



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020



music worth sharing

Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Del.1.2

Version 1.0

Μαστραλέξη Χριστίνα Μαρία cmastral@ece.auth.gr
Μουρατίδης Αναστάσιος atmourat@ece.auth.gr
Σαουλίδης Νικόλαος saouling@ece.auth.gr
Χαρίτου Ελένη charitoue@ece.auth.gr

Ημερομηνία 09/06/2020



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers [✉] και του M. Smialek's.	0.1
Α. Συμεωνίδης	29/3/2014	Μικρή αναθεώρηση – τροποποίηση ενοτήτων	0.1.3
Χ. Ζολώτας	10/4/2020	Μεγάλη αναθεώρηση – αφαίρεση ενοτήτων	0.4
Χ. Ζολώτας	15/4/2020	Μεγάλη αναθεώρηση – προσθήκη ενότητας REST προδιαγραφών	0.5.3
Κ. Παναγιώτου	25/4/2020	Μεγάλη αναθεώρηση – προσθήκη ενότητας Nodered περιγραφής	0.5.7
Α. Συμεωνίδης	30/4/2020	Αναθεώρηση και τελική δομή προτύπου	0.6

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Κ. Παναγιώτου	*	klpanagi@issel.ee.auth.gr
Χ. Ζολώτας	*	christopherzolotas@issel.ee.auth.gr
Αναστάσιος Μουρατίδης	mob	atmourat@ece.auth.gr
Νικόλαος Σαουλίδης	mob	saouling@ece.auth.gr
Χριστίνα Μαρία Μαστραλέξη	mob	cmastral@ece.auth.gr
Ελένη Χαρίτου	mob	charitoue@ece.auth.gr

[✉] Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Original template is available at: <http://www.processimpact.com/>



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	3
Λίστα Σχημάτων	5
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	6
1.1 Bridge Pattern	6
1.2 Proxy Pattern.....	7
1.3 Observer Pattern	8
2. Αρχιτεκτονική Συστήματος	9
2.1 Αναγνώριση Πόρων (Resources) Συστήματος	9
2.2 Τεκμηρίωση REST διεπαφής	11
2.2.1 Πόρος UserEntity.....	11
2.2.1.1 Μοντέλο δεδομένων UserEntity	11
2.2.1.2 Μοντέλο δεδομένων UserLoginBody.....	11
2.2.1.3 Μοντέλο δεδομένων UserCreationBody.....	11
Μοντέλο δεδομένων ChangePasswordBody	11
2.2.1.4 Endpoints πόρου UserEntity	12
2.2.2 Πόρος ActivityEntity	16
2.2.2.1 Μοντέλο δεδομένων ActivityCreationBody	16
2.2.2.2 Μοντέλο δεδομένων ActivityEntity.....	16
2.2.2.3 Endpoints πόρου ActivityEntity	17
2.2.3 Πόρος PlaylistEntity.....	20
2.2.3.1 Μοντέλο δεδομένων PlaylistEntity	20
2.2.3.2 Μοντέλο δεδομένων PlaylistCreationBody.....	20
2.2.3.3 Μοντέλο δεδομένων PlaylistUpdateBody	20
2.2.3.4 Endpoints πόρου PlaylistEntity	21
2.2.4 Πόρος PlayqueueEntity	27
2.2.4.1 Μοντέλο δεδομένων PlayqueueEntity.....	27
2.2.4.2 Μοντέλο δεδομένων PlayqueueCreationBody	27
2.2.4.3 Endpoints πόρου PlayqueueEntity	28
2.2.5 Πόρος AlbumEntity.....	35
2.2.5.1 Μοντέλο δεδομένων AlbumEntity	35
2.2.5.2 Endpoints πόρου AlbumEntity	35
2.2.6 Πόρος SongEntity	38
2.2.6.1 Μοντέλο δεδομένων SongEntity.....	38
2.2.6.2 Endpoints πόρου SongEntity	38
2.2.7 Πόρος ProfileEntity.....	41
2.2.7.1 Μοντέλο δεδομένων ProfileEntity	41



2.2.7.2	Μοντέλο δεδομένων ProfileCreationBody	42
2.2.7.3	Μοντέλο δεδομένων ChangeProfileBody	42
2.2.7.4	Endpoints πόρου ProfileEntity	42
2.2.8	Πόρος StatisticsEntity	48
2.2.8.1	Μοντέλο Δεδομένων StatisticsEntity	48
2.2.8.2	Endpoint πόρου StatisticsEntity	48
2.2.9	Πόρος SearchEntity	49
2.2.9.1	Μοντέλο δεδομένων SerarchEntity	49
2.2.9.2	Endpoint πόρου SerachEntity	50
2.2.10	Πόρος PlayerEntity	51
2.2.10.1	Μοντέλο δεδομένων PlayerEntity	51
2.2.10.2	Endpoints πόρου PlayerEntity	51
2.2.11	Πόρος NewsfeedEntity	56
2.2.11.1	Μοντέλο δεδομένων NewsfeedEntity	56
2.2.11.2	Μοντέλο δεδομένων NewsfeedCreationBody	56
2.2.11.3	Endpoints πόρου NewsfeedEntity	56
3.	Υλοποίηση Συστήματος με Node-RED	58
3.1	Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed	58
3.1.1	Ροές πόρου UserEntity	58
3.1.2	Ροές πόρου ActivityEntity	58
3.1.3	Ροές πόρου PlaylistEntity	59
3.1.4	Ροές πόρου PlayqueueEntity	60
3.1.5	Ροές πόρου AlbumEntity	61
3.1.6	Ροές πόρου SongEntity	61
3.1.7	Ροές πόρου ProfileEntity	62
3.1.8	Ροές πόρου StatisticsEntity	63
3.1.9	Ροές πόρου SearchEntity	63
3.1.10	Ροές πόρου PlayerEntity	63
3.1.11	Ροές πόρου NewsfeedEntity	64
3.2	Υλοποίηση Ιστοριών χρήστη	64
3.2.1	View News Feed	64
3.2.2	Search Music or Profiles	65
3.2.3	View Profile	65
3.2.4	Follow/Unfollow User	66
3.2.5	Update Profile Info	67
3.2.6	Create Public Playlist	67
3.2.7	Make Playlist Private	67
3.2.8	Edit Playlist	67



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

3.2.9 Delete Playlist	68
3.2.10 Control Music Player	68
3.2.11 Manage PlayQueue	70
3.2.12 Upload Music.....	71
3.2.13 View Audience Stats	72

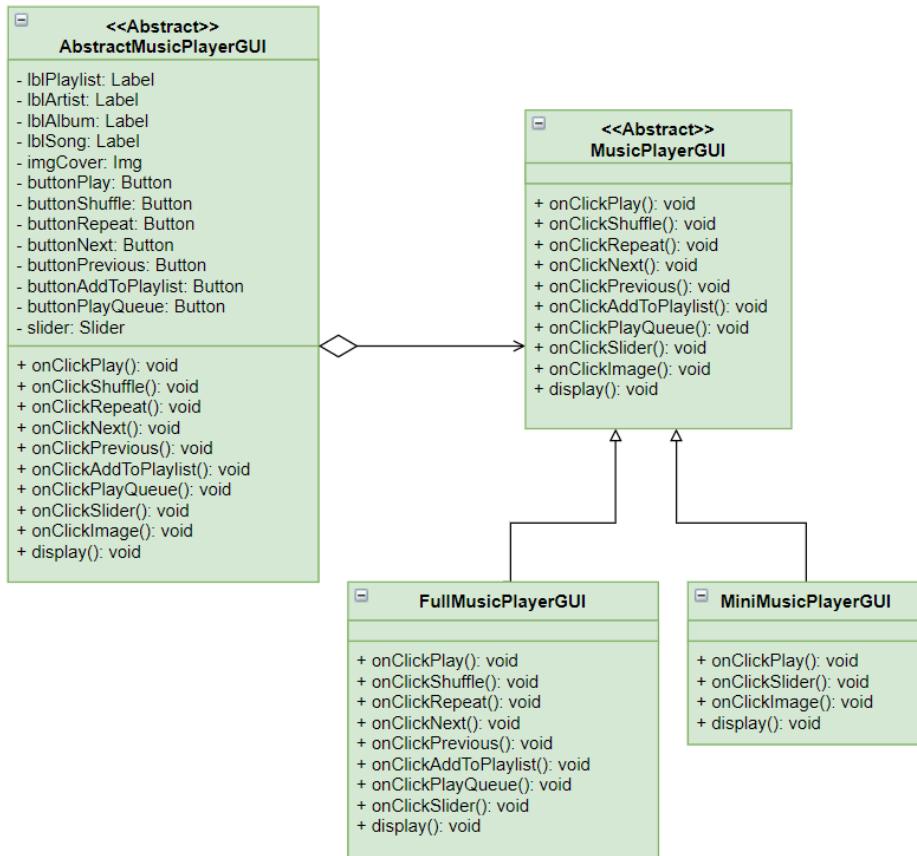
Λίστα Σχημάτων

Εικόνα 1: Bridge Pattern	6
Εικόνα 2: Proxy Pattern.....	7
Εικόνα 3: Observer Pattern.....	8



1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

1.1 Bridge Pattern



Εικόνα 1: Bridge Pattern

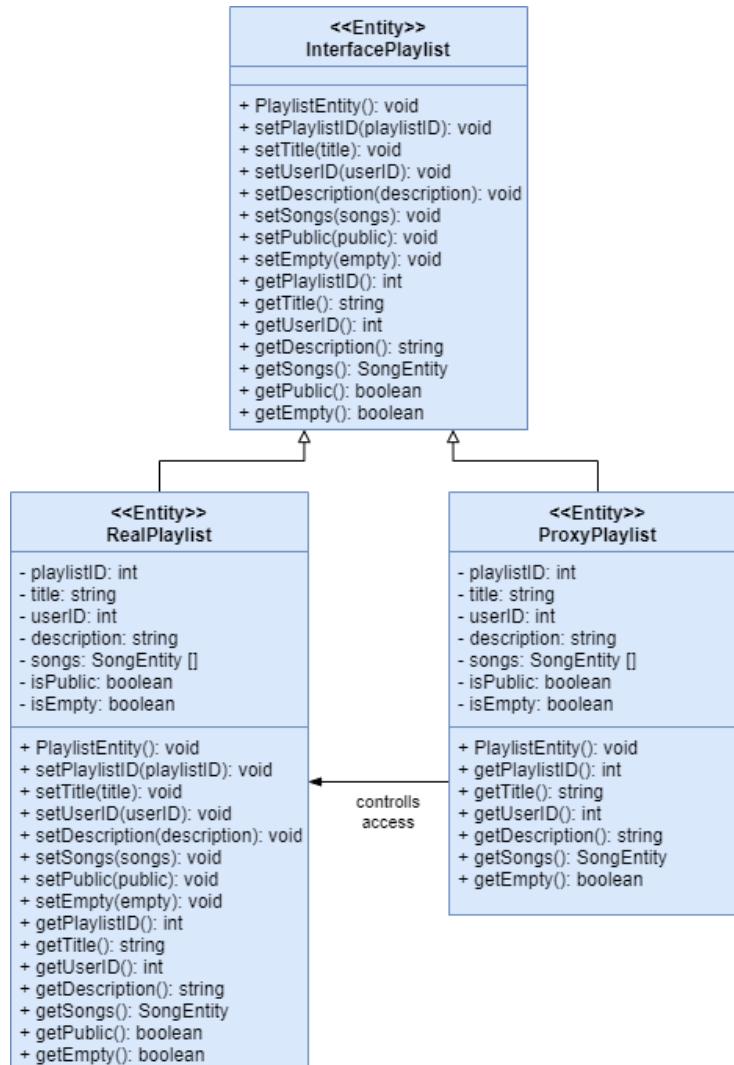
Το **Bridge Pattern** ανήκει στην κατηγορία των **δομικών** προτύπων σχεδιασμού. Χρησιμοποιήθηκε για την παροχή δύο διαφορετικών υλοποιήσεων της ίδιας διεπαφής, του **MusicPlayerGUI** Interface. Συγκεκριμένα, η υλοποίηση του **MiniMusicPlayerGUI** εμφανίζεται πάντα ως μέρος της διεπαφής χρήστη, ενώ το **FullMusicPlayerGUI** εμφανίζεται όταν κάποιος πατήσει στο **MiniMusicPlayerGUI** καταλαμβάνοντας ολόκληρη την οθόνη. Το **FullMusicPlayerGUI** παρέχει πρόσθετη λειτουργικότητα σε σχέση με το **MiniMusicPlayer** (shuffle, next κτλ), καθώς και μια διαφορετική υλοποίηση του slider (γραμμικό και κυκλικό αντίστοιχα).

