

Music Web Application API Documentation

Table of Contents

1.	User Authentication
	<ul style="list-style-type: none">• Logout• Login• SignUp
2.	Song Recommendation
	<ul style="list-style-type: none">• Recommend Song• Recommend last liked song from friend
3.	User Activities
	<ul style="list-style-type: none">• (Get) Follow/Unfollow• Get User's Liked Songs• Create Playlists• Add songs to created playlists• Increase Popularity• Decrease Popularity• Rate Song
4.	Search Functionality
	<ul style="list-style-type: none">• Search User• Search Song• Search Tracks by Artist
5.	Data Analysis
	<ul style="list-style-type: none">• Get Higher Rated Genre• Get Genre Percentage• Get Average Release Year
6.	Other Endpoints
	<ul style="list-style-type: none">• Fetch from ID• Search Song from Database• Add track to Database• Delete track form Database• Extract track from file• Extract track manually form input

User Authentication

SIGNUP

- **Endpoint:** /signup
- **Method:** POST
- **Parameters:**
 - `username` (string): User's unique username.
 - `password` (string): User's password.
 - `Password2` (string): User's password for matching.
 - `name` (string): User's name.
 - `surname` (string): User's surname.
- **Response:**
 - `message` (string): Registration success or failure message.

Login

- **Endpoint:** /login
- **Method:** POST
- **Parameters:**
 - `username` (string): User's username.
 - `password` (string): User's password.
- **Response:**
 - `message` (string): Login success or failure message.

Checking through session

Logout

- **Endpoint:** /logout
- **Method:** POST
- **Parameters:**
 - `username` (string): User's username.
 -
- **Response:**
 - `message` (string): Logout success or failure message.

Checking through session

Song Recommendation

Recommend Song

- **Endpoint:** `/recommend_song`
 - **Method:** `GET`
 - **Response:**
 - `recommended_song` (string): Recommended song.
 - `genre` (string): Genre of the recommended song.
- ```

{
 "genre": "pop",
 "recommended_song": "Is It Over Now? (Taylor's Version) (From The Vault)"
}

```

## Recommend Last Liked Song from Friend

- **Endpoint:** `/recommend_last_liked_song_from_friend`
- **Method:** `GET`
- **Response:**
  - `friend` (string): Friend's username.
  - `added` (string): Last liked song by the friend.
  - `by` (string): Artist of the last liked song

```

{
 "added": "Your Latest Trick",
 "by": "Dire Straits",
 "friend": "baran"
}

```

## User Activities

### Follow

- **Endpoint:** `/add_followings`
- **Method:** `POST`
- **Parameter(S):** `target_username`
- **Response:** `message` (string): follow success or failure message

### Unfollow

- **Endpoint:** `/unfollow`
- **Method:** `POST`
- **Parameter(S):** `target_username`
- **Response:** `message` (string): unfollow success or failure message

## Get User's Liked Songs

- **Endpoint:** `/get_users_liked_songs`

- **Method:** GET
- **Parameters:**
  - `username` (string): User's username.
- **Response:**
  - `liked_songs` (list): List of liked songs.

## Create Playlist

- **Endpoint:** / `create_playlist`
- **Method:** POST
- **Parameters:**
  - `playlist_name` (string): User's wanted playlist name.
- **Response:** `message` (string): creating playlist success or failure message

## Add Songs to Created Playlist

### Endpoint

- **Endpoint:** /`add_songs_to_playlist`
- **Method:** POST

### Parameters.

- `playlist_id` (string): ID of the playlist to which songs will be added.
- `track` (list): List of songs to be added to the playlist. Each song is represented as a dictionary with the following keys:
  - `name` (string): Name of the song.
  - `artists` (list): List of artists for the song, where each artist is represented as a dictionary with the following keys:
    - `name` (string): Name of the artist.
    - `id` (string): ID of the artist.
    - `Genre` (list): genre(s) of the artist.
  - `album` (dictionary): Information about the album, with the following keys:
    - `name` (string): Name of the album.
    - `release_date` (string): Release date of the album.
  - `duration_ms` (int): Duration of the song in milliseconds.
  - `popularity` (int): Popularity score of the song.

### Response

- `message` (string): Success or failure message.

