

WEBHUB - API DOC

1. Mẫu doc cho 1 endpoint

2. Endpoint

<METHOD> <endpoint>

3. Mô tả

Nhiệm vụ của endpoint, đầu vào và đầu ra của endpoint (tóm tắt)

4. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json hoặc multipart/form-data	Có	Dữ liệu dạng json hoặc Gửi file hoặc dữ liệu dạng biểu mẫu
Authorization	Bearer <access_token>	Có / Không	Token xác thực người dùng

5. Request body

```
{  
    <args[1]> : <type>,  
    <args[2]>:<type>,  
    ....  
}
```

6. Response

5.1 Success

```
{  
    <args[1]> : <type>,  
    <args[2]>:<type>,  
    ....  
}
```

5.2 Failure - <Error code>

```
{  
    "error": "<Error message>"  
}
```

Tham khảo: [Bảng mã lỗi thường gặp](#)

7. Yêu cầu quyền truy cập

VD: ROLE >= MEMBER, ROLE = ADMIN ROLE>=GUEST

II. Các module công khai

4II.1. Authentication & Authorization (Phúc)

1.1 Đăng nhập

POST /api/v1/auth/login

Mô tả: Đăng nhập người dùng

Header: content-type : application/json

Request body

```
{
  "email": String,
  "password": String
}
```

Response Success

```
{
  "accessToken": String,
  "refreshToken": String
}
```

Response Failure

- **Sai tên đăng nhập hoặc mật khẩu (403 Forbidden)**

```
{
  "error": "Sai tên đăng nhập hoặc mật khẩu"
}
```
- **Tài khoản không tồn tại (401)**

```
{
  "error": "Tài khoản không tồn tại"
}
```
- **Tài khoản bị khóa (403)**

```
{
  "error": "Tài khoản hiện bị khóa"
}
```

1.2 Đăng xuất

POST /api/v1/auth/logout

Mô tả: Đăng xuất người dùng

Header: Authorization: Bearer

Request body: None

Response Success

```
{
  "message": "Logout successful"
}
```

}

1.3 Quên mật khẩu

POST /api/v1/auth/forgot-password

Mô tả: Nhập email quên mật khẩu

Header: content-type = json

Request body:

```
{
  "email" : String
}
```

Response Success

```
{
  "message": "Link đặt lại mật khẩu đã gửi vào email của bạn"
}
```

Response Failure (403)

- Email không tồn tại
- ```
{
 "error": "Email không tồn tại"
}
```

#### **GET /api/v1/auth/reset-password?token={token} (redirect from email's token)**

**Mô tả:** Chuyển sang trang đổi mật khẩu

#### **POST /api/v1/auth/reset-password**

**Mô tả:** Đặt lại mật khẩu mới

**Header:** content-type = json

**Request body**

```
{
 "token":String,
 "password":String
}
```

**Response Success**

```
{
 "message": "Reset password successfully"
}
```

**Response Failure**

- Token sai / hết hạn (401)
- ```
{ "error": "Yêu cầu hết hạn hoặc không hợp lệ" }
```

1.4 Làm mới token

POST /api/v1/auth/refresh

Header: Authorization token (access token)

II.2. User management & User profile management (Trần kiệt)

II.2.1 User management

2.1 Xem danh sách người dùng

1. URL

GET api/v1/users : lấy tất cả danh sách users.

{

GET api/v1/users?name=full_name : tìm user theo tên.

GET api/v1/users?email=email : tìm user theo email.

GET api/v1/users?role=role ('MEMBER', 'ADMIN','ROOT') : lọc danh sách users theo role

GET api/v1/users?status=is_active (true,false) : lọc danh sách users theo status

}

2. Mô tả

đầu vào: các tham số query.

đầu ra: danh sách user sau khi tìm,lọc.

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Accept	application/json	có	nhận dữ liệu dưới dạng json
Authorization	Bearer<access_token>	có	token xác thực Admin,root

4. Response

4.1 Success

```
[
  {
    "id":integer,
    "avatar_url":string,
    "full_name":string,
    "email":string,
    "role":string,
    "is_active":boolean
  },
  ...
]
```

4.2 Failure - 403

```
{
  "error": "Không có quyền truy cập danh sách người dùng."
}
```

Failure - 500

```
{
```

```

        "error": "Xảy ra lỗi không xác định ở server."
    }
Failure - 400
    {
        "error": "Sai tham số"
    }

```

2.2 Tạo tài khoản người dùng mới

1. URL

POST /api/v1/users

2. Mô tả

đầu vào: json chứa các dữ liệu cần.