Το συγκεκριμένο πρότυπο σχεδίασης σχετίζεται με τη:

<ΛΑ-11> Ο χρήστης πρέπει να μπορεί να επιλέγει αναπαραγωγή / παύση / επόμενο / προηγούμενο και να μετακινεί το slider στο σημείο αναπαραγωγής που επιθυμεί.



1.2 Proxy Pattern



Εικόνα 2: Proxy Pattern

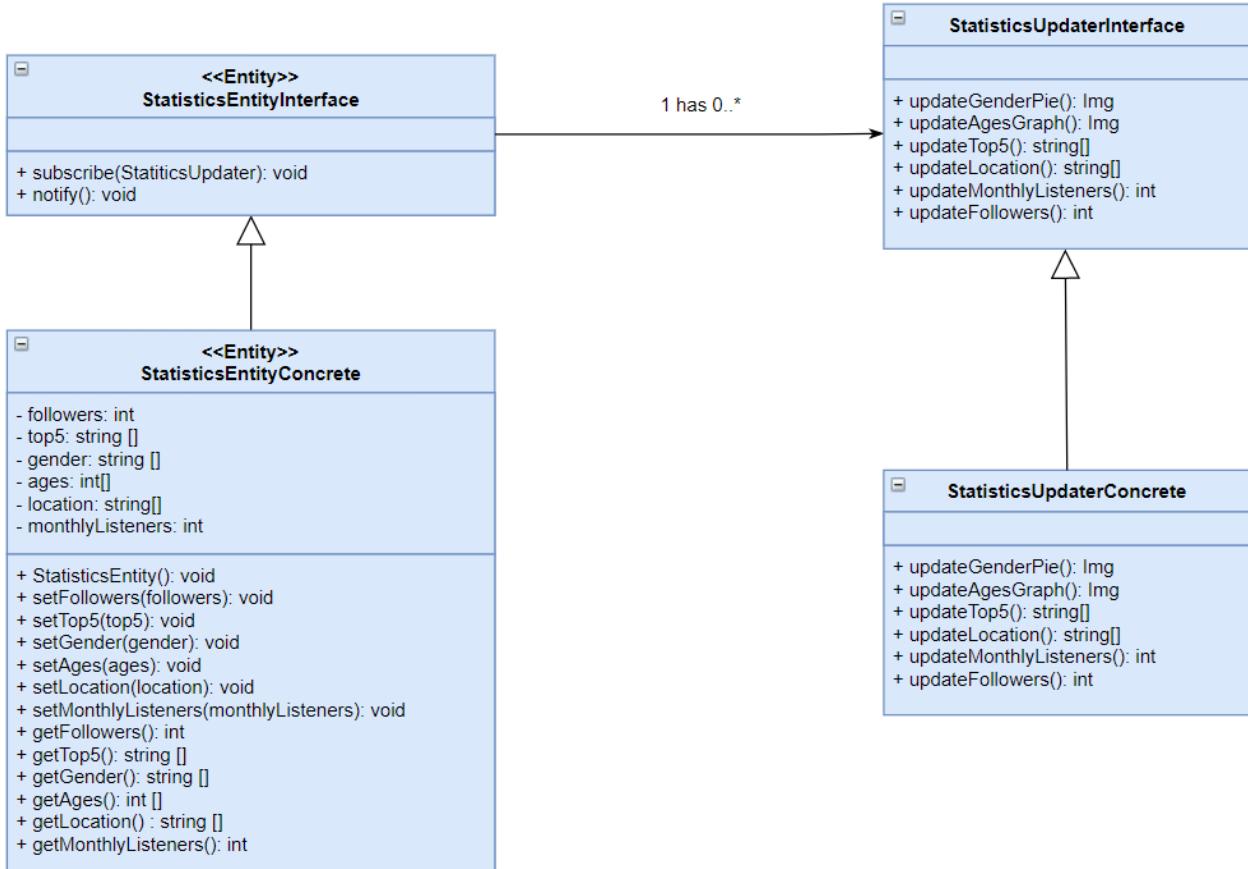
Το **Proxy Pattern** ανήκει στην κατηγορία των **δομικών** προτύπων. Όλοι οι χρήστες του μονάχα έχουν δικαιώματα προβολής των περιεχομένων των δημόσιων Playlists που ανήκουν σε άλλους χρήστες (τίτλος, περιγραφή, τραγούδια κτλ). Ωστόσο δεν έχουν δικαιώματα τροποποίησης του περιεχομένου και των ιδιοτήτων μιας Playlist που δεν τους ανήκει. Έτσι δημιουργήθηκε η ανάγκη χρήσης μιας Proxy Playlist, ώστε μόνο οι δημιουργοί τους να έχουν δυνατότητα τροποποίησης των στοιχείων τους (set methods).

Το συγκεκριμένο πρότυπο σχεδίασης ικανοποιεί μια πρόσθετη ΜΛΑ που ορίζεται ως εξής:

Το σύστημα πρέπει να επιτρέπει προσπάθειες επεξεργασίας του περιεχομένου μιας Playlist μόνο σε χρήστες στους οποίους ανήκει.



1.3 Observer Pattern



Εικόνα 3: Observer Pattern

Το **Observer Pattern** ανήκει στην κατηγορία των προτύπων **συμπεριφοράς**. Στην προκειμένη περίπτωση το **StatisticsEntity** είναι το **Subject** (Publisher), καθώς είναι η οντότητα που μεταβάλλεται συχνά, ανάλογα με την αλληλεπίδραση των ακροατών με τη μουσική του εκάστοτε Καλλιτέχνη. Αντίστοιχα, οι **Observers** είναι οι κλάσεις **StatisticsUpdater** που οπτικοποιούν τα στατιστικά του Καλλιτέχνη. Κάθε φορά που αλλάζει μια μεταβλητή στην οντότητα (με την κλήση του αντίστοιχου setter) καλείται η κατάλληλη συνάρτηση **update** των **observers** που έχουν **subscribe** στην συγκεκριμένη οντότητα.

Το παραπάνω πρότυπο σχεδίασης καλύπτει τη λειτουργική απαίτηση που περιεγράφηκε στο προηγούμενο παραδοτέο:

<ΛΑ-10> Ο Καλλιτέχνης θα πρέπει να μπορεί να βλέπει τα στατιστικά για το ακροατήριό του.



2. Αρχιτεκτονική Συστήματος

Το specification του API, ο server της εφαρμογής και οι ροές του Node-RED σε μορφή JSON μπορούν να βρεθούν [εδώ](#).

2.1 Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
UserEntity	/user	POST
	/user/login	PUT
	/user/{username}	PUT
	/user/{username}	GET
ActivityEntity	/activity	POST
	/activity	GET
	/activity/{activityID}	GET
PlaylistEntity	/playlist	POST
	/playlist/{playlistID}	GET
	/playlist/{playlistID}	DELETE
	/playlist/{playlistID}	PUT
	/playlist/{playlistID}/song/{songID}	DELETE
	/playlist/{playlistID}/song/{songID}	PUT
PlayqueueEntity	/playqueue	POST
	/playqueue	GET
	/playqueue	PUT
	/playqueue/song/{songID}	PUT
	/playqueue/song/{songID}	DELETE
	/playqueue/album/{albumID}	PUT
	/playqueue/playlist/{playlistID}	PUT
AlbumEntity	/album	POST
	/album/{albumID}	GET
	/album/{albumID}	DELETE
SongEntity	/song	POST
	/song/{songID}	GET
	song/{songID}	DELETE
ProfileEntity	/profile	POST
	/profile/{userID}	PUT
	/profile/{userID}	GET



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

	/profile/follow/{followingID}	PUT
	/profile/unfollow/{followingID}	DELETE
	/profile/followedby/{followersID}	PUT
	/profile/unfollowedby/{followersID}	DELETE
StatisticsEntity	/statistics	GET
SerachEntity	/search	GET
PlayerEntity	/player	GET
	/player/play	PUT
	/player/pause	PUT
	/player/next	PUT
	/player/previous	PUT
	/player/seek	PUT
NewsfeedEntity	/newsfeed	POST
	/newsfeed	GET



2.2 Τεκμηρίωση REST διεπαφής

2.2.1 Πόρος UserEntity

2.2.1.1 Μοντέλο δεδομένων UserEntity

```
UserEntity ▼ {  
    username      string  
                example: niksaoul  
    password      string  
                example: 123456789abcd!  
    isArtist      boolean  
                example: true  
    userID        integer  
                example: 1312  
    email         string  
                example: niksaoul@mob.com  
}
```

2.2.1.2 Μοντέλο δεδομένων UserLoginBody

```
UserLoginBody ▼ {  
    username*    string  
                example: niksaoul  
    password*    string  
                example: 123456789abcd!  
}
```

2.2.1.3 Μοντέλο δεδομένων UserCreationBody

```
UserCreationBody ▼ {  
    username*    string  
                example: niksaoul  
    password*    string  
                example: 123456789abcd!  
    isArtist*    boolean  
                example: true  
    email*       string  
                example: niksaoul@mob.com  
}
```

Μοντέλο δεδομένων ChangePasswordBody

```
ChangePasswordBody ▼ {  
    newPassword* string  
                example: 987654321dcba?  
    token*       string  
                example: laskfjjsdnsfus5674  
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.1.4 Endpoints πόρου UserEntity

User Operations related to user resource



POST

/user Create user



Create user by providing: username,password,email and choose Listener or Artist.

Parameters

Try it out

Name

Description

userCreationBody

Elements required to sign up

object
(body)

Example Value | Model

```
{  
    "username": "niksaoul",  
    "password": "123456789abcd!",  
    "isArtist": true,  
    "email": "niksaoul@mob.com"  
}
```

Parameter content type

application/json



Responses

Response content type

application/json



Code

Description

201

Created.

Example Value | Model

```
{  
    "username": "niksaoul",  
    "password": "123456789abcd!",  
    "isArtist": true,  
    "userID": 1312,  
    "email": "niksaoul@mob.com"  
}
```

400

Bad Request. Invalid input.

409

Conflict. A user with the same username or email already exists.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

PUT /user/login Login

User login by providing username and password.

Parameters

Name **Description**

userLogin object (body)	Login with username and password
--------------------------------------	----------------------------------

Example Value | Model

```
{  
    "username": "niksaoul",  
    "password": "123456789abcd!"  
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code **Description**

200	OK.
-----	-----

Example Value | Model

```
{  
    "token": "string"  
}
```

Headers:

Name	Description	Type
X-Rate-Limit	calls per hour allowed by the user	integer

400

Bad Request. Wrong username-password combination.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

PUT /user/{username} Change Password

Change user password by providing a new valid user password.

