

# Music Streaming Android App

## Distributed Systems 2019-2020 aueb class Project

### Team info:

- Ηλίας Παυλάκος, 3160143
- Παναγιώτης Λαμπρόπουλος, 3050085
- Δημήτριος Αλεξάκης, 3160003

### How to setup the project (Phase 1)

1. Κάνουμε import το project στο IntelliJ(ή στο IDE της επιλογής μας)
2. Πατάμε next σε όλα, **εάν δεν εντοπίσει** απο μονο του την βιβλιοθηκη mp3agic που περιεχεται, πατάμε δεξι κλικ στο mp3agic.jar και παταμε Add as library.

### How to run the project (Phase 1)

**Υπαρχει μικρό βίντεο demo με ήχο στον φάκελο 2 λεπτών. (5.5 MB)**

**Όλες οι main που θα τρέξουμε βρήσκονται στο package RUN**

1. Τρέχουμε πρώτα τους brokers με arguments την σειρά που φαίνεται στο αρχείο **SystemInfo.txt** αντικαθιστώντας στην διεύθυνση IP με αυτή του υπολογιστή μας.
2. Αλλάζουμε το πεδίο **pathToDataSet** στους **pyblisher1,2 και 3** με το path του φακέλου μας που περιέχει το **dataSet1**. Τρέχουμε τους publishers και περιμένουμε να εμφανίσουν το μήνυμα 'Awaiting requests...'  
(Εάν θελουμε αλλη συλλογή με μουσικη πρεπει να αλλάξουμε και τους καλλιτέχνες που ζητανα οι consumers)
3. Τρέχουμε τους consumers αλλάζοντας το που θα αποθηκεύσουν τα δεδομένα **σε ξεχωριστο φακελο στον καθενα**. Θα πρεπει να εχουμε διαφορετικο φακελο για κάθε consumer ωστε να αποθηκεύσουν τα chunks χωρις conflicts στα ονοματα για ευκολια στην εξεταση

*Για ευκολία έχει δημιουργηθει η μεθοδος `getSystemIP()` που βάζει μονη της το IP του PC μας στα πεδία που χρειάζεται, αν υπαρξει κάποιο πρόβλημα αντικαθησουμε στα πεδια που γινεται χρηση της (στο RUN package) με την IP μας.*

### Διαφορές του project από τις αρχικές προδιαγραφές της εκφώνησης

- Το αρχείο MSG.java χρησιμοποιείται για να δημιουργούμε αντικείμενα τύπου MSG τα οποία λειτουργούν σαν πρωτόκολο επικοινωνίας μεταξύ όλων των Nodes. Κάθε είδος επικοινωνίας μεταξύ broker-publisher και broker-consumer γίνεται αποκλειστικά με αντικείμενα τύπου MSG. Αυτό μας δίνει μια επιπλέον ελευθερία ώστε να υλοποιήσουμε επιπλέον λειτουργίες όπως μηνύματα

σφάλματος, μηνύματα ερωτήσεων(περιγράφονται παρακάτω), καθώς και την δυνατότητα ο consumer να μπορεί να ζητάει πληροφορίες για έναν καλλιτέχνη όπως τα metadata απο όλα τα τραγούδια ενός album. Στο Phase 2 αυτή η επιπλέον λειτουργικότητα θα μπορεί να χρησιμοποιείθι ώστε να στέλνουμε ακόμα και το artwork του album όπως γίνεται και στην εφαρμογή *Spotify*.

- Οι consumers αντί να κρατάνε λίστες με όλους τους καλλιτέχνες για τους οποίους οι brokers είναι υπεύθυνοι( Πράγμα που σε πραγματικές συνθήκες θα ήταν πολύ μεγάλος αριθμός δεδομένων λόγω των άπειρων καλλιτεχνών), όταν θέλουν να ζητήσουν ένα τραγούδι ή ένα album ρωτάνε τους consumer με ένα μήνυμα τύπου `Query(artistName)` και λαμβάνουν μια απάντηση τύπου `boolean`, εάν ο broker που ρώτησαν είναι υπεύθυνος επιστρέφει `true` και ο consumer συνεχίζει και ζητάει τα δεδομένα απο αυτόν, αλλιώς ρωτάει τον επόμενο broker.
- Τα interfaces έχουν αφαιρεθεί για να μειωθεί το clutter του project μιας και δεν είχαν ουσιαστική λειτουργικότητα. Έαν και εφόσον χρειάζονται στο phase 2 θα προστεθούν κατάλληλα. Επίσης, τα ονόματα των μεθόδων του skeleton έχουν αλλάξει για να φαναιρώνουν την πραγματική λειτουργικότητα κάθε μεθόδου. Τέλος, έχουν προστεθεί πολλές καινούργιες μέθοδοι που το skeleton δεν περιείχε.
- Έχουν προστεθεί κάποιες extra μέθοδοι στον Consumer, που δεν είναι απαραίτητες για την λειτουργικότητα του, προκειμένου να κάνουμε πιο εύκολη την δουλεία του εξεταστή. Βρίσκονται κάτω απο το comment `//Other methods`. Καθώς και η μέθοδος `getSystemIP()` που επιστρέφει το ip του του συστήματος αυτόματα για να μην χρειάζεται να το βάζουμε χειροκίνητα.