

# Hướng dẫn Sử dụng API (UTH Social App)

Tài liệu này mô tả các API endpoint cho dự án mạng xã hội.

**Base URL:** `http://localhost:5000/api` (Giả định, thay bằng URL backend của bạn)

## 1. Xác thực (Authentication)

Tất cả các request yêu cầu xác thực (Auth Required) phải đính kèm Authorization header.

- **Header:** Authorization
- **Value:** Bearer <ACCESS\_TOKEN>

### 1.1. Đăng ký

- **Endpoint:** POST /auth/signup
- **Mô tả:** Tạo tài khoản mới.
- **Body (JSON):**

```
{
  "username": "ten nguoi dung",
  "email": "user@example.com",
  "password": "matkhau123"
}
```

- **Response (201 Created):** { "message": "Đăng ký thành công!" }

### 1.2. Đăng nhập

- **Endpoint:** POST /auth/signin
- **Mô tả:** Đăng nhập và nhận Access Token.
- **Body (JSON):**

```
{
  "email": "user@example.com",
  "password": "matkhau123"
}
```

- **Response (200 OK):** { "accessToken": "..." } (và có thể cả thông tin user)

### 1.3. Lấy thông tin người dùng (Tôi)

- **Endpoint:** GET /users/me
- **Mô tả:** Lấy thông tin chi tiết của người dùng đang đăng nhập (dựa trên token).
- **Auth:** (Auth Required)
- **Response (200 OK):** { "user": UserObject }

#### 1.4. Đăng xuất

- **Endpoint:** POST /auth/signout
- **Mô tả:** (Thường là để xóa httpOnly refresh token cookie ở backend)
- **Auth:** (Auth Required)
- **Response (200 OK):** { "message": "Đăng xuất thành công" }

#### 1.5. Làm mới Token

- **Endpoint:** POST /auth/refresh
- **Mô tả:** Lấy Access Token mới (thường dùng httpOnly cookie).
- **Auth:** (Credentials Required)
- **Response (200 OK):** { "accessToken": "... " }

### 2. Upload File

#### 2.1. Upload Ảnh (Cho Bài Đăng)

- **Endpoint:** POST /api/upload/images
- **Mô tả:** Upload một hoặc nhiều ảnh (tối đa 3). Dùng API này *trước khi* tạo/cập nhật post.
- **Auth:** (Auth Required)
- **Body (form-data):**
  - **Key:** images (Type: File) - (Chọn nhiều file ảnh)
- **Response (200 OK):**

```
{
  "urls": [
    "[https://res.cloudinary.com/](https://res.cloudinary.com/)...",
    "[https://res.cloudinary.com/](https://res.cloudinary.com/)..."
  ]
}
```

#### 2.2. Upload Tài liệu (Cho Bài Đăng)

- **Endpoint:** POST /api/documents/upload
- **Mô tả:** Upload một file tài liệu (PDF, DOCX, PPTX...). Dùng API này *trước khi* tạo/cập nhật post.
- **Auth:** (Auth Required)
- **Body (form-data):**
  - **Key:** document (Type: File) - (Chọn 1 file tài liệu)
- **Response (201 Created):** { "document": DocumentObject } (Trả về object Document mới từ DB, frontend lấy \_id từ đây).

### 3. Bài đăng (Posts)

#### 3.1. Tạo Bài Đăng Mới

- **Endpoint:** POST /api/posts/create
- **Mô tả:** Tạo một bài đăng mới. Yêu cầu đã upload ảnh/tài liệu (nếu có) và lấy URL/ID từ API Upload.
- **Auth:** (Auth Required)
- **Body (JSON):**

```
{
  "text": "Nội dung bài viết",
  "privacy": "public", // 'public', 'friends', 'private'
  "mediaUrls": [ // (Tùy chọn) Mảng URL từ API Upload Ảnh
    "[https://res.cloudinary.com/](https://res.cloudinary.com/)..."
  ],
  "docId": "68f..." // (Tùy chọn) _id từ API Upload Tài liệu
}
```

- **Response (201 Created):** { PostObject } (Bài post mới đã được populate author).

### 3.2. Cập nhật Bài Đăng

- **Endpoint:** PUT /api/posts/:postId (Lưu ý: Backend [cite: post.controller.js] đang dùng :id , cần thống nhất)
- **Mô tả:** Cập nhật text , privacy , mediaUrls , docId . Backend sẽ tự động xóa file cũ trên Cloudinary nếu mediaUrls hoặc docId thay đổi.
- **Auth:** (Auth Required)
- **Body (JSON):** Gửi toàn bộ trạng thái cuối cùng của các trường.

```
{
  "text": "Nội dung đã chỉnh sửa",
  "privacy": "friends",
  "mediaUrls": [ // Mảng URL cuối cùng (gồm ảnh cũ + ảnh mới)
    "[https://url-anh-cu.com/](https://url-anh-cu.com/)...",
    "[https://url-anh-moi-vua-upload.com/](https://url-anh-moi-vua-upload.com/)..."
  ],
  "docId": null // Gửi null nếu muốn gỡ tài liệu
}
```

- **Response (200 OK):** { PostObject } (Bài post đã cập nhật).