Parameters

Name	Description
username * required <small>string (path)</small>	username <input type="text" value="username - username"/>

userChangePassword Login by providing the new password and login token
object (body)

```
{  
  "newpassword": "987654321dcba?",  
  "token": "laskfjjsdnsfus5674"  
}
```

Parameter content type

Responses

Code	Description
200	OK. <pre>{ "username": "niksaoul", "password": "123456789abcd!", "isArtist": true, "userID": 1312, "email": "niksaoul@mob.com" }</pre>
400	Bad Request. Invalid input.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

GET /user/{username} Get user info

Get user information: username, password, isArtist, userID, email

Parameters

Name	Description
username * required <small>string (path)</small>	username <input type="text" value="username - username"/>

Responses

Code	Description
200	OK. <pre>{ "username": "niksaoul", "password": "123456789abcd!", "isArtist": true, "userID": 1312, "email": "niksaoul@mob.com" }</pre>
404	Not Found.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.2 Πόρος ActivityEntity

2.2.2.1 Μοντέλο δεδομένων ActivityCreationBody

```
ActivityCreationBody ▼ {  
    userID*           integer  
                      example: 9999  
    typeOfActivity*  string  
                      example: uploaded an album  
    date*            string($date)  
                      example: 2020-04-12T23:20:50.52Z  
}
```

2.2.2.2 Μοντέλο δεδομένων ActivityEntity

```
ActivityEntity ▼ {  
    userID          integer  
                  example: 9999  
    activityID     integer  
                  example: 5555  
    typeOfActivity string  
                  example: uploaded an album  
    date           string($date)  
                  example: 2020-04-12T23:20:50.52Z  
}
```



2.2.2.3 Endpoints πόρου ActivityEntity

activity Operations related to activity resource ▼

POST /activity Create activity

Create activity of a user. It contains the userID, the type of the activity and a date.

Parameters Try it out

Name	Description
activityCreationBody	Elements required to create activity
object (body)	Example Value Model
{ "userID": 9999, "typeOfActivity": "uploaded an album", "date": "2020-04-12T23:20:50.52Z" }	

Parameter content type application/json

Responses application/json

Code	Description
201	Created. Example Value Model
{ "userID": 9999, "activityID": 5555, "typeOfActivity": "uploaded an album", "date": "2020-04-12T23:20:50.52Z" }	



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

GET /activity Get activities of a user

Get activities of a specific user by date.

Parameters

Try it out

Name	Description
userID * required	userID
date	date

Responses

Response content type application/json

Code	Description
200	OK.
404	Not Found.

Example Value | Model

```
[  
  {  
    "userID": 9999,  
    "activityID": 5555,  
    "typeOfActivity": "uploaded an album",  
    "date": "2020-04-12T23:20:50.52Z"  
  }  
]
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

GET /activity/{activityID} Get activity info

Get activity info by providing an activityID.

Parameters

Try it out

Name	Description
activityID * required	activity
integer	
(path)	activityID - activity

Responses

Response content type application/json

Code	Description
200	OK. Example Value Model <pre>{ "userID": 9999, "activityID": 5555, "typeOfActivity": "uploaded an album", "date": "2020-04-12T23:20:50.52Z" }</pre>
404	Not Found.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.3 Πόρος PlaylistEntity

2.2.3.1 Μοντέλο δεδομένων PlaylistEntity

```
PlaylistEntity ▼ {  
    playlistID      integer  
    example: 123456  
    title           string  
    example: A playlist that wants to be adored  
    userID          integer  
    example: 123  
    coverURI        string  
    example: https://mob/media/File:Adored_playlist_cover.jpg  
    description      string  
    example: Sometimes I fantasise when the streets are cold and lonely  
    songs            ▼ [SongEntity ▼ {  
        title           string  
        example: Come together  
        artist          string  
        example: The Beatles  
        album           string  
        example: Abbey Road  
        coverURI        string  
        example: https://mob/media/File:AbbeyRoad_song_cover.jpg  
        duration         integer  
        example: 259  
        songID          integer  
        example: 123456789  
        songURI         string  
    }]  
    isPublic         boolean  
    example: true  
    isEmpty          boolean  
    example: false  
}
```

2.2.3.2 Μοντέλο δεδομένων PlaylistCreationBody

```
PlaylistCreationBody ▼ {  
    title*           string  
    example: A playlist that wants to be adored  
    description*     string  
    example: Sometimes I fantasise when the streets are cold and lonely  
    songs            ▶ [...]  
    isPublic         boolean  
    example: true  
    coverURI        string  
    example: https://mob/media/File:Adored_playlist_cover.jpg  
}
```

2.2.3.3 Μοντέλο δεδομένων PlaylistUpdateBody

```
PlaylistUpdateBody ▼ {  
    title           string  
    example: A playlist that wants to be adored  
    description      string  
    example: Sometimes I fantasise when the streets are cold and lonely  
    isPublic         boolean  
    example: true  
    coverURI        string  
    example: https://mob/media/File:Adored_playlist_cover.jpg  
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.3.4 Endpoints πόρου PlaylistEntity

playlist Operations related to playlist resource

▼

POST /playlist Create playlist

Create a playlist by providing a title, a description, an array of songs, if it's public, and a coverURI.

Parameters

Try it out

Name	Description
PlaylistCreationBody object (body)	Elements required to create a playlist Example Value Model <pre>{ "title": "A playlist that wants to be adored", "description": "Sometimes I fantasise when the streets are cold and lonely", "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }], "isPublic": true, "coverURI": "https://mob/media/File:Adored_playlist_cover.jpg" }</pre>

Parameter content type
application/json

Responses

Response content type application/json

Code	Description
201	Created. Example Value Model <pre>{ "playlistID": 123456, "title": "A playlist that wants to be adored", "userID": 123, "coverURI": "https://mob/media/File:Adored_playlist_cover.jpg", "description": "Sometimes I fantasise when the streets are cold and lonely", "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }], "isPublic": true, "isEmpty": false }</pre>

400 Bad Request. Invalid input.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

GET /playlist/{playlistID} Get a specific playlist

Get the data of a playlist entity by providing its unique playlistID

Parameters

Try it out

Name	Description
playlistID * required <small>integer (path)</small>	The unique id of a playlist playlistID - The unique id of a playlist

Responses

Response content type application/json

Code	Description
200	OK. Example Value Model

```
{  
    "playlistID": 123456,  
    "title": "A playlist that wants to be adored",  
    "userID": 123,  
    "coverURI": "https://mob/media/File:Adored_playlist_cover.jpg",  
    "description": "Sometimes I fantasise when the streets are cold and lonely",  
    "songs": [  
        {  
            "title": "Come together",  
            "artist": "The Beatles",  
            "album": "Abbey Road",  
            "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
            "duration": 259,  
            "songID": 123456789,  
            "songURI": "string"  
        }  
    ],  
    "isPublic": true,  
    "isEmpty": false  
}
```

404 Not Found. The playlistID provided doesn't match any playlist in the database.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

DELETE /playlist/{playlistID} Delete a specific playlist

Delete the data of a playlist entity by providing its unique playlistID.

Parameters

Try it out

Name	Description
playlistID * required	The unique id of a playlist
string	
(path)	playlistID - The unique id of a playlist

Responses

Response content type application/json

Code	Description
200	OK.
404	Not Found. The playlistID provided doesn't match any song in the database.

PUT /playlist/{playlistID} Update info of a playlist

Update a playlist entity by providing its unique playlistID and the data you wish to change.

Parameters

Try it out

Name	Description
playlistID * required	The unique id of a playlist
integer	
(path)	playlistID - The unique id of a playlist

PlaylistCreationBody

object

(body)

Elements required to create a playlist

Example Value | Model

```
{  
    "title": "A playlist that wants to be adored",  
    "description": "Sometimes I fantasise when the streets  
are cold and lonely",  
    "isPublic": true,  
    "coverURI":  
    "https://mob/media/File:Adored_playlist_cover.jpg"  
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Parameter content type
application/json

Responses Response content type application/json

Code	Description
200	OK. Example Value Model <pre>{ "playlistID": 123456, "title": "A playlist that wants to be adored", "userID": 123, "coverURI": "https://mob/media/File:Adored_playlist_cover.jpg", "description": "Sometimes I fantasise when the streets are cold and lonely", "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }], "isPublic": true, "isEmpty": false }</pre>
404	Not Found. The playlistID provided doesn't match any playlist in the database.

DELETE /playlist/{playlistID}/song/{songID} Remove a specific song from a playlist

Remove a song from a playlist by providing its unique songID and the corresponding playlistID.

Parameters Try it out

Name	Description
playlistID * required string (path)	The unique id of a playlist playlistID - The unique id of a playlist



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

songID * required
string
(path)

The unique id of a song
songID - The unique id of a song

Responses Response content type application/json ▾

Code	Description
200	OK. Example Value Model
	<pre>{ "playlistID": 123456, "title": "A playlist that wants to be adored", "userID": 123, "coverURI": "https://mob/media/File:Adored_playlist_cover.jpg", "description": "Sometimes I fantasise when the streets are cold and lonely", "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }], "isPublic": true, "isEmpty": false }</pre>
404	Not Found. The songID provided doesn't match any song in the playlist.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

PUT /playlist/{playlistID}/song/{songID}

Add a song to a playlist

Parameters

Try it out

Name	Description
playlistID * required	The unique id of a playlist
string	(path)
playlistID - The unique id of a playlist	
songID * required	The unique id of a song
string	(path)
songID - The unique id of a song	

Responses

Response content type application/json

Code	Description
200	OK. Song added to playlist.

Example Value | Model

```
{ "playlistID": 123456, "title": "A playlist that wants to be adored", "userID": 123, "coverURI": "https://mob/media/File:Adored_playlist_cover.jpg", "description": "Sometimes I fantasise when the streets are cold and lonely", "songs": [ { "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" } ], "isPublic": true, "isEmpty": false }
```

404

Not Found. The songID provided doesn't match any song in the database or the playlist doesn't exist.



2.2.4 Πόρος PlayqueueEntity

2.2.4.1 Μοντέλο δεδομένων PlayqueueEntity

```
PlayqueueEntity ▼ {  
    isEmpty  
        boolean  
        example: false  
    isShuffle  
        boolean  
        example: true  
    isRepeat  
        boolean  
        example: true  
    songs  
        ▼ [SongEntity ▼ {  
            title  
                string  
                example: Come together  
            artist  
                string  
                example: The Beatles  
            album  
                string  
                example: Abbey Road  
            coverURI  
                string  
                example: https://mob/media/File:AbbeyRoad_song_cover.jpg  
            duration  
                integer  
                example: 259  
            songID  
                integer  
                example: 123456789  
            songURI  
                string  
        }]  
    }  
}
```

2.2.4.2 Μοντέλο δεδομένων PlayqueueCreationBody

```
PlayqueueCreationBody ▼ {  
    songs*  
        ▼ [SongEntity ▼ {  
            title  
                string  
                example: Come together  
            artist  
                string  
                example: The Beatles  
            album  
                string  
                example: Abbey Road  
            coverURI  
                string  
                example: https://mob/media/File:AbbeyRoad_song_cover.jpg  
            duration  
                integer  
                example: 259  
            songID  
                integer  
                example: 123456789  
            songURI  
                string  
        }]  
    }  
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.4.3 Endpoints πόρου PlayqueueEntity

playqueue Operations related to playqueue resource

POST /playqueue Initialize Playqueue.

Create Playqueue and add the provided songs.

Parameters

Name Description

userPlayqueue Songs in a playqueue
object Example Value | Model
(body)

```
{
  "songs": [
    {
      "title": "Come together",
      "artist": "The Beatles",
      "album": "Abbey Road",
      "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",
      "duration": 259,
      "songID": 123456789,
      "songURI": "string"
    }
  ]
}
```

Parameter content type
application/json

Responses Response content type application/json

Code Description

201 Created.
Example Value | Model

```
{
  "isEmpty": false,
  "isShuffle": true,
  "isRepeat": true,
  "songs": [
    {
      "title": "Come together",
      "artist": "The Beatles",
      "album": "Abbey Road",
      "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",
      "duration": 259,
      "songID": 123456789,
      "songURI": "string"
    }
  ]
}
```

400 Bad Request. Empty playqueue.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

GET /playqueue Load playqueue

Load playqueue of a user, containing the songs and playback info.

Parameters

No parameters

Try it out

Responses

Response content type application/json

Code	Description
200	OK. Example Value Model

```
{  
    "isEmpty": false,  
    "isShuffle": true,  
    "isRepeat": true,  
    "songs": [  
        {  
            "title": "Come together",  
            "artist": "The Beatles",  
            "album": "Abbey Road",  
            "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
            "duration": 259,  
            "songID": 123456789,  
            "songURI": "string"  
        }  
    ]  
}
```

Code	Description
404	Not Found. Empty Playqueue.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

PUT /playqueue Edit playback settings

Edit Playback settings.

