My-LinkedIn Backend

Βάιος Λύτρας

Θεοδώρα Παντελιού

Πίνακας περιεχομένων

Πληροφορίες:

Η εργασία φτιάχτηκε σε angular για front-end και με spring boot για το back-end και μέσω Intellij IDEA 2021.1.3.

Σημεία που χρειάζονται αναφορά περί backend:

Γενικά:

Πληροφορίες για τις οντότητες της βάσης μπορείτε να δείτε μέσα από τα πεδία των κλάσεών τους. Γενικά όλες οι οντότητες έχουν ένα μοναδικό id (αριθμό) και οι οντότητες που θα πρέπει να συνδεθούν με άλλες οντότητες άμεσα (πχ τα comments ανήκουν σε post) έχουν γίνει με sql annotations του τύπου OneToMany, ManyToOne, OneToOne κλπ. Πιο συγκεκριμένα για κάθε οντότητα μπορείτε να δείτε μέσα στον κώδικα απλά οι περισσότερες σχέσεις μεταξύ των οντοτήτων που έχουμε δεν είναι bidirectional και οι λόγοι είναι ότι:
1) Δεν χρειαζόταν να είναι bidirectional καθώς κάνουμε την δουλειά μας μέσω του ενός
2) Το bidirectional σε περιορίζει σε κάποια πράγματα.

Security:

CORS policy.

Αιτήματα φιλίας:

Η οντότητα friends στο backend αποτελείται από τα id των δύο user και ένα String το οποίο είναι η κατάσταση της σύνδεσης των δύο (εδώ να σημειωθεί ότι ο user one είναι αυτός που έχει στείλει το αίτημα). Αν οι δύο έχουν συνδεθεί τότε η κατάσταση γίνεται “completed” και αν ο ένας έχει στείλει αίτημα και ο δεύτερος δεν το έχει αποδεχτεί τότε είναι “pending” η κατάσταση. Αν το αίτημα σύνδεσης απορριφθεί από τον δεύτερο τότε διαγράφεται και το αντικείμενο φιλίας στο backend.

Job implementation:

Για το GET των αγγελιών με βάση τις δεξιότητες του χρήστη, περνάει ως παράμετρος στην αίτηση η συμβολοσειρά με τις δεξιότητες του χρήστη. Η συμβολοσειρά γίνεται decode και έπειτα αφαιρούνται τα σημεία στίξης και τα πολλαπλά κενά. Έπειτα η συμβολοσειρά χωρίζεται σε λέξεις και για κάθε λέξη που δεν ανήκει στην λίστα των stopwords καλείται η συνάρτηση getJobBasedOnWord η του job repository η οποία βρίσκει τις αγγελίες οι οποίες περιέχουν την συγκεκριμένη λέξη. Η getJobBasedOnWord χρησιμοποιεί fuzzy query ώστε να βρει τις αγγελίες στις οποίες εμφανίζεται η λέξη με διαφορά έως και 2 γραμμάτων από την αρχική λέξη ώστε να πιάνει και περιπτώσεις όπου η λέξη είναι στον πληθυντικό κλπ.

Images, audio, videos:

Οι εικόνες και τα audio files λαμβάνονται από το get και αποθηκεύονται στην βάση ως bytes. Το όριο για τα ερωτήματα στην βάση δεν επιτρέπει μεγάλου μεγέθους αρχεία (>4mb).

Για τα βίντεο λόγω του ότι το μέγεθος τους είναι μεγαλύτερο από τις εικόνες και τα αρχεία ήχου η αποθήκευση τους στην βάση όπως τις εικόνες και τα αρχεία ήχου δεν δούλεψε λόγω του max_allowed_packet της βάσης. Έτσι ακολουθήσαμε άλλη προσέγγιση όπου λαμβάνονται και πάλι ως bytes από το get αλλά αποθηκεύονται ως αρχεία στο file system του υπολογιστή που τρέχει το backend, το οποίο διαβάσαμε ότι είναι καλύτερη πρακτική από την αποθήκευση στην βάση. Ο constructor του controller θα δημιουργήσει έναν φάκελο temp, αν δεν υπάρχει έτσι ώστε να αποθηκεύονται εκεί αυτά τα αρχεία. Έτσι στα βίντεο το όριο είναι το max-file-size που στο application.properties έχει οριστεί στα

100mb. Με τον τρόπο αυτό θα μπορούσαν να γίνουν και οι εικόνες και τα αρχεία ήχου αλλά το αφήσαμε όπως ήταν για να φανούν και οι δύο τρόποι.