đầu ra: json chứa thông tin cần thiết

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json	có	gửi dữ liệu dạng json
Authorization	Bearer<access_token>	có	token xác thực Admin,root

4. Request body

```

{
    "email":string,
    "full_name":string,
    "password_hash":string,
    "role":string,
    "is_active":boolean
}

```

5. Response

5.1 Success

```

{
    "id":interger,
    "full_name":string,
    "email":string,
    "role":string,
    "is_active":boolean,
    "avatar_url":string
}

```

5.2 Failure - 400

```

{
    "error":"Email này đã được sử dụng. Vui lòng chọn email khác."
}

```

2.3 Xem và Chỉnh sửa thông tin người dùng và vô hiệu hóa tài khoản

1. URL

PATCH /api/v1/users/{id} : chỉnh sửa các thông tin của user trong bảng

PATCH /api/v1/users/{id}/status : vô hiệu hóa tài khoản

PATCH /api/v1/users/{id}/file : đổi avatar theo file

2. Mô tả

đầu vào: id của user cần sửa và nội dung chỉnh sửa trong js, id của user cần resetpass.

đầu ra: thông tin user sau chỉnh sửa, thông báo resetpass.

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json,multipart/form-data	có	gửi dữ liệu dạng json hoặc file
Authorization	Bearer<access_token>	có	token xác thực Admin,root

4. Request body

PATCH /api/v1/users/{id}

application/json

```
{
  "avatar_url":string,
  "full_name":string,
  "role":string,
}
```

PATCH /api/v1/users/{id}/multipart

:

requestpast avatar: file (binary)

requestparam full_name:string

requestparam role:string

PATCH /api/v1/users/{id}/status

```
{
  "is_active": boolean
}
```

5. Response

5.1 Success

```
{
  "id":interger,
  "avatar_url":string,
```

```

        "full_name":string,
        "email":string, (read-only)
        "role":string,
        "is_active":boolean
    }

```

5.2 Failure - <error_code>

```

{
    "error": "... "
}

```

2.5 Xóa tài khoản người dùng (chỉ Root)

1. URL

DELETE /api/v1/users/{id}

2. Mô tả

Đầu vào:

Email xác nhận xóa (trong JSON request body)

Đầu ra:

Thông báo xóa thành công, kèm thông tin các nội dung đã được chuyển quyền cho Admin Hệ thống

Hoặc trả về lỗi nếu email xác nhận sai, user không tồn tại, hoặc không đủ quyền

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json	có	gửi dữ liệu dạng json
Authorization	Bearer<access_token>	có	token xác thực root

3. Request body

```

{
    "email": string
}

```

4. Response

5.1 Success

```

{
    "message": "Xóa người dùng thành công. Các nội dung đã được chuyển quyền sở hữu cho Admin Hệ thống."
    "id":interger
}

```

```
}
```

5.2 Failure - 403

```
{
```

```
  "error": "Bạn không có quyền thực hiện thao tác này."
```

```
}
```

Failure - 400

```
{
```

```
  "error": "Email xác nhận không đúng."
```

```
}
```

Failure - 404

```
{
```

```
  "error": "Không tìm thấy người dùng với ID này."
```

```
}
```

II.2.2 User profile management

2.1 Xem thông tin hồ sơ cá nhân

1. URL

GET api/v1/users/me

2. Mô tả

Đầu vào: token người dùng

Đầu ra: thông tin user (full_name, email, role, avatar_url)

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Accept	application/json	có	nhận dữ liệu dạng json
Authorization	Bearer<access_token>	có	Authenticated User (Member, Admin, Root).

4. Response

4.1 Success

```
{
```

```
  "id":integer,
```

```
  "avatar_url":string,
```

```
  "full_name":string,
```

```
  "email":string,
```

```
  "role":string,
```



```
        "is_active":boolean
    }
```

4.2 Failure - 500

```
{
    "error": "Không thể tải thông tin hồ sơ. Vui lòng thử lại."
}
```

2.2 Cập nhật thông tin cá nhân cơ bản và ảnh đại diện avatar

5. URL

PATCH /api/v1/users/me : đổi tên

PATCH /api/v1/users/me/avatar : đổi avt

6. Mô tả

Đầu vào: token + tên mới hoặc file ảnh

Đầu ra: trả thông tin user mới hoặc avatar mới

7. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json/multipart/form-data	có	gửi dữ liệu dạng json hoặc file
Authorization	Bearer<access_token>	có	Authenticated User (Member, Admin, Root).