Parameters

Try it out

Name	Description
playbackSettings * required	Update playback settings
object (body)	Example Value Model

```
{  
    "isShuffle": true,  
    "isRepeat": true  
}
```

Parameter content type

application/json

Responses

Response content type

application/json

Code	Description
200	OK.

Example Value | Model

```
{  
    "isEmpty": false,  
    "isShuffle": true,  
    "isRepeat": true,  
    "songs": [  
        {  
            "title": "Come together",  
            "artist": "The Beatles",  
            "album": "Abbey Road",  
            "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
            "duration": 259,  
            "songID": 123456789,  
            "songURI": "string"  
        }  
    ]  
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

PUT

/playqueue/song/{songID} Add a song to a playqueue



Add a song to the playqueue by providing the corresponding songID.

Parameters

Try it out

Name	Description
------	-------------

songID * required	
--------------------------	--

integer

(path)

The unique id of a song

songID - The unique id of a song

Responses

Response content type

application/json



Code	Description
------	-------------

200	OK.
-----	-----

Example Value | Model

```
{
  "isEmpty": false,
  "isShuffle": true,
  "isRepeat": true,
  "songs": [
    {
      "title": "Come together",
      "artist": "The Beatles",
      "album": "Abbey Road",
      "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",
      "duration": 259,
      "songID": 123456789,
      "songURI": "string"
    }
  ]
}
```

404	
-----	--

Not Found. The songID provided doesn't match any song in the database or the playlist doesn't exist.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

DELETE

/playqueue/song

/{songID}

Remove a specific song from a
Playqueue



Remove a song from a Playqueue by providing its unique songID and the corresponding playlistID.

Parameters

Try it out

Name

Description

songID * required

integer

(path)

The unique id of a song

songID - The unique id of a song

Responses

Response content type

application/json



Code

Description

200

OK.

Example Value | Model

```
{
  "isEmpty": false,
  "isShuffle": true,
  "isRepeat": true,
  "songs": [
    {
      "title": "Come together",
      "artist": "The Beatles",
      "album": "Abbey Road",
      "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",
      "duration": 259,
      "songID": 123456789,
      "songURI": "string"
    }
  ]
}
```

404

Not Found. The songID provided doesn't match any song in the Playqueue.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

PUT

/playqueue/album/{albumID} Add album to a playqueue



Add album to the playqueue by providing the corresponding albumID.

Parameters

Try it out

Name

Description

albumID * required

The unique id of an album

integer

(path)

albumID - The unique id of an album

Responses

Response content type

application/json



Code

Description

200

OK.

Example Value | Model

```
{  
    "isEmpty": false,  
    "isShuffle": true,  
    "isRepeat": true,  
    "songs": [  
        {  
            "title": "Come together",  
            "artist": "The Beatles",  
            "album": "Abbey Road",  
            "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
            "duration": 259,  
            "songID": 123456789,  
            "songURI": "string"  
        }  
    ]  
}
```

404

Not Found.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

PUT

/playqueue/playlist/{playlistID} Add playlist to a playqueue



Add playlist to the playqueue by providing the corresponding playlistID.

Parameters

Try it out

Name	Description
------	-------------

playlistID * required	The unique id of an playlist
-----------------------	------------------------------

integer

(path)

playlistID - The unique id of an playlist

Responses

Response content type

application/json



Code	Description
------	-------------

200	OK.
-----	-----

Example Value | Model

```
{  
    "isEmpty": false,  
    "isShuffle": true,  
    "isRepeat": true,  
    "songs": [  
        {  
            "title": "Come together",  
            "artist": "The Beatles",  
            "album": "Abbey Road",  
            "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
            "duration": 259,  
            "songID": 123456789,  
            "songURI": "string"  
        }  
    ]  
}
```

404	Not Found.
-----	------------



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.5 Πόρος AlbumEntity

2.2.5.1 Μοντέλο δεδομένων AlbumEntity

```
AlbumEntity ▼ {  
    albumID      integer  
    example: 10  
    title        string  
    example: Cleopatra  
    artist       string  
    example: The Lumineers  
    coverURI    string  
    example: https://mob/media/File:Cleopatra_album_cover.jpg  
    songs        ▼ [SongEntity ▼ {  
        title        string  
        example: Come together  
        artist       string  
        example: The Beatles  
        album        string  
        example: Abbey Road  
        coverURI    string  
        example: https://mob/media/File:AbbeyRoad_song_cover.jpg  
        duration    integer  
        example: 259  
        songID      integer  
        example: 123456789  
        songURI    string  
    }]  
}
```

2.2.5.2 Endpoints πόρου AlbumEntity

album Operations related to album resource ▼

POST /album Create album  

Create album by uploading a .zip file.

Parameters 

Name	Description
Album * required file (formData)	Songs of album(.zip) Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Responses Response content type application/json

Code	Description
201	Created. Example Value Model <pre>{ "albumID": 10, "title": "Cleopatra", "artist": "The Lumineers", "coverURI": "https://mob/media/File:Cleopatra_album_cover.jpg", "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }] }</pre>
400	Bad Request. Invalid input.

GET /album/{albumID} Get album info

Get album info by providing its albumID.

Parameters Try it out

Name	Description
albumID * required integer (path)	Unique ID of the Album albumID - Unique ID of the Album



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

Responses

Response content type

application/json



Code

Description

200

OK.

Example Value | Model

```
{  
    "albumID": 10,  
    "title": "Cleopatra",  
    "artist": "The Lumineers",  
    "coverURI": "https://mob/media/File:Cleopatra_album_cover.jpg",  
    "songs": [  
        {  
            "title": "Come together",  
            "artist": "The Beatles",  
            "album": "Abbey Road",  
            "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
            "duration": 259,  
            "songID": 123456789,  
            "songURI": "string"  
        }  
    ]  
}
```

404

DELETE

/album/{albumID} Delete an album



Delete the data of an album entity by providing its unique albumID.

Parameters

Try it out

Name

Description

albumID * required

The unique id of an album

integer

(path)

albumID - The unique id of an album

Responses

Response content type

application/json



Code

Description

200

OK.

404

Not Found. The albumID provided doesn't match any album in the database.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

2.2.6 Πόρος SongEntity

2.2.6.1 Μοντέλο δεδομένων SongEntity

```
SongEntity {  
    title           string  
    example: Come together  
    artist          string  
    example: The Beatles  
    album           string  
    example: Abbey Road  
    coverURI       string  
    example: https://mob/media/File:AbbeyRoad_song_cover.jpg  
    duration        integer  
    example: 259  
    songID         integer  
    example: 123456789  
    songURI        string  
}
```

2.2.6.2 Endpoints πόρου SongEntity

song Operations related to song resource ▾

POST	/song Create song	🔒 ↻
Create a song by uploading a .zip file.		
Parameters		Try it out
Name	Description	
Song * required	The file to upload (.zip).	
file (formData)	<input type="button" value="Επιλογή αρχείου"/> Δεν επιλέχθηκε κανένα αρχείο.	



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

Responses	Response content type
	application/json
Code	Description
201	Created. Example Value Model
	<pre>{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }</pre>
400	Bad Request. Invalid input.

GET	/song/{songID}	Get info of a specific song		
Get the data of a song entity by providing its unique songID				
Parameters	<button>Try it out</button>			
Name	Description			
songID * required <small>string (path)</small>	The unique id of a song songID - The unique id of a song			
Responses	Response content type			
	application/json			
Code	Description			
200	OK.			



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Example Value | Model

```
{  
    "title": "Come together",  
    "artist": "The Beatles",  
    "album": "Abbey Road",  
    "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
    "duration": 259,  
    "songID": 123456789,  
    "songURI": "string"  
}
```

404

Not Found. The songID provided doesn't match any song in the database.

DELETE

/song/{songID} Delete a specific song



Delete the data of a song entity by providing its unique songID.

Parameters

[Try it out](#)

Name

Description

songID * required

The unique id of a song

string

(path)

songID - The unique id of a song

Responses

Response content type

application/json



Code

Description

200

OK. The song was deleted.

404

Not Found. The songID provided doesn't match any song in the database.



2.2.7 Πόρος ProfileEntity

2.2.7.1 Μοντέλο δεδομένων ProfileEntity

```
ProfileEntity {
    username           string
    example: niksaoul
    description        string
    example: Follow me if you love Pantelidis
    followingID       [
        [
            example: List [ 22, 33, 44, 55, 666 ]
            integer
        ]
    ]
    followersID       [
        [
            example: List [ 999, 4444, 44, 234 ]
            integer
        ]
    ]
    coverURI          string
    example: https://mob/media/File:Niksaoul_profile_cover.jpg
    media             [
        [
            [
                icon      string
                example: fb.png
                link      string
                example: www.facebook.com/niksaoul
            ]
        ]
    ]
    playlists         [
        [
            PlaylistEntity {
                playlistID   integer
                example: 123456
                title        string
                example: A playlist that wants to be adored
                userID       integer
                example: 123
                coverURI     string
                example: https://mob/media/File:Adored_playlist_cover.jpg
                description   string
                example: Sometimes I fantasise when the streets are cold and lonely
                songs         [
                    [
                        SongEntity {
                            title      string
                            example: Come together
                            artist    string
                            example: The Beatles
                            album     string
                            example: Abbey Road
                            coverURI  string
                            example: https://mob/media/File:AbbeyRoad_song_cover.jpg
                            duration   integer
                            example: 259
                            songID    integer
                            example: 123456789
                            songURI   string
                        }
                    ]
                ]
                isPublic    boolean
                example: true
                isEmpty    boolean
                example: false
            }
        ]
    ]
    isArtist          boolean
    example: true
    userID            integer
    example: 1312
    email             string
    example: niksaoul@mob.com
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.7.2 Μοντέλο δεδομένων ProfileCreationBody

```
ProfileCreationBody ▼ {  
    username*           string  
                        example: niksaoul  
    userID*            integer  
                        example: 1312  
    isArtist*          boolean  
                        example: true  
    coverURI           string  
                        example: https://mob/media/File:Niksaoul_profile_cover.jpg  
}
```

2.2.7.3 Μοντέλο δεδομένων ChangeProfileBody

```
ChangeProfileInfoBody ▼ {  
    newUsername        string  
                        example: Nikolaos77  
    newDescription     string  
                        example: Hi! I'm Nikos  
    coverURI          string  
                        example: https://mob/media/File:Niksaoul_profile_cover.jpg  
    newMedia           ▼ [ ▼ {  
        icon             string  
                        example: fb.png  
        link             string  
                        example: www.facebook.com/niksaoul  
    }]  
}
```

2.2.7.4 Endpoints πόρου ProfileEntity

profile Operations related to profile resource

POST /profile Create user profile

Create a profile after an account creation.

Parameters

Try it out

Name	Description
profileCreationBody	Profile Information
object (body)	Example Value Model

```
{  
    "username": "niksaoul",  
    "userID": 1312,  
    "isArtist": true,  
    "coverURI":  
    "https://mob/media/File:Niksaoul_profile_cover.jpg"  
}
```

Parameter content type

application/json



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Responses Response content type application/json

Code	Description
201	Created. Example Value Model <pre>{ "username": "niksaoul", "description": "Follow me if you love Pantelidis", "followingID": [22, 33, 44, 55, 666], "followersID": [999, 4444, 44, 234], "coverURI": "https://mob/media/File:Niksaoul_profile_cover.jpg", "media": [{ "icon": "fb.png", "link": "www.facebook.com/niksaoul" }] }</br></br></pre>

PUT /profile/{userID} Change Profile Info [Try it out](#)

Change profile info by providing the userID and the data you wish to change.

