

Social App API Documentation

This document provides an overview of the REST API endpoints for the **Social App**, built using **FastAPI** and **MySQL**.

Base URL

```
http://localhost:8000/
```

Authentication

Currently, no token-based authentication is implemented. All endpoints rely on passing user identifiers (`user_id`, `email`, etc.) as query parameters.

Endpoints

User Management

Sign Up

POST `/sign-up` - **Parameters:** `name`, `email`, `password` - **Response:**

```
{
  "id": 1
}
```

Sign In

GET `/sign-in` - **Parameters:** `email`, `password` - **Response:** User ID if authenticated, `404` if not found

Get User Data

GET `/get-user_data` - **Parameters:** `user_id` - **Response:** JSON object containing user data

Delete User

DELETE `/delete-user` - **Parameters:** `user_id` - **Response:**

```
"user with id = X deleted successfully"
```

Notifications

Send Notification

POST /send-notification - **Parameters:** receiver_id, content, friend_request_id (optional) - **Response:** { "id": <notification_id> }

Delete Notification

DELETE /delete-notification - **Parameters:** notification_id

Mark Notification as Read

POST /mark-notification-as-read - **Parameters:** notification_id

Get Notifications

GET /get-notifications - **Parameters:** user_id

Friends

Send Friend Request

POST /send-friend-request - **Parameters:** sender_id, receiver_id

Add Friend

POST /add-friend - **Parameters:** sender_id, receiver_id

Delete Friend

DELETE /delete-friend - **Parameters:** sender_id, receiver_id

Show Friends

GET /show-friends - **Parameters:** user_id - **Response:** List of friend IDs

Followers

Follow User

POST /follow - Parameters: sender_id, receiver_id

Unfollow User

DELETE /unfollow - Parameters: sender_id, receiver_id

Get Followers

GET /get-followers - Parameters: user_id

Get Followings

GET /get-followings - Parameters: user_id

Posts

Create Post

POST /create-post - Parameters: user_id, content

Delete Post

DELETE /delete-post - Parameters: post_id

Get All Posts

GET /get-all-posts

Get User Posts

GET /get-user-posts - Parameters: user_id

Share Post

POST /share-post - Parameters: user_id, post_id

Delete Post Share

DELETE /delete-post-share - Parameters: user_id, post_id

Like Post

POST /like-post - **Parameters:** user_id, post_id

Delete Post Like

DELETE /delete-post-like - **Parameters:** user_id, post_id

Comments

Create Comment

POST /create-comment - **Parameters:** user_id, post_id, content

Delete Comment

DELETE /delete-comment - **Parameters:** comment_id, user_id, post_id

Get Post Comments

GET /get-post-comments - **Parameters:** post_id

Like Comment

POST /like-comment - **Parameters:** user_id, comment_id, comment_user_id, post_id

Delete Comment Like

DELETE /delete-comment-like - **Parameters:** user_id, comment_id, comment_user_id, post_id

AI Chat

Send Prompt

POST /send-prompt - **Parameters:** user_id, content - **Response:**

```
{
  "response": "AI generated reply"
}
```

Delete Message

DELETE `/delete-message` - **Parameters:** `user_id`, `message_id`

Error Handling

- **400:** Bad Request (e.g., record not found, already exists)
 - **404:** Not Found (e.g., user not found)
 - **500:** Internal Server Error (e.g., database errors)
-

Notes

- All database connections use MySQL (`mysql.connector`).
 - IDs are auto-generated via `GetNewID` function.
 - Some queries use raw SQL (⚠ potential SQL injection risk).
-