Laulu Api - Documentation

Introduction

This is the documentation for the Laulu API. Laulu API is an API that helps you set up a register of users that includes username, password and display name. It also provides boards, reviews and images for said users.

Resource Formats

The default resource format is JSON (application/json)

Prefix

Filezilla prefix/server/

End Points

End Point: addBoardOrReview.php

This endpoint is used to create a new board or review.

!important when creating a new board, the body must be sent as FromData and include a file input. If an image should be added with the board, the only file types allowed are “jpg”, “jpeg” & “png” and the uploaded file’s name has to be specified as “imageInput”.

Request - New board

- URL: prefix/addBoardOrReview.php
- Method: POST
- Body parameters:
    - `nameInput (string, required): The name of the board.`
    - `userId (string, required): The id of the user.`

Request - New review

All of the keys sent below are required for the request to work properly

- URL: prefix/addBoardOrReview.php
- Method: POST
- Body parameters:
    - `Rating (integer): rating of album`
    - `reviewDescription (string): description of review`
    - `boardId (integer): Id of board`
    - `userId (string): The id of the user`
    - `albumName (string): Name of the album`
    - `artistName (string): Name of the artist`
    - `albumId (string): Id of album`

Response

Success (status code 200):

```
{"message": "Board added!"}
```

Error - No input added (status code 405):

```
{"message": "Invalid Method"}
```

Error - No input added (status code 405):

```
{"message": "No input added"}
```

Error - A board with the same name already exists (status code 200):

```
{"message": "A board with that name already exists"}
```

Error - Photo size too large (status code 200):

```
{
  "message": "Image size too large"
}
```

Error - No input added (status code 405):

```
{"message": "You've already reviewed this album in this board"}
```

Error - Failed to upload file (status code 400):

```
{"error": "Failed to upload file"}
```

End Point: addToListenList.php

This endpoint is used to manage a user's favourites, including adding and removing albums from their favourites list.

Request

- URL: prefix/addToListenList.php
- Headers: "Content-Type: application/json"
- Method: POST
- Body parameters:
  - `userId (string, required): The ID of the user.`
  - `album (object, required): The album details to be added or removed from favourites.`
    - `albumName (string, required): The name of the album.`
    - `artist (string, required): The artist of the album.`
    - `albumId (string, required): The ID of the album.`
    - `albumCover (string, required): The URL or path of the album cover image.`

Response

Success - Album added to favourites (status code 200):

```
{"message": "success"}
```

Success - Album removed from favourites (status code 202):

```
{"message": "success"}
```

Error - Invalid or missing inputs (status code 405):

```
{"message": "Invalid, missing inputs"}
```

Error - Internal error: Issues with users (status code 200):

```
{"message": "Internal error: Issues with users"}
```

End Point: deleteReview.php

This endpoint is used to delete a user's review from their album data.

Request

- URL: prefix/deleteReview.php
- Headers: "Content-Type: application/json"
- Method: DELETE
- Body parameters:
    - `userId (string, required): The id of the user.`
    - `reviewId (string, required): The id of the review to be deleted.`

Response

Success - Review deleted (status code 200):

```
{"message": "success"}
```

Error - Invalid HTTP method (status code 405):

```
{"message": "Invalid HTTP-method"}
```

End Point: follow.php

This endpoint is used to follow or unfollow another user.

Request

- URL: prefix/follow.php
- Headers: “Content-Type: application/json”
- Method: POST
- Body parameters:
    - `id (string, required): The ID of the user to follow/unfollow.`
    - `currentUserId (string, required): The ID of the currently logged-in user.`

Response

Success - User followed/unfollowed (status code 200):

```
{"message": "success"}
```

Error - Invalid method (status code 405):

```
{"message": "Invalid method"}
```

Error - Invalid Content-Type (status code 415):

```
{"message": "invalid Content-Type"}
```

Error - Missing IDs (status code 400):

```
{"message": "Missing ids"}
```

End Point: getReviews.php

This endpoint is used to retrieve user reviews or album reviews.

