αγγελίες: 3000000001 – 3999999999
- Ειδοποιήσεις: 4000000001 – 4999999999
- Επαγγελματική Εμπειρία: 5000000001 – 5999999999
- Εκπαιδευτική Σταδιοδρομία: 6000000001 – 6999999999
- Δεξιότητες: 7000000001 – 7999999999

Το συγκεκριμένο περιθώριο αφήνει ένα σχετικά μεγάλο παράθυρο στην εφαρμογή για την αρχική λειτουργία της. Προτείνεται σε μεταγενέστερο στάδιο, να χρησιμοποιηθεί μια δομή του τύπου GUID (Globally Unique Identifier) ώστε να αποφευχθεί το πρόβλημα του περιορισμένου χώρου.

### Δυνατότητες Εξέλιξης:



Η εφαρμογή έχει πολύ μεγάλα περιθώρια εξέλιξης. Υπάρχουν πολλοί τομείς οι οποίοι, αν και δεν υλοποιήθηκαν στη παρούσα φάση, θα μπορούσαν να εξελίξουν την εφαρμογή στο μέλλον.

Αρχικά, θα προστεθούν εικόνες ή άλλα πολυμέσα στις δημοσιεύσεις. Κάθε χρήστης θα έχει τη δυνατότητα να επισυνάπτει στη δημοσίευσή του εικόνες, βίντεο ή αρχεία ήχου που να αναπαριστούν ή να υποστηρίζουν τη θέση του. Ακόμα, υπάρχει η επιλογή της αποθήκευσης πάνω από μιας εικόνας ανά χρήστη, ώστε να παρουσιάζεται μια πιο ολοκληρωμένη εικόνα. Υπάρχουν ακόμα επιλογές, όπως η δυνατότητα διαγραφής σχολίων ή συνδέσεων που θα προσδώσουν μεγαλύτερη ευελιξία στην εφαρμογή και θα καθιστούν τη διαχείρισή της πιο εύκολη. Τέλος, μαζί με την οπτική βελτίωση, μεγάλο βάρος θα δοθεί και στην εξέλιξη των αλγορίθμων που επιλέγουν τις αγγελίες ή τις δημοσιεύσεις που εμφανίζονται σε κάθε χρήστη, με σκοπό τη καλύτερη δυνατή προσωπική εμπειρία.