

# Αναφορά 2ης Προγραμματιστικής Άσκησης

ΟΝΟΜΑ: Μαυρογιώργης Δημήτρης

ΑΜ: 2016030016

ΠΛΗ 518 - Υπηρεσίες στο Υπολογιστικό Νέφος και την Ομίχλη  
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

December 21, 2020

## Fiware Keyrock IDM

Για το login και register των χρηστών μέσω του Keyrock, χρησιμοποιούνται αυτούσιες οι σελίδες από την 1 φάση του προτζεκτ. Πιο συγκεκριμένα, για το login στέλνονται στον Keyrock το email και το password του χρήστη με grand type 'password', προκειμένου να πάρει ο χρήστης ένα token το οποίο αποθηκεύεται στο session. Επίσης, με ένα δεύτερο request στον Keyrock παίρνουμε τις πληροφορίες ενός χρήστη, όπως το user id του στον Keyrock και το role που του έχει ανατεθεί. Όσον αφορά το register, αρχικά με τα credentials του admin στέλνεται ένα request για τη δημιουργία ενός token. Στη συνέχεια, αυτό το token χρησιμοποιείται στα επόμενα δύο request, στα οποία πρώτα δημιουργούμε ένα χρήστη στον Keyrock και, έπειτα, με δεύτερο request τον κάνουμε authorize στην εφαρμογή και του δίνουμε το role που επέλεξε κατά το register.

## Fiware Orion Context Broker

Κατά τη δημιουργία μίας καινούριας ταινίας από έναν cinema owner γίνεται παράλληλα create ένα Movie entity με τα ίδια attributes που είχαμε και στην πρώτη φάση του προτζεκτ. Εκτός από τα συγκεκριμένα attributes, προστέθηκαν και άλλα attributes isPlaying και isComing δύο με data type boolean, στα οποία αποθηκεύουμε την πληροφορία α) αν μια ταινία πρόκειται να προβληθεί σε διάστημα μικρότερο ή ίσο των 10 ημερών σε κάποιο σινεμά και β) αν η ταινία που έκανε subscription παίζεται ακόμα σε κάποιο σινεμά. Επίσης, κατά τη διαγραφή μιας ταινίας, γίνεται και διαγραφή του συγκεκριμένου entity που αντιστοιχεί στην ταινία που διαγράφηκε.

Οι απλοί χρήστες όταν κάνουν εισαγωγή μιας ταινίας στα αγαπημένα τους, δημιουργείται και ένα subscription στην ταινία που προστέθηκε. Οι χρήστες θα λάβουν ένα notification από τον orion για τις εξής περιπτώσεις:

- α) Αν η έναρξη προβολής μια ταινίας πρόκειται να γίνει σε 10 ή λιγότερες μέρες ο χρήστης θα λάβει ένα notification το οποίο θα τον ειδοποιεί ότι η ταινία πρόκειται να προβληθεί σε λιγότερο από 10 μέρες.
- β) Αν η ταινία παίζεται ακόμα ο χρήστης θα λάβει μία ειδοποίηση για την ταινία είναι διαθέσιμη σε κάποιο σινεμά. Αν τώρα ο χρήστης κάνει subscribe σε μια ταινία που έχει περάσει η ημερομηνία προβολής, τότε θα λάβει σχετική ειδοποίηση ότι η ταινία δεν είναι διαθέσιμη σε κάποιο από τα διαθέσιμα σινεμά.

Για το feed των χρηστών μέσω του Orion, ολη η πληροφορία που στέλνει ο Orion για την αλλαγή ενός από τα attribute isPlaying και isComing, Αποθηκεύονται σε ένα collection Notifications, έτσι ώστε ο χρήστης να έχει και ιστορικό όλων των ειδοποιήσεων ακόμα και όταν είναι εκτός εφαρμογής. Επίσης, σε ένα άλλο collection Subscriptions, κρατάμε την πληροφορία ότι ένα συγκεκριμένο subscription id ανήκει σε έναν συγκεκριμένο χρήστη, έτσι ώστε στο feed να επιστρέφονται μόνο οι ειδοποιήσεις που αντιστοιχούν στο συγκεκριμένο χρήστη και όχι κάποιον άλλο.

## Fiware Wilma Pep Proxy

Στην εφαρμογή προστέθηκε και η υπηρεσία Wilma Pep Proxy, για να προστατεύει τον orion και το Rest API data storage από κακόβουλους χρήστες. Όλα τα request από την υπηρεσία web app αποστέλονται στο app logic και στη συνέχεια, αντί να χρησιμοποιήσουμε απευθείας τις ip και port του orion και του data storage, στέλνουμε τα request στη Wilma με header X-Auth-token με το magic key που δηλώθηκε κατά το docker compose.