8. Request body

thay đổi tên

```
{
    "full_name": string
}
```

9. Response

5.1 Success

```
{
    "id": Longerger,
    "full_name": string,
    "email": string,
    "role": string,
```

```
        "avatar_url": string,  
        "is_active":boolean  
    }
```

5.2 Failure - 400

```
{  
    "error": "Họ và tên không được để trống."  
}
```

Failure - 400

```
{  
    "error": "File ảnh không hợp lệ."  
}
```

Failure - 500

```
{  
    "error": "Có lỗi khi lưu ảnh."  
}
```

2.3 Đổi mật khẩu

1. URL

PATCH /api/v1/users/password

2. Mô tả

Đầu vào: token + mật khẩu cũ+mật khẩu mới

Đầu ra: thông báo thành công

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json	có	gửi dữ liệu dạng json
Authorization	Bearer<access_token>	có	Authenticated User (Member, Admin, Root).

4. Request body

```
{  
    "current_password":string,  
    "new_password":string
```

```
}
```

5. Response

5.1 Success

```
{  
    "error": "Đổi mật khẩu thành công",  
}
```

5.2 Failure - 403

```
{  
    "error": "Mật khẩu hiện tại không chính xác."  
}
```

Failure - 400

```
{  
    "error": "Mật khẩu mới không đủ mạnh."  
}
```

TÀI LIỆU API CHO BLOG

Mục lục:

1. Tổng quan về Phân quyền
 2. API cho Guest (Khách)
 3. API cho User (Thành viên đã đăng nhập)
 4. API cho Admin (Quản trị viên)
1. TỔNG QUAN VỀ PHÂN QUYỀN
 - Guest (Khách): Người dùng chưa xác thực. Có quyền truy cập công khai vào các API để đọc bài viết.
 - User (Thành viên): Người dùng đã đăng nhập nhưng không phải Admin. Hiện tại, vai trò này có các quyền tương tự như Guest. Không có API đặc thù nào cho User trong module blog.
 - Admin (Quản trị viên): Người dùng có quyền cao nhất. Yêu cầu xác thực và có vai trò 'ADMIN'. Có toàn quyền quản lý các bài viết (tạo, sửa, xóa, thay đổi trạng thái).

Lưu ý: Phân quyền dưới đây được suy luận từ cấu trúc của `BlogController.java`. Cấu hình bảo mật hiện tại (`SecurityConfiguration.java`) có thể chưa thực thi đầy đủ các quy tắc này.

2. API CHO GUEST (KHÁCH)

Các API này không yêu cầu xác thực.

2.1. Lấy tất cả bài viết đã xuất bản

- Mô tả: Lấy danh sách phân trang các bài viết đã được xuất bản.
- Quyền: Guest
- Method: GET
- Endpoint: `/api/blog-posts`
- Tham số (Query Params):
 - `page` (number, tùy chọn): Số trang.
 - `size` (number, tùy chọn): Số lượng mục trên mỗi trang.
 - `search` (string, tùy chọn): Từ khóa tìm kiếm trong tiêu đề và nội dung.
 - `tag` (string, tùy chọn): Lọc bài viết theo thẻ.
- Phản hồi thành công (200 OK):
 - Cấu trúc `GetBlogAllResponse`:

```
{ "data": [ { "id": 1, "title": "Tiêu đề bài viết", "slug": "tiu-de-bai-viet", "thumbnailUrl": "http://example.com/image.jpg", "authorName": "Tên Tác Giả", "publishedAt": "2023-10-27T10:00:00Z", "shortContent": "Nội dung tóm tắt..." } ], "pagination": { "totalItems": 100, "totalPages": 10, "currentPage": 1, "nextPage": 2, "prevPage": null } }
```
- Ví dụ: `GET /api/blog-posts?page=0&size=10&tag=java`

2.2. Lấy các bài viết nổi bật

- Mô tả: Lấy danh sách các bài viết được ghim/nổi bật.
- Quyền: Guest
- Method: GET
- Endpoint: `/api/blog-posts/featured`
- Phản hồi thành công (200 OK):

- Cấu trúc `List<FeaturedBlogResponse>`: [{ "id": 2, "title": "Bài viết nổi bật", "slug": "bai-viet-noi-bat", "thumbnailUrl": "<http://example.com/featured.jpg>", "authorName": "Tên Tác Giả", "publishedAt": "2023-10-26T15:30:00Z", "shortContent": "Nội dung tóm tắt của bài viết nổi bật..." }]
- Ví dụ: GET `/api/blog-posts/featured`

2.3. Lấy chi tiết bài viết theo Slug

- Mô tả: Lấy toàn bộ nội dung của một bài viết dựa vào `slug`.
- Quyền: Guest
- Method: GET
- Endpoint: `/api/blog-posts/{slug}`
- Tham số (Path Variable):
 - `slug` (string, bắt buộc): Slug của bài viết.
- Phản hồi thành công (200 OK):
 - Cấu trúc `GetBlogBySlugResponse`: { "id": 1, "title": "Tiêu đề bài viết", "slug": "tieu-de-bai-viet", "content": "<p>Nội dung đầy đủ của bài viết.</p>", "thumbnailUrl": "<http://example.com/image.jpg>", "authorId": 10, "authorName": "Tên Tác Giả", "status": "PUBLISHED", "isPinned": false, "publishedAt": "2023-10-27T10:00:00Z", "createdAt": "2023-10-27T09:00:00Z", "updatedAt": "2023-10-27T11:00:00Z", "tags": [{ "id": 1, "name": "Java" }, { "id": 2, "name": "Spring" }] }
- Phản hồi lỗi (404 Not Found): Nếu không tìm thấy bài viết.
- Ví dụ: GET `/api/blog-posts/huong-dan-lap-trinh-spring-boot`