Request - Get all reviews of one user

- URL: prefix/getReviews.php/?id=string
- Method: GET

- Body parameter:
    - `id (string, required): The ID of the user to retrieve reviews for.`

Request - Get all reviews of one album

- URL: prefix/getReviews.php/?albumId=string
- Method: GET
- Body parameter:
    - `albumId (string, required): The ID of the album to retrieve reviews for.`

Response - Get all reviews of one user

Success - User Reviews Found (status code 200):

```
[
    {
      "albumName": "Album Name",
      "albumId": "album123",
      "timestamp": 1621849876,
      "displayName": "John Doe",
      "userId": "user123",
      "artist": "Artist Name",
      "rating": 4.5,
      "reviewDescription": "This album is great!",
      "reviewId": "review123",
      "albumCover": "https://example.com/album-cover.jpg"
    },
    ...
  ]
```

Error - User not found (status code 404):

```
{"message": "User Not Found"}
```

Error - Missing GET input (status code 405):

```
{"message": "Missing get input"}
```

Error - Invalid method (status code 405):

```
{"message": "You need to use the GET method"}
```

Response - Get all reviews of one album

Success - Album reviews found (status code 200):


```
{
  "averageRating": 4.2,
  "totalReviews": 20,
  "reviews": [
    {
      "albumName": "Album Name",
      "albumId": "album123",
      "timestamp": 1621849876,
      "displayName": "John Doe",
      "userId": "user123",
      "artist": "Artist Name",
      "rating": 4.5,
      "reviewDescription": "This album is great!",
      "reviewId": "review123",
      "albumCover": "https://example.com/album-cover.jpg",
      "userProfilePicture": "https://example.com/user-profile.jpg"
    },
    ...
  ]
}
```


Error - No rating score yet (status code 200):

```
{"message": "No rating score yet"}
```

Error - Missing GET input (status code 405):

```
{"message": "Missing get input"}
```

Error - Invalid method (status code 405):

```
{"message": "You need to use the GET method"}
```

End Point: getUser.php

This endpoint is used to retrieve details of a user.

Request

- URL: prefix/getUser.php/?id=string
- Method: GET
- Body parameter:
    - `id (string): The id of the user to retrieve details for.`

Response

Success - User details found (status code 200):

```
{
  "userIdentity": {
    "id": "user123",
    "displayName": "John Doe",
    "profilePic": "https://example.com/profile-pic.jpg"
  },
  "userSocial": {
    "followers": ["follower1", "follower2"],
    "following": ["following1", "following2"]
  },
  "albumData": {
    "reviews": [
      {
        "albumName": "Album Name",
        "albumId": "album123",
        "timestamp": 1621849876,
        "artist": "Artist Name",
        "rating": 4.5,
        "reviewDescription": "This album is great!",
        "reviewId": "review123",
        "albumCover": "https://example.com/album-cover.jpg"
      },
      ...
    ],
    "boards": [
```

```
        {
          "boardId": "board123",
          "boardName": "Board Name",
          "reviews": ["review123", "review456"]
        },
        ...
      ]
    }
}
```

Error - Invalid HTTP method (status code 405):

```
{"message": "Invalid HTTP-method"}
```

Error - Missing user ID (status code 400):

```
{"message": "Missing id input key"}
```

End Point: login.php

This endpoint is used for user login.

Request

There are two ways of using the login request, This is done via different values for the access key.

Value of the access key must be the following for using the regular Login method

!important: when using the login method the loginTokenKey may not be included

Access: (string)”Access-Login: Auth”

- URL: prefix/login.php
- Headers: "Content-Type: application/json"
- Method: POST
- Body parameters:
    - username (string): The username of the user.
    - password (string): The password of the user.
    - access (string): Access type specifying login method

Value of the access key must be the following for using the key method

Access: (string)”Access-Key: Auth”

- URL: prefix/login.php
- Headers: "Content-Type: application/json"
- Method: POST
- Body parameters:
    - access (string): Access type specifying login method
    - loginTokenKey (string): The username of the user.

Response

Success - User Logged In (status code 202):

```
{
  "displayName": "John Doe",
  "id": "12345",
  "profilePicture": "profile_picture.jpg"
}
```

Success - User Logged In with Token (status code 200):

```
{
  "displayName": "John Doe",
```

```
  "id": "12345",
  "profilePicture": "profile_picture.jpg"
}
```


Error - Invalid HTTP Method (status code 405):


```
{"message": "Invalid method"}
```


Error - Invalid Content Type (status code 415):


```
{ "message": "Invalid Content-Type"}
```


Error - Empty Username or Password (status code 400):


```
{"message": "Empty username or password"}
```


Error - Incorrect Username or Password (status code 400):


```
{"message": "Incorrect Username or password"}
```


Error - Authentication Failed (status code 400):


```
{"message": "Authentication failed"}
```

End Point: register.php

This endpoint is used to register a new user.