## Data Storage - Rest API

Για την εξηπυρέτηση όλων των request προς τη MongoDB με τα δεδομένα των ταινιών, των σινμά και των αγαπημένων, δημιουργήθηκε ένα API το οποίο δέχεται requests με curl από την υπηρεσία app logic. Με βάση τις παραμέτρους (π.χ. user id, movie id κλπ) που αποστέλονται μαζί με το URL κατά το curl και ανάλογα με την παράμετρο request type, κάνουμε εισαγωγές, ενημέρωσεις, διαγραφές και αναζητήσεις στη βάση δεδομένων MongoDB, και επιστρέφουμε τα δεδομένα πίσω στο app logic και έπειτα στο web app, για να προβληθούν στο χρήστη.

## AJAX Requests

Παράλληλα, για οποιαδήποτε λειτουργία εισαγωγής, ενημέρωσης, διαγραφής και αναζήτησης, χρησιμοποιούμε Ajax jqueryies για την ανανέωση ενός συγκεκριμένου μέρους της σελίδας και όχι ολόκληρης όπως γίνεται στην περίπτωση του reload.

Για την αναζήτηση ταινιών στους χρήστες, έγιναν 2 διαφορετικές αναζητήσεις, καθώς η αναζήτηση με βάση κάποια ημερομηνία στη MongoDB πρέπει να γίνεται με κάποιο εύρος. Πιο συγκεκριμένα, έχουμε τις εξής αναζητήσεις:

- i) Αναζήτηση με βάση τον τίτλο, την κατηγορία και το όνομα ενός σινεμά. Εδώ ψάχνουμε στα αντίστοιχα fields ενός document αν υπάρχει match με βάση ένα regular expression, το οποίο το εισάγειο χρήστης στο search bar.
- ii) Αναζήτηση εύρους ημερομηνιών. Ο χρήστης επιλέγει 2 ημερομηνίες και επιστρέφουμε όλες τις ταινίες που έχουν ημερομηνία έναρξης και λήξης εντός αυτού του εύρους.

## Migration GCP

Για το migration σε GCP, δημιουργήθηκε ένα VM Instance με λειτουργικό Ubuntu και έγιναν κάποιες ρυθμίσεις όπως φαίνεται στις παρακάτω εικόνες. Στη συνέχεια, μόλις δημιουργήθηκε το instance, εγκαταστάθηκε το Docker και Docker-Compose σε αυτό. Έπειτα, μέσω git hub κατέβηκε ο κώδικας μέσα στο VM και με το docker-compose up σηκώθηκε η εφαρμογή.

Για τα ajax requests δημιουργήθηκε ένα firewall rule έτσι, ώστε να μπορούν να στέλνονται στο server στη συγκεκριμένη external ip, που μας δώθηκε κατά τη δημιουργία του instance, και tcp port 81 που είναι το port του service app logic. Επιπλέον, έχω δημιουργήσει ένα αρχείο global.js στο οποίο έχω μια μεταβλητή για την ip που στέλνονται τα requests. Όταν το τρεξέτε στο docker απλώς θα πρέπει να κάνετε uncomment την πρώτη μεταβλητή και να βάλετε σε comments τη μεταβλητή που έχω την external ip του GCP. Ομοίως, για να μπορέσει να εισέλθει ο admin της εφαρμογής στον Keyrock IDM δημιουργήθηκε ένα ακόμη firewall rule με tcp port 3005 που είναι το port του Keyrock.

The first screenshot shows the 'Create an instance' wizard in the Google Cloud Platform console. It displays the configuration for a new VM instance named 'clouddev'. The configuration includes the region 'Europe West 2 (Ireland)', the zone 'europe-west2-b', the machine type 'e2-standard-2', and the OS 'Ubuntu 20.04 LTS'. The network is set to 'cloud-dev-net'. The 'Boot disk' is a new 40 GB standard persistent disk. The 'Service account' is 'default'. The 'Create' button is visible at the bottom right.

The second screenshot shows the 'Create an instance' wizard with the 'Create' button highlighted, indicating the instance is being created.

The third screenshot shows the 'Firewall' rules in the Google Cloud Platform console. It lists several firewall rules, including 'default-allow-http', 'default-allow-https', 'default-allow-ssh', 'default-allow-icmp', 'default-allow-internal', 'default-allow-ftp', and 'default-allow-ftp-data'. The 'default-allow-ftp' rule is highlighted, showing its configuration: Name: default-allow-ftp, Type: allow, Targets: all VM instances, Protocol: tcp, Action: allow, Priority: 1000, Status: enabled, and Ports: 21. The 'default-allow-ftp-data' rule is also highlighted, showing its configuration: Name: default-allow-ftp-data, Type: allow, Targets: all VM instances, Protocol: tcp, Action: allow, Priority: 1000, Status: enabled, and Ports: 20.