2.4. Lấy các bài viết gần đây

- Mô tả: Lấy danh sách các bài viết mới nhất.
- Quyền: Guest
- Method: GET
- Endpoint: `/api/blog-posts/recent`
- Tham số (Query Params):
 - `limit` (number, tùy chọn): Số lượng bài viết muốn lấy. Mặc định là 5.

- Phản hồi thành công (200 OK):
 - Cấu trúc `List<BlogSummaryResponse>`: (Tương tự mục 2.1, phần `data`)
- Ví dụ: GET `/api/blog-posts/recent?limit=3`

3. API CHO USER (THÀNH VIÊN)

Hiện tại, không có API nào được thiết kế riêng cho vai trò "User". Người dùng đã đăng nhập sẽ có các quyền tương tự như "Guest" đối với module Blog.

4. API CHO ADMIN (QUẢN TRỊ VIÊN)

Các API này yêu cầu người dùng phải được xác thực và có vai trò là Admin.

4.1. Lấy tất cả bài viết (Admin)

- Mô tả: Lấy danh sách phân trang tất cả bài viết trong hệ thống, không phân biệt trạng thái.
- Quyền: Admin
- Method: GET
- Endpoint: `/api/admin/blog-posts`
- Tham số (Query Params):
 - `pageable`: Các tham số phân trang của Spring (page, size, sort).
 - `search` (string, tùy chọn): Từ khóa tìm kiếm.
 - `status` (enum, tùy chọn): Lọc theo trạng thái (`DRAFT`, `PUBLISHED`, `ARCHIVED`).
 - `author_id` (long, tùy chọn): Lọc theo ID tác giả.
- Phản hồi thành công (200 OK):
 - Cấu trúc `AdminGetAllBlogWithPaginationResponse`: `{ "data": [{ "id": 1, "title": "Tiêu đề bài viết", "slug": "tieu-de-bai-viet", "status": "PUBLISHED", "authorName": "Tên Tác Giả", "isPinned": false, "createdAt": "2023-10-27T09:00:00Z", "updatedAt": "2023-10-27T11:00:00Z" }], "pagination": { "totalItems": 100, "totalPages": 10, "currentPage": 1, "nextPage": 2, "prevPage": null } }`
- Ví dụ: GET `/api/admin/blog-posts?status=DRAFT`

4.2. Tạo bài viết mới

- Mô tả: Tạo một bài viết mới.
- Quyền: Admin
- Method: POST
- Endpoint: /api/admin/blog-posts
- Tham số (Query Params):
 - `user_id` (long, bắt buộc): ID của người dùng (admin) tạo bài viết.
- Nội dung yêu cầu (Request Body):
 - Cấu trúc `CreateBlogRequest`: { "title": "Tiêu đề bài viết mới", "content": "<p>Nội dung chi tiết.</p>", "thumbnailUrl": "[http://example.com/new_thumb.jpg](\"http://example.com/new_thumb.jpg\")", "status": "DRAFT", "isPinned": false, "tagIds": [1, 5] }
- Phản hồi thành công (201 Created):
 - Cấu trúc `AdminCreateBlogResponse`: { "id": 101, "title": "Tiêu đề bài viết mới", "slug": "tiu-de-bai-viet-moi", "status": "DRAFT", "author_id": 1, "created_at": "2023-10-28T12:00:00Z" }
- Phản hồi lỗi (400 Bad Request): Nếu dữ liệu không hợp lệ.

4.3. Cập nhật bài viết

- Mô tả: Cập nhật thông tin một bài viết đã có.
- Quyền: Admin
- Method: PUT
- Endpoint: /api/admin/blog-posts/{id}
- Tham số (Path Variable):
 - `id` (long, bắt buộc): ID của bài viết cần cập nhật.
- Nội dung yêu cầu (Request Body):
 - Cấu trúc `UpdateBlogRequest`: (Tương tự `CreateBlogRequest`) { "title": "Tiêu đề đã cập nhật", "content": "<p>Nội dung đã cập nhật.</p>", "status": "PUBLISHED" }
- Phản hồi thành công (200 OK):
 - Cấu trúc `AdminUpdateBlogResponse`: { "id": 101, "title": "Tiêu đề đã cập nhật", "slug": "tiu-de-da-cap-nhat", "status": "PUBLISHED", "updatedAt": "2023-10-28T13:00:00Z" }

- Phản hồi lỗi (404 Not Found): Nếu không tìm thấy bài viết.