`access (string): The access key required for registration (must be "Access-Register: Auth").`

Request

- URL: prefix/register.php
- Headers: "Content-Type: application/json"
- Method: POST
- Body parameters:
    - `username (string, required): The username of the user.`
    - `password (string, required): The password of the user.`
    - `displayName (string, required): The display name of the user.`
    - `access (string, required): "Access-Register: Auth".`

Response

Success - User Registered (status code 202):

```
{"message": "Registered!"}
```

Error - Invalid HTTP method (status code 405):

```
{"message": "Invalid method"}
```

Error - Invalid Content-Type (status code 415):

```
{"message": "Invalid Content-Type"}
```

Error - Missing or Invalid Access Key (status code 401):

```
{"message": "Authentication failed"}
```

Error - Empty input <3 (status code 400):

```
{"message": "Please leave no field empty <3"}
```

Error - Spaces in input (status code 400):

```
{"message": "Username and password may not contain spaces"}
```

Error - Username longer than 25 characters (status code 400):

```
{"message": "Username may not be longer than 25 characters"}
```

Error - Password longer than 30 characters (status code 400):

```
{"message": "Password may not be longer than 30 characters"}
```

Error - Display name longer than 25 characters (status code 400):

```
{"message": "DisplayName may not be longer than 25 characters"}
```

Error - Username already exists (status code 400):

```
{"message": "Username already in use"}
```

Error - Password already exists (status code 400):

```
{"message": "Password already in use"}
```

Error - DisplayName already exists (status code 400):

```
{"message": "DisplayName already in use"}
```

Error - Internal server issue (status code 500):

```
{
  "message": "We're very sorry to inform you that we have an internal
   server issue :( We will resolve this as soon as possible! Please try
   in again in a moment”
}
```

Error - User files already exist (status code 400):

```
{
  "message": "failed to register",
   "error": "user dir"
}
```

Error - User board files already exist (status code 400):

```
{
  "message": "failed to register",
   "error": "user boards"
}
```

End Point: searchUsers.php

This endpoint is used to search for users based on their display name.

Request

- URL: prefix/search.php?search=<input>.
- Method: GET
- Body parameter:

- search (string, required): Replace <input> with the search query to find matching users

Response

Success - 5 found user matches (status code 200):

```
[
  {
    "displayName": "John Doe",
    "id": "123",
    "profilePicture": "https://example.com/profile_picture.jpg"
  },

  ...
]
```

```
displayName (string): The display name of the user.
id (string): The unique identifier of the user.
profilePicture (string): The URL of the user's profile picture.
```

Success - No users found (status code 200):

```
{"users": "No users found"}
```

Error - Invalid HTTP method (status code 405):

```
{"message": "Invalid method"}
```

Error - Missing search input (status code 400):

```
{"message": "Missing search input"}
```

End Point: updateUserProfile.php

This endpoint is used to update the user's profile information, including the display name and profile picture.

(!) Important: when updating profile picture, the body must be sent as FromData and the only file types allowed are “jpg”, “jpeg” & “png” and the uploaded file’s name has to be specified as “imageInput”.

Request - Change profile picture

- URL: prefix/updateUserProfile.php
- Method: POST
- Headers: "Content-Type: application/json"
- Body:
    - `Form-data (form data format)`
        - `userId (string): The unique identifier of the user.`
        - `imageInput (file), multipart: The new profile picture image file. The maximum file size is 2MB.`

Request - Change username

- URL: prefix/updateUserProfile.php
- Method: PATCH
- Headers: "Content-Type: application/json"
- Body:
    - `userId (string): The unique identifier of the user.`
    - `newDisplayName (string):`

Response

Success - Profile Picture Updated (status code 200):

```
{"message": "Profile updated"}
```

Success - Display Name Updated (status code 200):

```
{"message": "Profile updated"}
```

Error - Invalid HTTP Method (status code 405):

```
{"message": "Invalid method"}
```

Error - Missing user id when changing display name (status code 405):

```
{"message": "Id needed"}
```

Error - Missing new display name input (status code 400):

```
{"message": "Missing new display name input"}
```

Error - Image size too large (status code 400):

```
{ "message": "Image size too large"}
```

Error - Display name already in use (status code 400):

```
{ "message": "Display name already in use!"}
```

Error - File type not allowed (status code 400):

```
{ "message": "File type not allowed!" }
```

End Point: tokenAccess.php

This endpoint is used to access the token for the Spotify's external API. The request returns a necessary token needed to search via Spotify's music data.

(!)Important: The spotify API takes an input and returns the desired data (For more specific information look up spotifys's Api or follow the link provided. https://developer.spotify.com/documentation/web-api). Note that when sending a request the Spotify API the token is used in the headers key as following

Headers: { "Authorization": "Bearer " + token}

Request - Change profile picture

- URL: prefix/spotifyApi/tokenAccess.php
- Method: GET