Parameters

Name	Description
userID * required integer (path)	userID userID - userID
userProfileInfo * required object (body)	description, username media, profile change Example Value Model <pre>{ "newUsername": "Nikolaos77", "newDescription": "Hi! I'm Nikos", "coverURI": "https://mob/media/File:Niksaoul_profile_cover.jpg", "newMedia": [{ "icon": "fb.png", "link": "www.facebook.com/niksaoul" }] }</pre>

Parameter content type application/json



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Responses Response content type application/json

Code	Description
200	OK. Example Value Model <pre>{ "username": "niksaoul", "description": "Follow me if you love Pantelidis", "followingID": [22, 33, 44, 55, 666], "followersID": [999, 4444, 44, 234], "coverURI": "https://mob/media/File:Niksaoul_profile_cover.jpg", "media": [{ "icon": "fb.png", "link": "www.facebook.com/niksaoul" }] }</pre>
400	Bad Request. Invalid Info.

GET /profile/{userID} Get profile info

Get Profile information by providing the corresponding userID.

Parameters Try it out

Name	Description
userID * required	userID
integer	
(path)	userID - userID



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Responses Response content type application/json

Code	Description
200	OK. Example Value Model <pre>{ "username": "niksaoul", "description": "Follow me if you love Pantelidis", "followingID": [22, 33, 44, 55, 666], "followersID": [999, 4444, 44, 234], "coverURI": "https://mob/media/File:Niksaoul_profile_cover.jpg", "media": [{ "icon": "fb.png", "link": "www.facebook.com/niksaoul" }], }</pre>
404	Not Found.

PUT /profile/follow/{followingID} Follow a user

Follow someone. This userID provided is added in the FollowingID array.

Parameters Try it out

Name	Description
followingID * required integer (path)	The userID of the user you want to follow (followingID) followingID - The userID of the user you want to follow (follow)

Responses Response content type application/json

Code	Description
200	OK.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

DELETE /profile/unfollow/{followingID} Unfollow someone

Delete the userID of the user you want to unfollow from the FollowingsID array.

Parameters

Name	Description
followingID * required <small>integer (path)</small>	The userID of the user you want to unfollow (followingID) followingID - The userID of the user you want to unfollow (fol

Responses Response content type

Code	Description
200	OK.

PUT /profile/followedby/{followersID} Add a new follower

The userID of the user who follows you is added in the FollowersID array.

Parameters

Name	Description
followersID * required <small>integer (path)</small>	The userID of the user who wants to follow you (followersID) followersID - The userID of the user who wants to follow you (

Responses Response content type

Code	Description
200	OK.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

DELETE /profile/unfollowedby/{followersID} Delete a follower

Someone unfollowed you. Remove this followerID from your FollowersID array.

Parameters

Name	Description
followersID * required <small>integer (path)</small>	The userID of the user who wants to unfollow you (followersID) followersID - The userID of the user who wants to unfollow yo

Responses application/json

Code	Description
200	OK.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.8 Πόρος StatisticsEntity

2.2.8.1 Μοντέλο Δεδομένων StatisticsEntity

```
StatisticsEntity ▼ {  
    userID          integer  
    example: 8888  
    followers       integer  
    example: 150000  
    top5           ▼ [  
        example: List [ "Ophelia", "Ho hey", "Sleep on the floor", "Stubborn Love", "Gloria" ]  
        string]  
    gender          ▼ [  
        example: List [ "Male 40%", "Female 60%" ]  
        string]  
    ages            ▼ [  
        example: List [ "20-30: 50%", "30-40: 30%", "40-50: 20%" ]  
        string]  
    location         ▼ [  
        example: List [ "Chicago,US", "Los Angeles,US", "Dallas,US", "New York,US", "Atlanta,US" ]  
        string]  
    monthlyListeners integer  
    example: 2000000  
    imagesURL       ▼ [string  
    example: https://mob/media/File:Lumineers_statistics_graph.jpg]  
}
```

2.2.8.2 Endpoint πόρου StatisticsEntity

statistics Operations related to statistics resource ▼

GET	/statistics	Statistics of a specific Artist.	Try it out
The Artist can view his/her statistics.			
Parameters		Try it out	
No parameters			
Responses		Response content type	application/json ▾
Code	Description		
200	OK.	Example Value Model	



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

```
{  
    "userID": 8888,  
    "followers": 150000,  
    "top5": [  
        "Ophelia",  
        "Ho hey",  
        "Sleep on the floor",  
        "Stubborn Love",  
        "Gloria"  
    ],  
    "gender": [  
        "Male 40%",  
        "Female 60%"  
    ],  
    "ages": [  
        "20-30: 50%",  
        "30-40: 30%",  
        "40-50: 20%"  
    ],  
    "location": [  
        "Chicago,US",  
        "Los Angeles,US",  
        "Dallas,US",  
        "New York,US"  
    ]  
}
```

404 Not Found. This user has no statistics.

2.2.9 Πόρος SearchEntity

2.2.9.1 Μοντέλο δεδομένων SerarchEntity

```
SearchEntity ▼ {  
    songs  
    ▼ [ ▼ {  
        title          string  
        artist         string  
        album          string  
        coverURI       string  
        songURI        string  
    }]  
    albums  
    ▼ [ ▼ {  
        title          string  
        artist         string  
        coverURI       string  
    }]  
    playlists  
    ▼ [ ▼ {  
        title          string  
        coverURI       string  
    }]  
    profiles  
    ▼ [ ▼ {  
        username      string  
        isArtist       string  
        coverURI       string  
    }]  
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.9.2 Endpoint πόρου SerachEntity

search Operations related to search

▼

GET /search Search with a keyword

Search for Songs, Albums, Playlists and Profiles with a keyword.

Parameters

Try it out

Name	Description
keyword * required	Keyword string (query) keyword - Keyword
type * required	Type of searching objects: song, album, playlist, profile. Multiple types must be separated with commas. At least one type is required. string (query) type - Type of searching objects: song, album, playlist, profile.

Responses

Response content type application/json

▼

Code	Description
200	OK. Example Value Model

```
{  
    "songs": [  
        {  
            "title": "string",  
            "artist": "string",  
            "album": "string",  
            "coverURI": "string",  
            "songURI": "string"  
        }  
    ],  
    "albums": [  
        {  
            "title": "string",  
            "artist": "string",  
            "coverURI": "string"  
        }  
    ],  
    "playlists": [  
        {  
            "title": "string",  
            "coverURI": "string"  
        }  
    ]  
}
```

404 Not Found. There are no results matching the keyword.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.10 Πόρος PlayerEntity

2.2.10.1 Μοντέλο δεδομένων PlayerEntity

```
MusicPlayer ▼ {  
    progress          integer  
    example: 69  
   .isPlaying        boolean  
    example: true  
    currentSong      integer  
    example: 3  
    playqueue        PlayqueueEntity ▼ {  
        isEmpty       boolean  
        example: false  
        isShuffle     boolean  
        example: true  
        isRepeat      boolean  
        example: true  
        songs         ▼ [SongEntity ▼ {  
            title        string  
            example: Come together  
            artist       string  
            example: The Beatles  
            album        string  
            example: Abbey Road  
            coverURI    string  
            example: https://mob/media/File:AbbeyRoad_song_cover.jpg  
            duration     integer  
            example: 259  
            songID      integer  
            example: 123456789  
            songURI    string  
        }]  
    }  
}
```

2.2.10.2 Endpoints πόρου PlayerEntity

player Operations related to Music Player

▼

GET /player Playback Information

Playback Information

Get information about the user's current playback.

Parameters

No parameters

Try it out

Responses

Response content type application/json

Code	Description
200	OK.

Example Value | Model



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

```
{
  "progress": 69,
  "isPlaying": true,
  "currentSong": 3,
  "playqueue": {
    "isEmpty": false,
    "isShuffle": true,
    "isRepeat": true,
    "songs": [
      {
        "title": "Come together",
        "artist": "The Beatles",
        "album": "Abbey Road",
        "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",
        "duration": 259,
        "songID": 123456789,
        "songURI": "string"
      }
    ]
  }
}
```

404

Not Found.

PUT /player/play Start a user's playback

Start / Resume a user's playback.

Parameters

No parameters

Try it out

Responses

Response content type application/json

Code	Description
200	OK.

Example Value | Model

```
{
  "progress": 69,
  "isPlaying": true,
  "currentSong": 3,
  "playqueue": {
    "isEmpty": false,
    "isShuffle": true,
    "isRepeat": true,
    "songs": [
      {
        "title": "Come together",
        "artist": "The Beatles",
        "album": "Abbey Road",
        "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",
        "duration": 259,
        "songID": 123456789,
        "songURI": "string"
      }
    ]
  }
}
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

PUT /player/pause Pause a user's playback

Pause a user's playback.

Parameters Try it out

No parameters

Responses Response content type application/json

Code	Description
200	OK. Example Value Model

```
{  
    "progress": 69,  
    "isPlaying": true,  
    "currentSong": 3,  
    "playqueue": {  
        "isEmpty": false,  
        "isShuffle": true,  
        "isRepeat": true,  
        "songs": [  
            {  
                "title": "Come together",  
                "artist": "The Beatles",  
                "album": "Abbey Road",  
                "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg",  
                "duration": 259,  
                "songID": 123456789,  
                "songURI": "string"  
            }  
        ]  
    }  
}
```

PUT /player/next Move to next song

Skip a song and move to the next song of the Playqueue.

Parameters Try it out

No parameters

Responses Response content type application/json



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Code	Description
200	<p>OK.</p> <p>Example Value Model</p> <pre>{ "progress": 69, "isPlaying": true, "currentSong": 3, "playqueue": { "isEmpty": false, "isShuffle": true, "isRepeat": true, "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }] } }</pre>

PUT /player/previous Go back to previous song

Go back to previous song of the Playqueue.

Parameters

No parameters

Try it out

Responses

Response content type

Code	Description
200	<p>OK.</p> <p>Example Value Model</p> <pre>{ "progress": 69, "isPlaying": true, "currentSong": 3, "playqueue": { "isEmpty": false, "isShuffle": true, "isRepeat": true, "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }] } }</pre>



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

PUT /player/**seek** Change the progress variable in the currently playing song.

Change the progress variable in the currently playing song by moving the slider to the desired position.

Parameters

No parameters

Responses Response content type

Code	Description
200	OK.
<pre>{ "progress": 69, "isPlaying": true, "currentSong": 3, "playqueue": { "isEmpty": false, "isShuffle": true, "isRepeat": true, "songs": [{ "title": "Come together", "artist": "The Beatles", "album": "Abbey Road", "coverURI": "https://mob/media/File:AbbeyRoad_song_cover.jpg", "duration": 259, "songID": 123456789, "songURI": "string" }] } }</pre>	



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

2.2.11 Πόρος NewsfeedEntity

2.2.11.1 Μοντέλο δεδομένων NewsfeedEntity

```
NewsfeedEntity ▼ {  
    userID          integer  
    example: 55  
    date            string($date)  
    example: 2020-04-12T23:20:50.52Z  
    followingActivities  
    ▼ [ActivityEntity ▼ {  
        userID          integer  
        example: 9999  
        activityID      integer  
        example: 5555  
        typeOfActivity  string  
        example: uploaded an album  
        date            string($date)  
        example: 2020-04-12T23:20:50.52Z  
    }]  
}
```

2.2.11.2 Μοντέλο δεδομένων NewsfeedCreationBody

```
NewsfeedCreationBody ▼ {  
    followingID*  
    ▼ [  
        example: List [ 99, 7, 66, 23 ]  
        integer]  
}
```

2.2.11.3 Endpoints πόρου NewsfeedEntity

newsfeed Operations related to Newsfeed



POST /newsfeed Creation of the newsfeed

Creation of the newsfeed for the logged in Listener.

Parameters Try it out

Name	Description
newsfeedCreation object (body)	The newsfeed creation body Example Value Model

```
{  
    "followingID": [  
        99,  
        7,  
        66,  
        23  
    ]  
}
```

Parameter content type
application/json



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Responses Response content type application/json

Code	Description
201	Created. Example Value Model <pre>{ "userID": 55, "date": "2020-04-12T23:20:50.52Z", "followingActivities": [{ "userID": 9999, "activityID": 5555, "typeOfActivity": "uploaded an album", "date": "2020-04-12T23:20:50.52Z" }] }</pre>
400	Bad Request. Invalid input.

