Song Recognition Service assignment

1. Build instructions:
   1. Clone git repository <https://github.com/yuliako/Song-Recognition-management.git>
   2. Run mvn clean install
   3. Run SongPlayerManagementApplication.java
   4. Invoke POST "/enforcement/api/v0/getNumberOfAllowedResults" .

Body should contain user id (user\_id)

H2 local DB contains several examples.

For example:

{

"userId": "mosheFree"

}

1. Tools
   1. H2 Database.
      1. H2 Console is enabled – when application is up can be accessed via:

<http://localhost:8080/h2-console>

JDBC url is: jdbc:h2:file:./src/main/resources

User name: sa

Password: sa

* + 1. Data.sql file loads 2 tables with values for testing on application load .

*USER\_USAGE* table– applicative table owned by this microservice

Each request for song will increment the counter of usage - RECOGNITIONS\_NUM column (not implemented).

*USER\_PLAN* table - assumption: the system have a microservice which is responsible for user data ( user details, and all the details about the plan which this particular user have). Our service is using this service.

1. REST exposed:
   1. POST "/enforcement/api/v0/getNumberOfAllowedResults"

Body of the request should have user\_id.

For example:

{

"userId": "mosheFree"

}

**Service Business logic:**

check that user is allowed to get responses (maximum of allowed requests of recognitions is not reached based on user quota .

if allowed, the service will return the number of allowed results according to the user plan. (5 or unlimited (9999))

For this assignment , in case user is not a trial user and doesn’t have any limitation for the number of songs in response , the service will return a predefined max number as 9999.

* 1. POST "/enforcement/api/v0/getNumberOfAllowedRequests"

( since I developed a logic for getting allowed number of song recognitions, in addition I exposed this as a REST in case user will want to check number of remained\allowed request he can perform.)

Body of the request should have user\_id.

For example:

{ "userId": "mosheFree"}