4.4. Xóa bài viết

- Mô tả: Xóa vĩnh viễn một bài viết.
- Quyền: Admin
- Method: DELETE
- Endpoint: `/api/admin/blog-posts/{id}`
- Tham số (Path Variable):
 - `id` (long, bắt buộc): ID của bài viết cần xóa.
- Phản hồi thành công (204 No Content): Xóa thành công.
- Phản hồi lỗi (404 Not Found): Nếu không tìm thấy bài viết.

4.5. Thay đổi trạng thái bài viết

- Mô tả: Thay đổi trạng thái của bài viết (Draft, Published, Archived).
- Quyền: Admin
- Method: PATCH
- Endpoint: `/api/admin/blog-posts/{id}/status`
- Tham số (Path Variable):
 - `id` (long, bắt buộc): ID của bài viết.
- Nội dung yêu cầu (Request Body):
 - Cấu trúc `ChangeStatusBlogRequest`: `{ "status": "ARCHIVED" }`
- Phản hồi thành công (200 OK):
 - Cấu trúc `AdminChangeStatusBlogResponse`: `{ "id": 101, "title": "Tiêu đề đã cập nhật", "status": "ARCHIVED", "updatedAt": "2023-10-28T14:00:00Z" }`

4.6. Ghim / Bỏ ghim bài viết

- Mô tả: Đánh dấu một bài viết là nổi bật (ghim) hoặc bỏ ghim.
- Quyền: Admin
- Method: PATCH
- Endpoint: `/api/admin/blog-posts/{id}/pin`
- Tham số (Path Variable):
 - `id` (long, bắt buộc): ID của bài viết.

- Nội dung yêu cầu (Request Body):
 - Cấu trúc `PinBlogRequest`: { "isPinned": true }
- Phản hồi thành công (200 OK):
 - Cấu trúc `PinBlogResponse`: { "id": 101, "title": "Tiêu đề đã cập nhật", "isPinned": true, "updatedAt": "2023-10-28T15:00:00Z" }

4.7. Lấy bài viết theo ID (Admin)

- Mô tả: Lấy thông tin chi tiết một bài viết theo ID, dùng cho trang chỉnh sửa.
- Quyền: Admin
- Method: GET
- Endpoint: `/api/admin/blog-posts/{id}`
- Phản hồi thành công (200 OK): Cấu trúc `GetBlogBySlugResponse` (Xem mục 2.3).

4.8. Lấy bài viết theo Slug (Admin)

- Mô tả: Lấy thông tin chi tiết một bài viết theo slug.
- Quyền: Admin
- Method: GET
- Endpoint: `/api/admin/blog-posts/slug/{slug}`
- Phản hồi thành công (200 OK): Cấu trúc `GetBlogBySlugResponse` (Xem mục 2.3).

4.9. Các API tiện ích (Admin)

- Đếm tổng số bài viết:
 - `GET /api/admin/blog-posts/count` -> Trả về Long
- Đếm bài viết theo trạng thái:
 - `GET /api/admin/blog-posts/count/status/{status}` -> Trả về Long
- Kiểm tra bài viết tồn tại:
 - `GET /api/admin/blog-posts/exists/{id}` -> Trả về Boolean

|| 4.2. Project management

|| 4.2.1. Tạo dự án mới

1. URL

POST /api/v1/projects

2. Mô tả

- Tạo dự án mới.
- quyền tạo: admin hoặc root.

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json	có	gửi dữ liệu dạng json
Authorization	Bearer<access_token>	có	Authenticated User (Admin, Root).

4. Request body

```
{
  "project":{
    "name": String,
    "type":String;(INTERNAL, COLLAB, RESEARCH)
    "is_featured": Boolean,
    "short_description": String,
    "full_description": String,
    "thumbnail_url": String,
    "status": "String",
    "is_public": boolean,
    "start_date": Date,
    "end_date":Date,
    "git_url": String,
    "production_url": String
  },
  "tags":[
    "tag_id": Long;
  ]
}
```

```
}
```

5. Response

- 200 Success:

```
{  
  "message": "Dự án đã được tạo thành công",  
  "project":  
    {  
      "id":int,  
      "name": String,  
      "is_featured": Boolean,  
      "thumbnail_url": String,  
      "status ": "String",  
      "is_public": boolean,  
      "start_date": Date,  
      "end_date":Date,  
      "creator_id": int,  
      "slug": String,  
      "created_at": Date  
    }  
}
```