GET /newsfeed Get the newsfeed object

Get the newsfeed info for the logged in Listener.

Parameters Try it out

No parameters

Responses Response content type application/json

Code	Description
200	OK. Example Value Model <pre>{ "userID": 55, "date": "2020-04-12T23:20:50.52Z", "followingActivities": [{ "userID": 9999, "activityID": 5555, "typeOfActivity": "uploaded an album", "date": "2020-04-12T23:20:50.52Z" }] }</pre>

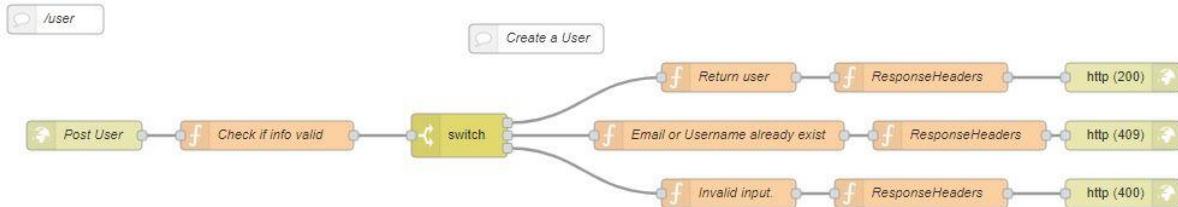


3. Υλοποίηση Συστήματος με Node-RED

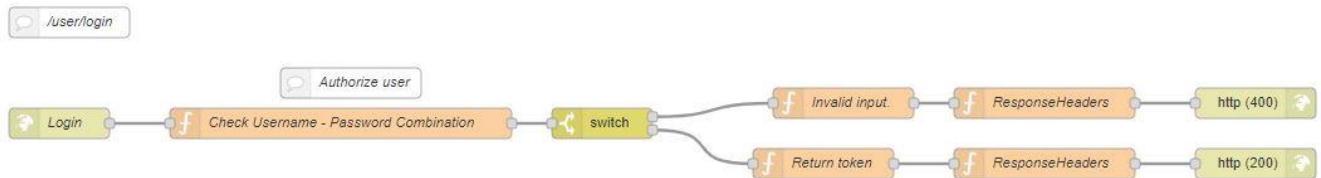
3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRedⁱ

3.1.1 Ροές πόρου UserEntity

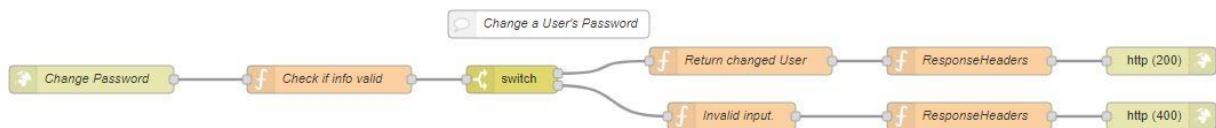
Ροή endpoint post /user



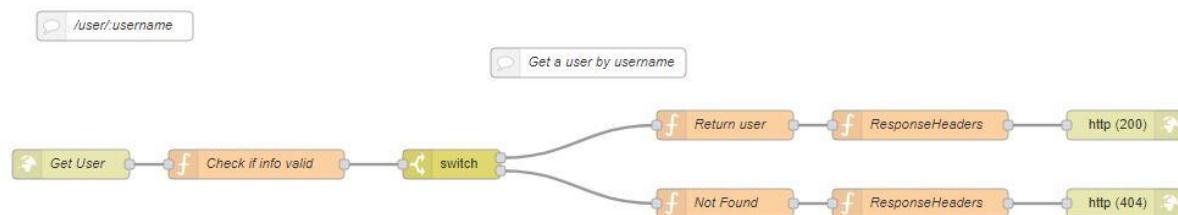
Ροή endpoint put /user/login



Ροή endpoint put /user/{username}



Ροή endpoint get /user/{username}

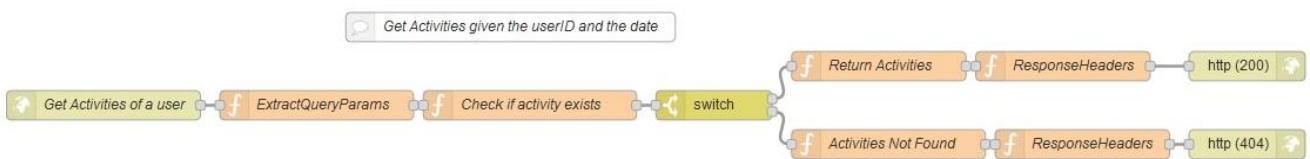


3.1.2 Ροές πόρου ActivityEntity

Ροή endpoint post /activity



Ροή endpoint get /activity





Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

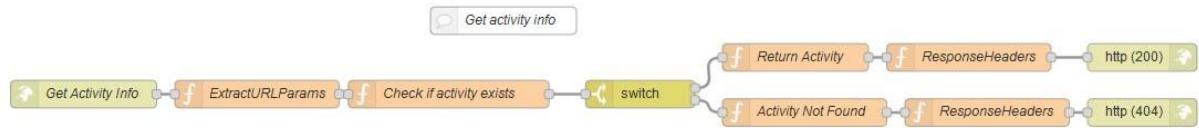
Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

Ποή endpoint get /activity/{activityID}

/activity/activityID

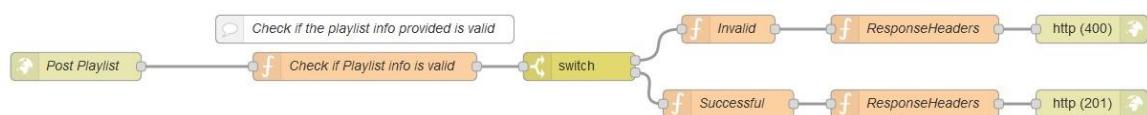


3.1.3 Ροές πόρου PlaylistEntity

Ποή endpoint post /playlist

/playlist

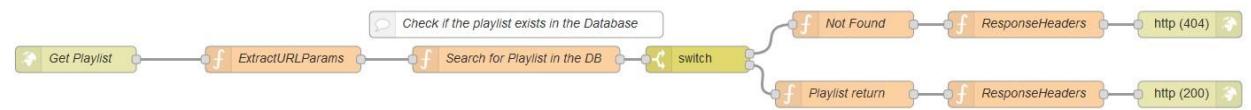
Create a Playlist



Ποή endpoint get /playlist/{playlistID}

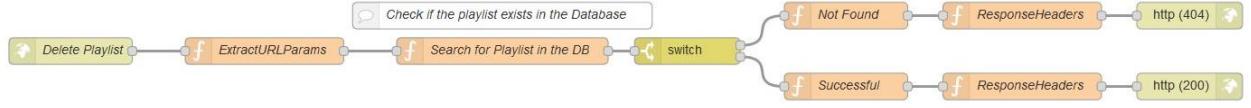
/playlist/playlistID

Get a playlist by its ID



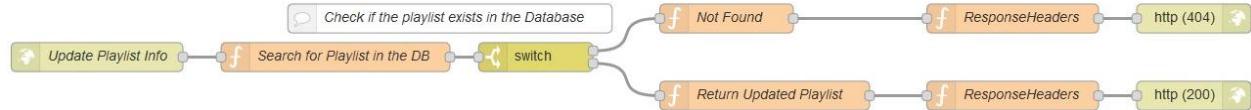
Ποή endpoint delete /playlist/{playlistID}

Delete a playlist



Ποή endpoint put /playlist/{playlistID}

Update a playlist's info



Ποή endpoint delete /playlist/{playlistID}/song/{songID}

/playlist/playlistID/song/songID

Delete a playlist's song





Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

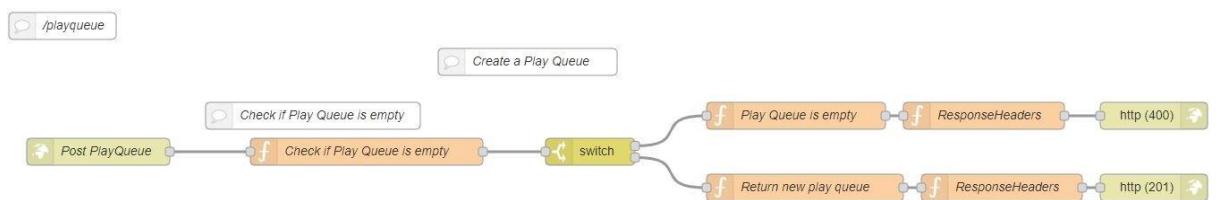
8^ο Εξάμηνο
Άνοιξη 2020

Ποή endpoint put /playlist/{playlistID}/song/{songID}

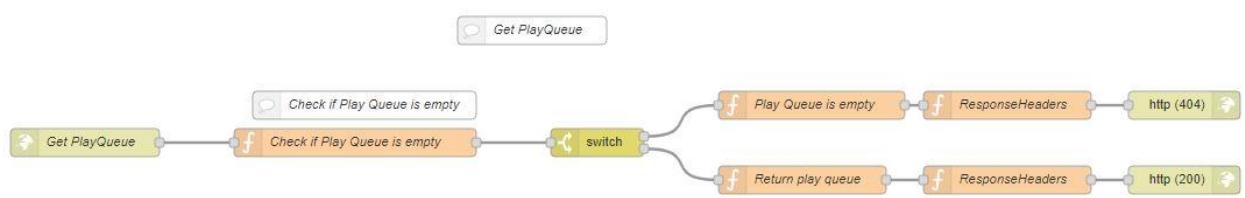


3.1.4 Ροές πόρου PlayqueueEntity

Ποή endpoint post /playqueue



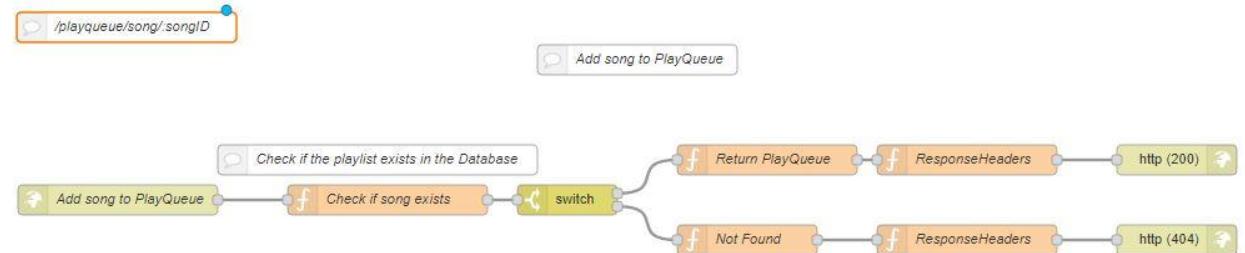
Ποή endpoint get /playqueue



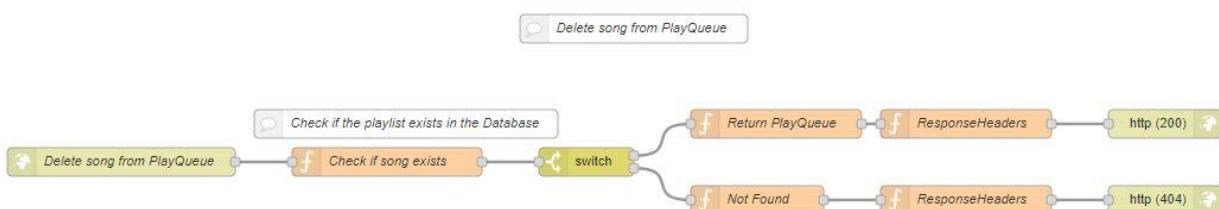
Ποή endpoint put/playqueue



Ποή endpoint put/playqueue/song/{songID}



Ποή endpoint delete /playqueue/song/{songID}





Τεχνολογία Λογισμικού

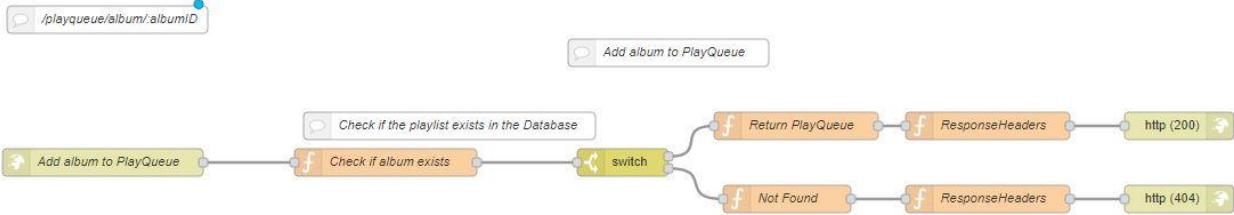
Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