### 3.3. Xóa Bài Đăng

- **Endpoint:** DELETE /api/posts/:postId
- **Mô tả:** Xóa một bài đăng (chỉ chủ sở hữu).
- **Auth:** (Auth Required)
- **Response (200 OK):** { "message": "Xóa bài đăng thành công." }

### 3.4. Lấy Post Trang Chủ (Home Feed)

- **Endpoint:** GET /api/posts/home

- **Mô tả:** Lấy danh sách post cho trang chủ.
- **Auth:** (Auth Optional) - (Gửi token nếu đã đăng nhập để lấy myReactionType ).
- **Query Params:**
  - page (number): Số trang (ví dụ: 0, 1, 2...).
  - limit (number): Số lượng post mỗi trang (ví dụ: 10).
  - feed (string): Loại feed mong muốn.
    - ?feed=public (Mặc định): Lấy các bài public (cho khách & tab "Khám phá").
    - ?feed=following : (Chỉ khi đăng nhập) Lấy bài của người mình follow.
- **Response (200 OK):** [ PostObject, PostObject, ... ] (Mảng các bài post).

### 3.5. Lấy Post Trang Profile

- **Endpoint:** GET /api/posts/profile/:username
- **Mô tả:** Lấy danh sách post của một user theo username .
- **Auth:** (Auth Optional) - (Gửi token để xử lý privacy friends và myReactionType ).
- **Query Params:** ?page=0&limit=10
- **Response (200 OK):** [ PostObject, PostObject, ... ] (Mảng các bài post).

### 3.6. Reaction (Like/Dislike) Bài Đăng

- **Endpoint:** POST /api/posts/:postId/react
- **Mô tả:** Like, dislike, hoặc bỏ reaction một bài đăng.
- **Auth:** (Auth Required)
- **Body (JSON):**

```
{
  "type": "like" // hoặc "dislike"
}
```

- **Response (200 OK):** { "newReactionType": "like" } (hoặc dislike , null ).

## 4. Người dùng & Follow

### 4.1. Lấy Thông tin Profile User

- **Endpoint:** GET /api/users/by-username/:username
- **Mô tả:** Lấy thông tin công khai của user cho trang profile (Header).
- **Auth:** (Auth Optional) - (Gửi token để lấy isOwner , isFollowing ).
- **Response (200 OK):** { "userRes": UserObjectWithRelations } (Bao gồm isOwner , isFollowing ).

### 4.2. Follow

- **Endpoint:** POST /api/users/:userId/follow

- **Mô tả:** Theo dõi một người dùng bằng `userId` của họ.
- **Auth:** (Auth Required)
- **Response (200 OK):** { "message": "Theo dõi thành công." }

#### 4.3. Unfollow

- **Endpoint:** DELETE `/api/users/:userId/follow`
- **Mô tả:** Bỏ theo dõi một người dùng bằng `userId` của họ.
- **Auth:** (Auth Required)
- **Response (200 OK):** { "message": "Bỏ theo dõi thành công." }

### 5. Bình luận (Comments) & Trả lời (Replies)

(Sử dụng kiến trúc "Tách riêng" - Cách 2)

#### 5.1. Tạo Comment (Gốc hoặc Reply)

- **Endpoint:** POST `/api/comments`
- **Mô tả:** Tạo một bình luận mới.
- **Auth:** (Auth Required)
- **Body (JSON - Comment Gốc):**

```
{
  "text": "Bình luận cấp 1",
  "postId": "ID_CUA_BAI_DANG"
}
```

- **Body (JSON - Reply):**

```
{
  "text": "Trả lời bình luận",
  "parentId": "ID_CUA_COMMENT_CHA"
}
```

- **Response (201 Created):** { "comment": CommentObject } (Comment mới đã populate author).

#### 5.2. Lấy Danh sách Comments (Gốc hoặc Replies)

- **Endpoint:** GET `/api/comments`
- **Mô tả:** Lấy danh sách comment.
- **Auth:** (Auth Optional) - (Gửi token để lấy `myReactionType` cho comment).
- **Query Params (Chọn 1):**
  - `?postId=:postId&page=0&limit=10` (Lấy comment gốc, mới nhất trước)
  - `?parentId=:commentId&page=0&limit=10` (Lấy replies, cũ nhất trước)
- **Response (200 OK):** [ CommentObject, CommentObject, ... ]

### 5.3. Xóa Comment

- **Endpoint:** DELETE /api/comments/:commentId
- **Mô tả:** Xóa một comment (hoặc reply). Nếu xóa comment gốc, tất cả replies con cũng bị xóa.
- **Auth:** (Auth Required)
- **Response (200 OK):** { "message": "Xóa bình luận thành công." }

### 5.4. React (Like) Comment

- **Endpoint:** POST /api/comments/:commentId/react
- **Mô tả:** Like hoặc bỏ like một comment (hoặc reply).
- **Auth:** (Auth Required)
- **Body (JSON):**

```
{
  "type": "like"
}
```
- **Response (200 OK):** { "newReactionType": "like" } (hoặc null ).