## Increase Popularity of a Track

### Endpoint

- **Endpoint:** `/increase_rate/<track_name>`
- **Method:** `GET`

### Parameters

- `track_name` (string): The name of the track for which the popularity will be increased.

### Response

- `message` (string): Success message indicating the increased popularity.

## Decrease Popularity of a Track

### Endpoint

- **Endpoint:** `/decrease_rate/<track_name>`
- **Method:** `GET`

### Parameters

- `track_name` (string): The name of the track for which the popularity will be decreased.

### Response

- `message` (string): Success message indicating the decreased popularity.

## Search Functionality

### Search User

- **Endpoint:** `/search_user`
- **Method:** `POST`
- **Parameters:**
  - `username` (string): Search term.

- **Response:**
  - `results` (list): List of user search results.

## Search Song

- **Endpoint:** `/search_song`
- **Method:** `POST`
- **Parameters:**
  - `song_name` (string): Search term.
- **Response:**
  - `results` (list): List of song search results.

## Search Tracks by Artist

- **Endpoint:** `/search_tracks_by_artist`
- **Method:** `POST`
- **Parameters:**
  - `artist_name` (string): Search term.
- **Response:**
  - `results` (list): List of song search results.

## Data Analysis

### Get Higher Rated Genre

- **Endpoint:** `/get_higher Rated Genre`
- **Method:** `GET`
- **Parameters:**
  - `username` (string): User's username.
- **Response:**
  - `higher Rated Genre` (string): Genre with the highest average rating.
  - `average_rating` (float): Average rating for the genre.

### Get Genre Percentage

- **Endpoint:** `/get_genre_percentage`
- **Method:** `GET`
- **Parameters:**
  - `username` (string): User's username.
- **Response:**
  - `genre_percentage` (dict): Dictionary containing genre and its percentage rating.

## Get Average Release Year

- **Endpoint:** `/get_average_release_year`
- **Method:** `GET`
- **Parameters:**
  - `username` (string): User's username.
- **Response:**
  - `most_liked_average_year` (int): Most liked average release year.

## Other Endpoints

### Fetch from ID

- **Endpoint:** `/fetch_from_id`
- **Method:** `GET`
- **Parameters:**
  - `id` (string): Song ID.
- **Response:**
  - Custom response for fetching a song with a specific ID.

### Search Song from Database

- **Endpoint:** `/search_song_from_db`
- **Method:** `GET`
- **Parameters:**
  - `song_name` (string): Search term.
- **Response:**
  - Custom response for searching a song in the database.

## Add Tracks to Database

### Endpoint

- **Endpoint:** `/add_tracks_to_db/<artist_name>`
- **Method:** `GET`

### Parameters

- `artist_name` (string): The name of the artist for whom top tracks will be fetched and added to the database.

## Response

- `message` (string): Success message indicating whether the operation was successful or failed.

## Delete Track

### Endpoint

- **Endpoint:** `/delete_track`
- **Method:** `DELETE`

### Parameters

- `song_name` (string): The name of the track to be deleted.
- `artist_name` (string): The name of the artist associated with the track.

## Response

- `message` (string): Success message indicating whether the track was deleted successfully or not found.

## Extract Track Info from Text File

### Endpoint

- **Endpoint:** `/extract_track_info`
- **Method:** `POST`

### Parameters

- `file` (file): The text file containing track information.

## Response

- `inserted_data` (list): A list of MongoDB ObjectIds for the inserted track data.

## Add Track Manually



## Endpoint

- **Endpoint:** /add\_track\_man
- **Method:** POST

## Form Parameters

- **album\_id** (string): The ID of the album.
- **album\_name** (string): The name of the album.
- **release\_date** (string): The release date of the album.
- **artist\_id** (string): The ID of the artist.
- **artist\_name** (string): The name of the artist.
- **num\_artists** (integer): The number of artists involved in the track.
- **duration\_ms** (integer): The duration of the track in milliseconds.
- **track\_id** (string): The ID of the track.
- **track\_name** (string): The name of the track.
- **popularity** (integer): The popularity score of the track.

## Response

- **message** (string): A message indicating whether the track was added successfully.
- **track\_id** (string): The ObjectId of the inserted track data.