**Musify**

**Overview**

Musify is a service for listening to music. You will have to implement the administrative part of it, not the streaming itself.

The app can do the following:

* manage users (register new users, login, update, inactivate a user)
* manage artists (create, update)
* manage songs, albums (create/update)
* manage playlists (create/update/delete)
* have a search capability for searching artists, albums, songs and public playlists.
* the users should be able to:
  + search songs, artists, albums (text search, by title, name, etc)
  + search public playlists
  + view an artist’s details, view all albums of that artist, view all songs of that artist
  + view all songs of an album
  + view playlists (own playlists and the ones that he/she is following)
  + add/remove songs to playlists; change order of a songs
  + view all songs in a playlist

**Users**

Users are persons that will use the service.

Users must register into the application and can delete their accounts. The information about the users should not be deleted from the database, instead they should only be marked as deleted.

Users must login into the app before using it. They will be using a username and password, and the login will return a “token”. This can be a hash-ed version of the userId. This token will be passed on all EndPoints that require a user identification. The hash can be stored in the database.

Users can have roles (admin or regular user). Some operations can only be performed by admins. Creating artists, albums and songs can only be done by admins.

A User can be deactivated (login is no longer possible).

A User needs to have the following details:

* first name, last name
* email (used also as username)
* password (should be encrypted/hashed when saved into DB)
* country of origin
* role (admin or regular)

**Artists**

Artists can be persons or bands. For bands we need to capture the members of the band.

An Artist needs to have the following details:

* first name and last name for persons
* stage name (for persons)
* band’s name (for bands)
* birthday – for persons
* location (city or country of origin) – for bands
* active period (start and maybe end date; should not be precise, maybe we know only the year, or year/month)
* type (person or band)

**Songs**

A Song needs to have the following details:

* title
* alternative titles (the alternate title can be another title or a translation of the original title. The translation should also have the language in which the title is translated)
* artists (one or more)
* duration
* creation date

**Albums**

They represent a list of songs. The order of songs in the album should be persistent.

Albums have:

* title
* description
* artist (only one artist)
* genre
* release date
* label

**Playlists**

Playlists are lists of songs, created by users. The order of the songs in the playlist should be preserved.

Users can create their own playlists. The playlist can be private (it can only be seen by the user that created it), or public (other users can see that playlist).

A user has a list of playlists (their own playlists or public ones that he/she is following).

Only the owner of the playlist can modify a public playlist.

A playlist can be created by adding songs to it or an album. If an album is used, all the songs of that album should be added to the list, in the order presented in the album.

Playlist can have:

* name
* owner user
* type (private or public)
* created date
* last updated date