- Failure:

```
{  
  "statusCode": 0,  
  "error": "string",  
  "message": {},  
  "data": {}  
}
```

|| 4.2.2 Cập nhật dự án

1. URL

PATCH /api/v1/projects/{id}

2. Mô tả

- Cập nhật thông tin và hình ảnh dự án

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Content-Type	application/json	Có	gửi dữ liệu dạng json
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

4. Request body

```
{
  "name": String,
  "type": String,
  "is_featured": Boolean,
  "short_description": String,
  "full_description": String,
  "status": String,
  "thumbnail_url": String,
  "is_public": boolean,
  "start_date": Date,
  "end_date": String,
  "git_url": String,
  "production_url": String,
  "tags":[
    "tag_id": Long;
  ]
}
```

1. Response

- **200 Success:**

```
{
  "message": "Cập nhật dự án thành công";
  "Project": [
```

```

        "id": Long,
        "name": String,
        "is_featured": Boolean,
        "full_description": String,
        "thumbnail_url": String,
        "status":String,
        "is_public": boolean,
        "start_date": String,
        "end_date": String
        "creator_id": int,
        "slug": String,
        "created_at": String
    ]
}

- 400 Failure:
{
    "message": "Thiếu thông tin dự án";
}

- 404 Failure:
{
    "error": "Không tìm thấy dự án hoặc không có quyền";
}

```

|| 4.2.3 Xóa dự án

1. URL

DELETE /api/v1/projects/{id}

2. Mô tả

- Xóa dự án.
- Quyền: admin hoặc root.

3. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

4. Request body

```
{  
    "confirm_delete":boolean;  
}
```

5. Response

- 204 Success:

```
{  
    "message": "Xóa dự án thành công";  
    "id": Long;  
}
```

- 400 Failure:

```
{  
    "message": "Dự án không tồn tại";  
}
```

- 404 Failure:

```
{  
    "error": "Không tìm thấy dự án hoặc không có quyền";  
}
```

|| 4.2.4 Lấy danh sách tags của project

1. **URL:** GET /api/v1/projects/{id}/tags

2. **Mô tả:** Lấy danh sách dự án của project

3. **Headers**

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

4. Response

- 201 OK

```
{  
    "projectId": Long,  
    "tags": [  
        {  
            "id": Long,  
            "name": String  
        }  
    ]  
}
```

// chưa có điều kiện kiểm tra id dự án tồn tại

```
- Failure:
{
  "statusCode": int,
  "error": String,
  "message": String,
  "data": {}
}
```

|| 4.2.5 Sửa type của dự án

1. URL

PUT /api/v1/projects/{id}/type

2. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

3. Request body

```
{
  "status": String;
}
```

4. Response

- **200 Success:**

```
{
  "message": "string",
  "project": {
    "id": Long,
    "name": "string",
    "slug": "string",
    "is_featured": true,
    "full_description": "string",
    "thumbnail_url": "string",
    "project_status": "string",
    "is_public": boolean,
    "start_date": "2025-07-29",
```

```

        "end_date": "2025-07-29",
        "creator_id": Long,
        "created_at": "2025-07-29"
    }
}

- Failure:
{
    "statusCode": int,
    "error": String,
    "message": String,
    "data": {}
}

```

|| 4.2.6 Sửa status của dự án

1. URL

PUT /api/v1/projects/{id}/status

2. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

3. Request body

```

{
    "is_feature": String;
}

```

4. Response

- **200 Success:**

```

{
    "message": "string",
    "project": {
        "id": Long,
        "name": "string",
        "slug": "string",
        "is_featured": true,
    }
}

```



```

    "full_description": "string",
    "thumbnail_url": "string",
    "project_status": "string",
    "is_public": boolean,
    "start_date": "2025-07-29",
    "end_date": "2025-07-29",
    "creator_id": Long,
    "created_at": "2025-07-29"
  }
}

```

- Failure:

```

{
  "statusCode": int,
  "error": String,
  "message": String,
  "data": {}
}

```

|| 4.2.6 Sửa trạng thái nổi bật của dự án

5. URL

PUT /api/v1/projects/{id}/isFeatured

6. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

7. Request body

```

{
  "type": String;
}

```

8. Response

- 200 Success:

```

{
  "message": "string",

```

```

"project": {
  "id": Long,
  "name": "string",
  "slug": "string",
  "is_featured": true,
  "full_description": "string",
  "thumbnail_url": "string",
  "project_status": "string",
  "is_public": boolean,
  "start_date": "2025-07-29",
  "end_date": "2025-07-29",
  "creator_id": Long,
  "created_at": "2025-07-29"
}
}

```

- Failure:

```

{
  "statusCode": int,
  "error": String,
  "message": String,
  "data": {}
}