*Υποστηρίζεται και το ανέβασμα αρχείων gif.

Matrix factorization:

Για τον αλγόριθμο παραγωγής συστάσεων υπάρχει στο backend ένα ξεχωριστό component. Ο αλγόριθμος βρίσκεται μέσα στην κλάση matrix_factorization που έχει δηλωθεί ως service για να υπάρχει εύκολη πρόσβαση στις συναρτήσεις της πάνω σε ένα αντικείμενο. Ο αλγόριθμος του matrix factorization είναι η συνάρτηση algorithm της κλάσης. Δέχεται τον πίνακα των δεδομένων, τους πίνακες V και F, το K και το learning rate h και επιστρέφει τον πίνακα συστάσεων και το σφάλμα. Ο αλγόριθμος αυτός καλείται από τις συναρτήσεις mf_posts και mf_jobs. Αυτές γεμίζουν τον πίνακα των δεδομένων κατάλληλα, αρχικοποιούν τυχαία τους πίνακες V και F και καλούν τον αλγόριθμο με κάποια διαφορετικά h ώστε να βρεθεί το h με το μικρότερο σφάλμα και να κρατηθεί ο αντίστοιχος πίνακας συστάσεων. Τα ratings του πίνακα δεδομένων των post βγαίνουν από τον τύπο number_of_likes+number_of_comments και στην περίπτωση που δεν υπάρχουν ούτε like ούτε comments από έναν χρήστη για κανένα post το rating βγαίνει από τον αριθμό των προβολών. Στις αγγελίες μετράει μόνο ο αριθμός των προβολών σύμφωνα με την εκφώνηση. Οι πίνακες V και F και στις δυο συναρτήσεις αρχικοποιούνται από 1 ως 5 επειδή φαίνονται αρκετά ρεαλιστικά όρια για τον αριθμό των like+comment και των views αν και στην πραγματικότητα δεν υπάρχει κάποιο όριο. Οι συναρτήσεις mf_posts και mf_jobs τρέχουν περιοδικά κάθε 20 λεπτά μέσω της LoadMatrixFactorization.run_mf. Ο χρόνος αυτός είναι ενδεικτικός. Έτσι υπάρχουν ανά πάσα στιγμή δεδομένα στους πίνακες συστάσεων. Όταν γίνει η αίτηση GET καλεί την συνάρτηση job_recommendations ή post_recommendations η οποία παίρνει το διάνυσμα του χρήστη από τον πίνακα συστάσεων και ταξινομεί τα post ή τα jobs με βάση το rating και επιστρέφει την λίστα με τα ids των post/jobs ταξινομημένα από το καλύτερο στο χειρότερο. Η παράμετρος size χρησημοποιείται ώστε να μην επιστρέφεται ολόκληρη η λίστα αλλα μόνο οι κορυφαίες προτάσεις, επειδή αυτές που βρίσκονται στο τέλος της λίστας δεν έχει νόημα να προτείνονται. Επειδή έχουμε περίπου 20 post στην βάση μας έχουμε ορίσει μικρό size (5).

*Στο backend terminal εκτυπώνονται οι τελικοί πίνακες με τα ratings των post και των job μαζί με το h (που επιλέχθηκε από τον αλγόριθμο) και το σφάλμα που είχαν.

*Για διευκόλυνση χρησιμοποιήθηκε το org.ejml.simple.SimpleMatrix για το ορισμό και τις πράξεις πινάκων.

Τεχνητό Dataset:

Όταν τρέχει το springboot φορτώνει αυτόματα μέσω συναρτήσεων δεδομένα που ανταποκρίνονται στην πραγματικότητα το οποίο αποτελείται από 15 users 20+ posts και jobs και αρκετά likes και comments που είναι αρκετά για να διαπιστωθεί η λειτουργικότητα του matrix factorization. Για λόγους ευκολίας όλοι οι χρήστες έχουν κωδικό: 1234 και public πληροφορίες, επίσης τα email (username) τους μπορείτε να τα δείτε από το αρχείο LoadDatabase μέσα στον φάκελο user_impl. Ο χρήστης bilbo@gmail.com είναι λίγο πιο προσεγμένος και έχει μέσα έτοιμα μηνύματα για να δείτε και την λειτουργικότητα του chat. Επίσης αρχικοποιείται και ένας user ο οποίος είναι admin και έχει username: admin@gmail.com και κωδικό: 1234.