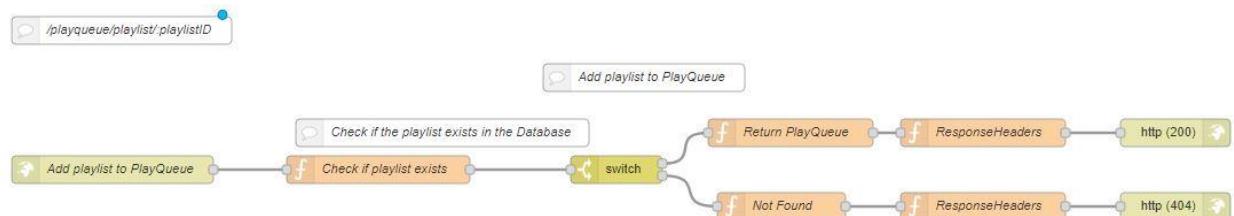
Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Ροή endpoint put /playqueue/album/{albumID}

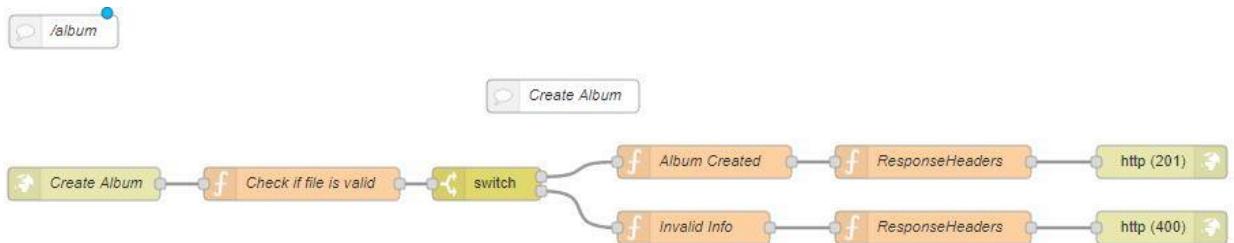


Ροή endpoint put /playqueue/playlist/{playlistID}

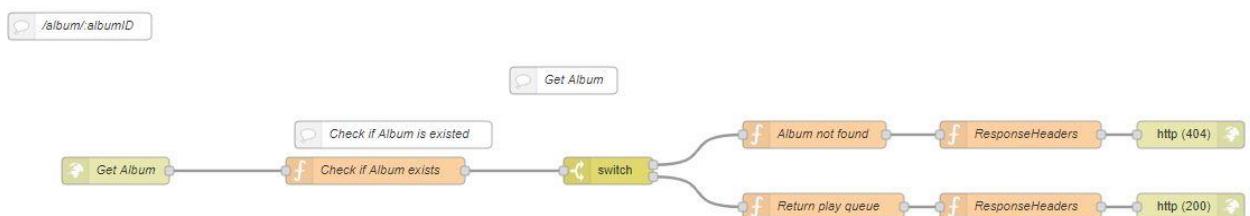


3.1.5 Ροές πόρου AlbumEntity

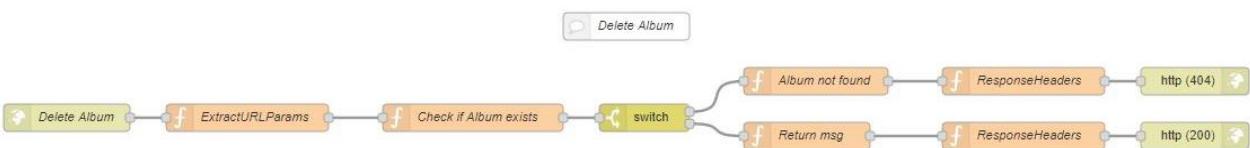
Ροή endpoint post /album



Ροή endpoint get /album/{albumID}

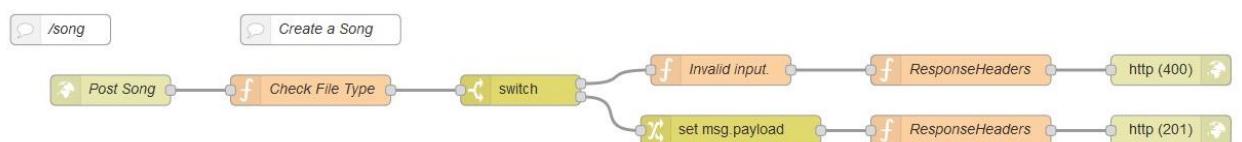


Ροή endpoint delete /album/{albumID}



3.1.6 Ροές πόρου SongEntity

Ροή endpoint post /song





Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

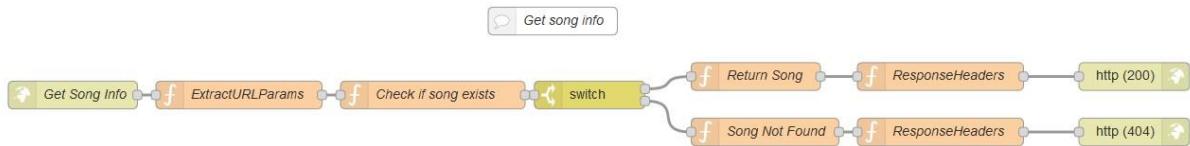
Α.Π.Θ.

8^ο Εξάμηνο

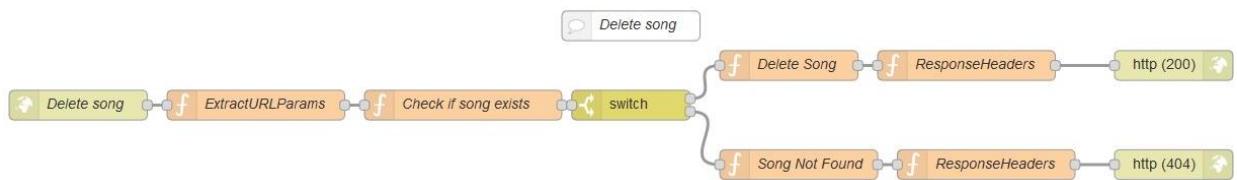
Άνοιξη 2020

Πού endpoint get /song/{songID}

HTTP /song/songID



Πού endpoint delete /song/{songID}

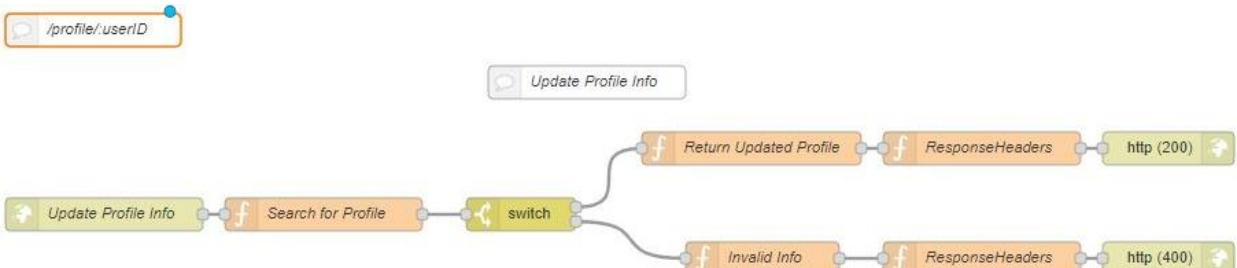


3.1.7 Ροές πόρου ProfileEntity

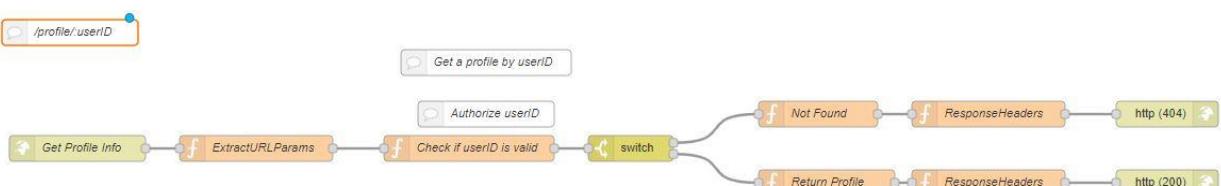
Πού endpoint post /profile



Πού endpoint put /profile/{userID}



Πού endpoint get /profile/{userID}



Πού endpoint put /profile/follow/{followingID}

HTTP /profile/follow/followingID
Follow someone. This userID provided is added in the FollowingID array.

```
graph TD; StartFollowUser[Follow a User] --> ReturnProfileWithUpdatedFollowings[Return Profile with updated Followings]; ReturnProfileWithUpdatedFollowings --> ReturnMessage[Return message]; ReturnMessage --> ResponseHeaders1[ResponseHeaders]; ResponseHeaders1 --> Http200[http (200)];
```

Πού endpoint delete /profile/unfollow/{followingID}

HTTP /profile/unfollow/followingID
Unfollow someone. Delete the userID of the User you want to unfollow from the FollowingID array.

```
graph TD; StartUnfollowUser[Unfollow a User] --> ReturnProfileWithUpdatedFollowingID[Return profile with updated followingID]; ReturnProfileWithUpdatedFollowingID --> ReturnMessage[Return message]; ReturnMessage --> ResponseHeaders1[ResponseHeaders]; ResponseHeaders1 --> Http200[http (200)];
```