```

|| 4.2.7 Lấy danh sách dự án

1. URL

GET /api/v1/projects

2. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

3. Paramater

```

{
  "page": int, (optional)
}

```

```
        "size": int (optional)
    }
}
```

4. Response

- 200 Success:

```
{
  "content": [
    {
      "id": Long,
      "name": "string",
      "type": "string",
      "status": "string",
      "thumbnail_url": "string",
      "is_public": boolean,
      "is_featured": boolean
    }
  ],
  "pageNumber": 0,
  "pageSize": 0,
  "totalElements": 0,
  "totalPages": 0,
  "last": true
}
```

- Failure:

```
{
  "statusCode": int,
  "error": String,
  "message": String,
  "data": {}
}
```

|| 4.2.8 Lấy danh sách dự án từ danh sách tags

1. URL

POST /api/v1/projects/tags

2. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

3. Paramaters

```
{  
    "page": int, (optional)  
    "size": int (optional)  
}
```

4. Requestbody

```
{  
    "tags": [  
        "id": Long  
    ]  
}
```

5. Response

- **200 Success:**

```
{  
    "content": [  
        {  
            "id": Long,  
            "name": "string",  
            "type": "string",  
            "status": "string",  
            "thumbnail_url": "string",  
            "is_public": boolean,  
            "is_featured": boolean  
        }  
    ],  
    "pageNumber": 0,  
    "pageSize": 0,  
    "totalElements": 0,  
    "totalPages": 0,  
    "last": true
```

```
}
```

- Failure:

```
{  
  "statusCode": int,  
  "error": String,  
  "message": String,  
  "data": {}  
}
```

|| 4.2.9 Lấy danh sách dự án theo bộ lọc

1. URL

GET /api/v1/projects

2. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

3. Paramaters

```
{  
  "page": int, (optional)  
  "size": int (optional)  
}
```

4. Requestbody

```
{  
  "name": "string",  
  "type": "string",  
  "status": "string",  
  "description": "string",  
  "tagIds": [  
    0  
  ],  
  "is_featured": true  
}
```

5. Response

- **200 Success:**

```
{
  "content": [
    {
      "id": Long,
      "name": "string",
      "type": "string",
      "status": "string",
      "thumbnail_url": "string",
      "is_public": boolean,
      "is_featured": boolean
    }
  ],
  "pageNumber": 0,
  "pageSize": 0,
  "totalElements": 0,
  "totalPages": 0,
  "last": true
}
```

- **Failure:**

```
{
  "statusCode": int,
  "error": String,
  "message": String,
  "data": {}
}
```

|| 4.2.10 Lấy danh sách dự án theo type

1. URL

POST /api/v1/projects/filter

2. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
-----	---------	----------	---------

Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).
---------------	----------------------	----	------------------------------------

3. Paramaters

```
{
  "type": String,
  "page": int, (optional)
  "size": int (optional)
}
```

4. Response

- 200 Success:

```
{
  "content": [
    {
      "id": Long,
      "name": "string",
      "type": "string",
      "status": "string",
      "thumbnail_url": "string",
      "is_public": boolean,
      "is_featured": boolean
    }
  ],
  "pageNumber": 0,
  "pageSize": 0,
  "totalElements": 0,
  "totalPages": 0,
  "last": true
}
```

- Failure:

```
{
  "statusCode": int,
```

```

    "error": String,
    "message": String,
    "data": {}
}

```

|| 4.3 Quản lý tags

|| 4.3.1 Thêm tags project

1. **URL:** POST /api/v1/projects/{id}/tags
2. **Mô tả:** gắn tag vào project
3. **Headers**

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

4. Parameters

```

{
    "id": Long
}

```

5. Request body

```

{
    "name": String
}

```

6. Response

- 201 OK

```

{
    "message": "string",
    "tag": {
        "id": Long,
        "name": "string"
    },
    "project_id": Long
}

```

- Failure:

```

{
    "statusCode": int,
    "error": String,
    "message": String,
    "data": {}
}

```


|| 4.3.1 Xóa tags project

7. **URL:** POST /api/v1/projects/{id}/tags

8. **Mô tả:** gắn tag vào project

9. Headers

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer<access_token>	Có	Authenticated User (Admin, Root).

