Music Library Documentation

Overview

The Music Library is like a smart computer program made with Java and Spring Boot. Its main job is to help organize a bunch of music stuff like artists, albums, songs, playlists, podcasts, hosts, labels, ads, and users. Think of it as a cool system that manages all your favorite music in one place. You can control it in two ways: through a simple terminal interface on your computer or by connecting it with other programs through a REST service. So, whether you're a music lover or a tech person, this Music Library has got something for everyone!

Features

Admin Operations:
Manage admin sessions, allowing login and logout.

adminLogin –email –password
adminLogout

Advertisement Operations:
Add, find, and delete advertisements.

addAd –name –length –type –releaseDate (admin only)
findAd –name (admin only)
deleteAd –id (admin only)
listAds (admin only)

Album Operations:
List, add, find, and delete albums.

listAlbums
addAlbum –name –artistId –songIds (admin only)
deleteAlbum –id (admin only)
findAlbum –name

Artist Operations:
Add, delete artists, view artist followers, and find artists.

addArtist –name –birthdate (admin only)
findArtist –name
getFollowers –artistId (admin only)
deleteArtist –id (admin only)

Host Operations:
List hosts, find hosts, delete hosts and manage host podcasts.

listHosts (admin only)
addHost –name –birthdate (admin only)
deleteHost –hostId (admin only)
findHost –name (admin only)

Label Operations:
Manage labels, add artists to labels, and find labels.

listLabels (admin only)
addLabel –name (admin only)
deleteLabel –labelId (admin only)
findLabel –name (admin only)
addArtistToLabel –artistId –labelId (admin only)

Normal User Operations:
Manage user sessions, follow/unfollow artists, view notifications, add, delete, and update users.

listUsers (admin only)
addUser –name –email –password –birthdate –isPremium (admin only)
deleteUser –userId (admin only)
updateUser –password –isPremium (admin only)
findUser (admin only)
login –email –password (no current user logged in)
logout
getCurrentUser
followArtist –artistId (normal user only)
unfollowArtist –artistId (normal user only)
seeNewNotifications (normal user only)

Playlist Operations:
List, add, delete, and update playlists, add and remove songs from playlists.

listPlaylists (normal user only)
addPlaylist –name (normal user only)
deletePlaylist –playlistId (normal user only)
updatePlaylist –id (normal user only)
addSongToPlaylist –songId –playlistId (normal user only)
removeSongFromPlaylist –songId –playlistId (normal user only)

Podcast Operations:
Play, add, delete, and find podcasts, adjust podcast playback speed, and add ads to podcasts.

listPodcasts
addPodcast –name –length –topic –releaseDate –hostId (admin only)
deletePodcast –podcastId (admin only)
findPodcast –name
addAdToPodcast –adId –podcastId (admin only)
playPodcast –name (normal user only)
playPodcastSpeed –name –speed (only premium normal users)

Song Operations:
Play, add, delete, and find songs, and list all songs.

listSongs
addSong –name –genre –length –releaseDate –artistId (admin only)
findSong –name
playSong –name (admin only)
deleteSong –id (admin only)

Permission system

In the Music Library, user roles play a crucial role in defining permissions. Admins and regular users are distinct roles with varying levels of access and control. Admins have elevated privileges, allowing them exclusive rights to add, delete, and edit various entities such as songs, albums, artists, playlists, podcasts, and more. Regular users, on the other hand, enjoy the ability to interact with the library by playing songs, following artists, managing playlists, and receiving notifications. This clear distinction in permissions ensures the secure and efficient management of the music library, empowering admins

with the authority to oversee and manipulate entities, while users engage with the content within defined boundaries.

Design patterns

The Music Library utilizes various design patterns to enhance its structure and functionality, ensuring a robust and adaptable codebase.

Singleton

The Singleton pattern is employed for the User Session entity and repositories, guaranteeing a single instance throughout the application, promoting centralized control.

Strategy

For playing songs with or without ads, the Strategy pattern allows dynamic switching between different playback strategies, ensuring adaptability and a seamless user experience.

Decorator

Podcasts benefit from the Decorator pattern, offering the flexibility to play them at normal or higher speeds, enhancing the overall listening experience.

Observer

The Observer pattern notifies users of new artist activity, fostering a more engaging and connected user experience by efficiently communicating updates.

Builder

The Builder pattern constructs podcasts and albums step by step, providing a clear and consistent way to create complex objects with varying attributes.

Factory

For creating hosts and artists, the Factory pattern encapsulates instantiation logic, ensuring a centralized and consistent approach, enhancing code maintainability and scalability.

These design patterns collectively contribute to the Music Library's architecture, creating a user-friendly and feature-rich environment for managing music entities.

Usage

Terminal CLI

To interact with the Music Library using the terminal CLI, follow these steps:
Open your terminal.
Navigate to the project directory.
Run the application using ./mvnw spring-boot:run.
Use the available commands to manage the music library.

REST Service

The Music Library also exposes a REST service for programmatic access. Refer to the API documentation for details on how to use the endpoints.

Database Connection

The Music Library utilizes a database to store and retrieve information. Configure the database connection in the application.properties file or provide environment-specific configurations.

Testing

We've thoroughly tested the Music Library to ensure it works smoothly and reliably. Our tests cover user interactions, database operations, and API functionalities. This process helps us find and fix any potential issues, making sure you have a trouble-free experience while managing your music collection. Your confidence in the Music Library is crucial, and our commitment to testing ensures a reliable and user-friendly platform for you.

Thank you for using our Music Library. If you encounter any issues or have suggestions, please refer to the GitHub repository for support.
https://github.com/Konolin/music-library