**Music Web Application API Documentation**

**Table of Contents**

1. [User Authentication](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#user-authentication)
   * Logout
   * [Login](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#login)
   * SignUp
2. [Song Recommendation](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#song-recommendation)
   * [Recommend Song](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#recommend-song)
   * Recommend last liked song from friend
3. [User Activities](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#user-activities)
   * (Get) Follow/Unfollow
   * [Get User's Liked Songs](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#get-users-liked-songs)
   * Create Playlists
   * Add songs to created playlists
   * Increase Popularity
   * Decrease Popularity
   * Rate Song
4. [Search Functionality](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#search-functionality)
   * [Search User](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#search-user)
   * [Search Song](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#search-song)
   * [Search Tracks by Artist](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#search-tracks-by-artist)
5. [Data Analysis](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#data-analysis)
   * [Get Higher Rated Genre](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#get-higher-rated-genre)
   * [Get Genre Percentage](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#get-genre-percentage)
   * [Get Average Release Year](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#get-average-release-year)
6. [Other Endpoints](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#other-endpoints)
   * [Fetch from ID](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#fetch-from-id)
   * [Search Song from Database](https://chat.openai.com/c/a3941096-c304-48a2-8ef4-f6de2aebd9c3#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.