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

Ποή endpoint put /profile/followedby/{followersID}

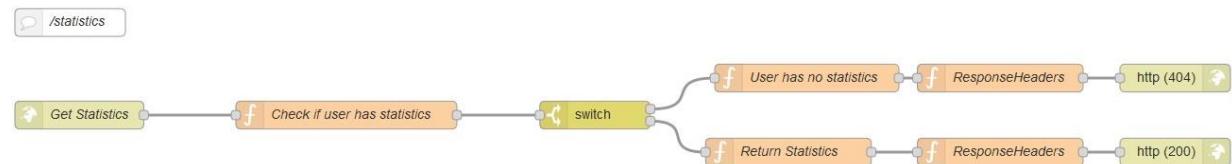


Ποή endpoint delete /profile/unfollowedby/{followersID}



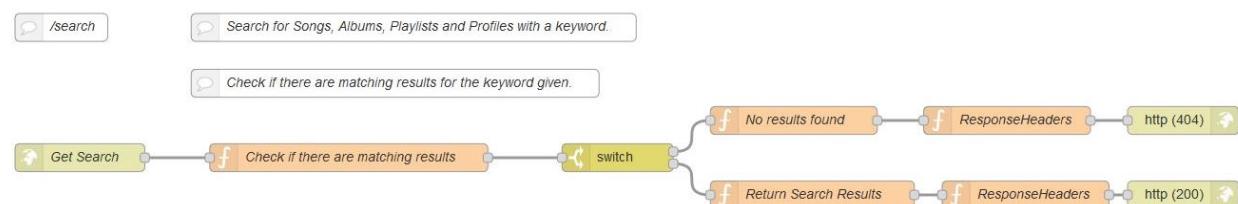
3.1.8 Ροές πόρου StatisticsEntity

Ποή endpoint get /statistics



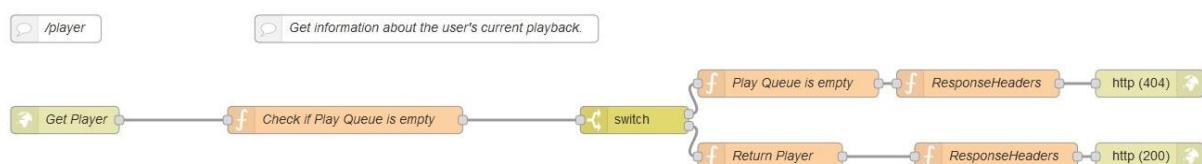
3.1.9 Ροές πόρου SearchEntity

Ποή endpoint get /search



3.1.10 Ροές πόρου PlayerEntity

Ποή endpoint get /player



Ποή endpoint put /player/play



Ποή endpoint put /player/pause





Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2020

Ροή endpoint put /player/next



Ροή endpoint put /player/previous

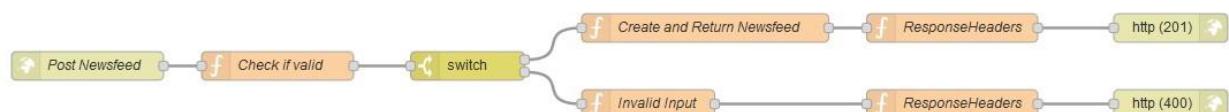


Ροή endpoint put /player/seek



3.1.11 Ροές πόρου NewsfeedEntity

Ροή endpoint post /newsfeed



Ροή endpoint get /newsfeed



3.2 Υλοποίηση Ιστοριών χρήστη

3.2.1 View News Feed



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Newsfeed	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /newsfeed η οποία επιστρέφει ένα αντικείμενο τύπου NewsfeedEntity.
msg.payload	debug	Τυπώνει στην κονσόλα ένα αντικείμενο NewsfeedEntity.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Create Newsfeed	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται ο πίνακας followingID.
http request	http-request	Κάνει κλήση προς την υπηρεσία /newsfeed η οποία επιστρέφει ένα αντικείμενο τύπου NewsfeedEntity.
msg.payload	debug	Τυπώνει στην κονσόλα ένα αντικείμενο NewsfeedEntity.

3.2.2 Search Music or Profiles

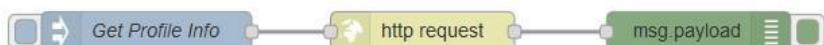


Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Search	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται ένα keyword και οι τύποι αντικειμένων προς αναζήτηση.
http request	http-request	Κάνει κλήση προς την υπηρεσία /search η οποία επιστρέφει ένα αντικείμενο SearchEntity.
msg.payload	debug	Τυπώνει στην κονσόλα ένα αντικείμενο SearchEntity.

3.2.3 View Profile



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Post Profile	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνονται: username, userID, isArtist, coverURI.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile η οποία επιστρέφει ένα αντικείμενο τύπου ProfileEntity.
msg.payload	debug	Τυπώνει στην κονσόλα ένα αντικείμενο ProfileEntity.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Profile Info	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile η οποία επιστρέφει ένα αντικείμενο τύπου ProfileEntity.
msg.payload	debug	Τυπώνει στην κονσόλα ένα αντικείμενο ProfileEntity.



Τεχνολογία Λογισμικού

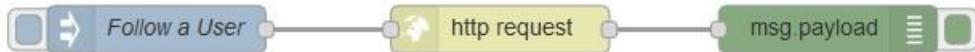
Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

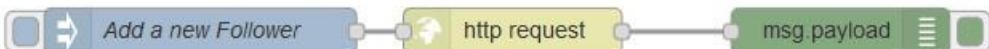
3.2.4 Follow/Unfollow User



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Follow a User	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile/follow/:followingID η οποία προσθέτει το συγκεκριμένο userID στον πίνακα followingID του προφίλ του χρήστη και επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Unfollow a User	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile/unfollow/:followingID η οποία αφαιρεί το συγκεκριμένο userID από τον πίνακα followingID του προφίλ του χρήστη και επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.



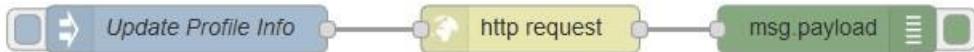
Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Add a new Follower	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile/followedby/:followersID η οποία προσθέτει το συγκεκριμένο userID στον πίνακα followersID του προφίλ του χρήστη και επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Delete a Follower	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile/unfollowedby/:followersID η οποία αφαιρεί το συγκεκριμένο userID από τον πίνακα followersID του προφίλ του χρήστη και επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.

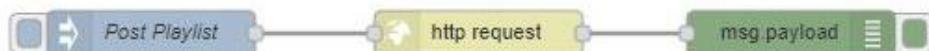


3.2.5 Update Profile Info



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Update Profile Info	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνονται: newUsername, newDescription, coverURI, newMedia.
http request	http-request	Κάνει κλήση προς την υπηρεσία /profile/:userID η οποία ενημερώνει το προφίλ σύμφωνα με τα δεδομένα που παρέχονται στο payload και επιστρέφει το ανανεωμένο ProfileEntity.
msg.payload	debug	Τυπώνει στην κονσόλα το ανανεωμένο ProfileEntity.

3.2.6 Create Public Playlist



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Post Playlist	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνονται: title, description, songs, isPublic, coverURI.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playlist η οποία δημιουργεί ένα αντικείμενο PlaylistEntity σύμφωνα με τα δεδομένα που παρέχονται στο payload, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το νέο PlaylistEntity.

3.2.7 Make Playlist Private



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Update Playlist Info	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνονται: title, description, isPublic, coverURI.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playlist/:playlistID η οποία τροποποιεί ένα αντικείμενο PlaylistEntity με το συγκεκριμένο playlistID σύμφωνα με τα δεδομένα που παρέχονται στο payload, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το ανανεωμένο PlaylistEntity.

3.2.8 Edit Playlist



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Update Playlist Info	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνονται: title, description, isPublic, coverURI.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playlist/:playlistID η οποία τροποποιεί ένα αντικείμενο PlaylistEntity με το συγκεκριμένο playlistID σύμφωνα με τα δεδομένα που παρέχονται στο payload, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το ανανεωμένο PlaylistEntity.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

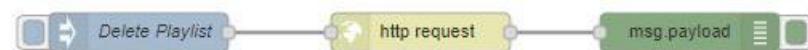


Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Add a song to Playlist	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playlist/:playlistID/song/:songID η οποία τροποποιεί ένα αντικείμενο PlaylistEntity με το συγκεκριμένο playlistID προσθέτοντας το τραγούδι με το αντίστοιχο songID που παρέχεται στο path, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το ανανεωμένο PlaylistEntity.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Delete song from Playlist	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playlist/:playlistID/song/:songID η οποία τροποποιεί ένα αντικείμενο PlaylistEntity με το συγκεκριμένο playlistID διαγράφοντας το τραγούδι με το αντίστοιχο songID που παρέχεται στο path, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το ανανεωμένο PlaylistEntity.

3.2.9 Delete Playlist



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Delete Playlist	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playlist/:playlistID η οποία διαγράφει την Playlist με το συγκεκριμένο playlistID και επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.

3.2.10 Control Music Player



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Player	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /player η οποία επιστρέφει ένα MediaPlayer.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

msg.payload

debug

Τυπώνει στην κονσόλα το MusicPlayer.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Play Current Song	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται η boolean μεταβλητή.isPlaying.
http request	http-request	Κάνει κλήση προς την υπηρεσία /player/play η οποία επιστρέφει ένα MusicPlayer με τα ανανεωμένα Playback Settings.
msg.payload	debug	Τυπώνει στην κονσόλα το MusicPlayer.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Pause Current Song	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται η boolean μεταβλητή.isPlaying.
http request	http-request	Κάνει κλήση προς την υπηρεσία /player/pause η οποία επιστρέφει ένα MusicPlayer με τα ανανεωμένα Playback Settings.
msg.payload	debug	Τυπώνει στην κονσόλα το MusicPlayer.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Play Next Song	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται η μεταβλητή currentSong.
http request	http-request	Κάνει κλήση προς την υπηρεσία /player/next η οποία επιστρέφει ένα MusicPlayer με τα ανανεωμένα Playback Settings.
msg.payload	debug	Τυπώνει στην κονσόλα το MusicPlayer.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Play Previous Song	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται η μεταβλητή currentSong.
http request	http-request	Κάνει κλήση προς την υπηρεσία /player/previous η οποία επιστρέφει ένα MusicPlayer με τα ανανεωμένα Playback Settings.
msg.payload	debug	Τυπώνει στην κονσόλα το MusicPlayer.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Slider	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται η μεταβλητή progress.
http request	http-request	Κάνει κλήση προς την υπηρεσία /player/seek η οποία επιστρέφει ένα



MusicPlayer με τα ανανεωμένα Playback Settings.

msg.payload

debug

Τυπώνει στην κονσόλα το MusicPlayer.

3.2.11 Manage PlayQueue



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Post PlayQueue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνεται ένας πίνακας με αντικείμενα SongEntity.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playqueue η οποία επιστρέφει ένα PlayqueueEntity με τα δεδομένα που δόθηκαν.
msg.payload	debug	Τυπώνει στην κονσόλα το PlayqueueEntity.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get PlayQueue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playqueue η οποία επιστρέφει ένα PlayqueueEntity.
msg.payload	debug	Τυπώνει στην κονσόλα το PlayqueueEntity.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Edit PlayQueue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. Στο payload δίνονται δύο boolean μεταβλητές isShuffle και isRepeat.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playqueue η οποία επιστρέφει ένα PlayqueueEntity με τα ανανεωμένα PlayBack Settings.
msg.payload	debug	Τυπώνει στην κονσόλα το PlayqueueEntity.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Add song to PlayQueue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playqueue/song/:songID η οποία τροποποιεί ένα αντικείμενο PlayQueueEntity προσθέτοντας το τραγούδι με το αντίστοιχο songID που παρέχεται στο path, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το PlayqueueEntity.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

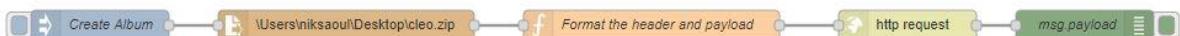


Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Delete song from PlayQueue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playqueue/song/:songID η οποία τροποποιεί ένα αντικείμενο PlayQueueEntity αφαιρώντας το τραγούδι με το αντίστοιχο songID που παρέχεται στο path, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το PlayqueueEntity.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Add playlist to PlayQueue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /playqueue/playlist/:playlistID η οποία τροποποιεί ένα αντικείμενο PlayQueueEntity προσθέτοντας τα τραγούδια της playlist με το αντίστοιχο playlistID που παρέχεται στο path, το οποίο και επιστρέφει.
msg.payload	debug	Τυπώνει στην κονσόλα το PlayqueueEntity.

3.2.12 Upload Music



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Create Album	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
\Users\niksaoul\Desktop\cleo.zip	File in	Διαβάζει ένα .zip αρχείο που περιέχει ένα album από την τοποθεσία που έχει προσδιορίσει ο Καλλιτέχνης.
Format the header and payload	function	Προσδιορίζει το header και το payload.
http request	http-request	Κάνει κλήση προς την υπηρεσία /album η οποία επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Post Song	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
\Users\Τάσος\Desktop\mob\ComeTogether.zip	File in	Διαβάζει ένα .zip αρχείο που περιέχει ένα song από την τοποθεσία που έχει προσδιορίσει ο Καλλιτέχνης.
Format the header and payload	function	Προσδιορίζει το header και το payload.
http request	http-request	Κάνει κλήση προς την υπηρεσία /song η οποία επιστρέφει ένα μήνυμα επιβεβαίωσης.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιβεβαίωσης.

3.2.13 View Audience Stats



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Statistics	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /statistics η οποία επιστρέφει ένα αντικείμενο StatisticsEntity.
msg.payload	debug	Τυπώνει στην κονσόλα το StatisticsEntity.

ⁱ Υπενθυμίζεται ότι οι ροές του Node-RED σε μορφή JSON βρίσκονται [εδώ](#).