10. Parameters

```
{
    "id": Long
}
```

11. Request body

```
{
    "name": String
}
```

12. Response

- 200 OK

```
{
    "message": "string",
    "tag": {
        "id": Long,
        "name": "string"
    },
    "project_id": Long
}
```

- Failure:

```
{
    "statusCode": int,
    "error": String,
    "message": String,
    "data": {}
}
```

II.5 Document management & Public document view (Anh Hưng)

1. General Rules

- **Resource:** documents, folders
- **URL Base:** /api/v1/
- **Authorization:** Bearer <access_token> (Required for all endpoints)
- **Error Format:** {"error": "error description"}

2. Folder Management Endpoints (/folders)

2.1. Lấy danh sách thư mục (UC-DM-08)

- **Endpoint:** GET /api/v1/folders
- **Mô tả:** Lấy danh sách cấu trúc thư mục của CLB. Có thể lọc theo thư mục cha.
- **Quyền:** ROLE >= MEMBER

Tên	Giá trị	Bắt buộc	Ghi chú
Authorization	Bearer;access_token;	Có	Token xác nhận

Tên	Kiểu	Bắt buộc	Mô tả
parent_id	integer	Không	ID của thư mục cha để lấy danh sách thư mục con. Nếu không có, lấy thư mục gốc.

- **Response (200 OK):**

```
{
  "data": [
    {
      "id": 1,
      "name": "Báo cáo Tháng 7",
```

```

    "owner_id": 5,
    "parent_folder_id": null,
    "created_at": "2025-07-16T10:00:00Z",
    "updated_at": "2025-07-16T10:00:00Z"
  },
  {
    "id": 2,
    "name": "Tài liệu Workshop AI",
    "owner_id": 5,
    "parent_folder_id": null,
    "created_at": "2025-07-15T11:00:00Z",
    "updated_at": "2025-07-15T11:00:00Z"
  }
]
}

```

2.2. Tạo thư mục mới (UC-DM-07)

- **Endpoint:** `POST /api/v1/folders`
- **Mô tả:** Tạo một thư mục mới.
- **Quyền:** `ROLE >= ADMIN`
- **Request Body (`application/json`):**

```

{
  "name": "Kế hoạch Quý 3",
  "parent_folder_id": 1
}

```

- **Response (201 Created):**

```

{
  "message": "Thư mục đã được tạo thành công!",
  "folder": {
    "id": 3,
    "name": "Kế hoạch Quý 3",
    "owner_id": 5,
    "parent_folder_id": 1,
    "created_at": "2025-07-16T12:00:00Z",
    "updated_at": "2025-07-16T12:00:00Z"
  }
}

```

```
}
```

- **Error Responses:**

- **400 Bad Request:** {"error": "Tên thư mục không được để trống."} hoặc {"error": "Tên thư mục đã tồn tại trong thư mục cha."}

2.3. Cập nhật thư mục (UC-DM-08)

- **Endpoint:** PATCH /api/v1/folders/{id}
- **Mô tả:** Đổi tên hoặc di chuyển một thư mục.
- **Quyền:** ROLE >= ADMIN
- **Request Body (application/json):**

```
{  
  "name": "Kế hoạch Quý 3 - Đã duyệt",  
  "parent_folder_id": null  
}
```

- **Response (200 OK):**

```
{  
  "message": "Cập nhật thư mục thành công!",  
  "folder": { ... } // Dữ liệu thư mục đã cập nhật  
}
```

- **Error Responses:**

- **404 Not Found:** {"error": "Không tìm thấy thư mục."}

2.4. Xóa thư mục (UC-DM-08)

- **Endpoint:** DELETE /api/v1/folders/{id}
- **Mô tả:** Xóa vĩnh viễn một thư mục và tất cả nội dung bên trong (thư mục con và tài liệu).
- **Quyền:** ROLE >= ADMIN
- **Response (204 No Content):** Không có nội dung phản hồi.
- **Error Responses:**
 - **404 Not Found:** {"error": "Không tìm thấy thư mục."}

3. Document Management Endpoints (/documents)

3.1. Lấy danh sách tài liệu (UC-IDV-01, 02, 05)

- **Endpoint:** GET /api/v1/documents

- **Mô tả:** Lấy danh sách các tài liệu của CLB mà người dùng có quyền xem. Mặc định chỉ lấy các tài liệu có `is_public=TRUE`. Có thể lọc theo từ khóa, tag, và thư mục.
- **Quyền:** `ROLE >= MEMBER`

Tên	Kiểu	Bắt buộc	Mô tả
<code>q</code>	<code>string</code>	Không	Từ khóa tìm kiếm (so sánh với tiêu đề và mô tả).
<code>tags</code>	<code>string</code>	Không	Chuỗi các ID của tag, cách nhau bởi dấu phẩy (vd:1,5,12).
<code>folder_id</code>	<code>integer</code>	Không	Lọc tài liệu trong một thư mục cụ thể. Nếu <code>folder_id=0</code> hoặc không có, lấy tài liệu ở thư mục gốc.
<code>page</code>	<code>integer</code>	Không	Số trang muốn lấy (default = 1).
<code>limit</code>	<code>integer</code>	Không	Số tài liệu/trang (default = 10).

- **Response (200 OK):**

```
{
  "data": [
    {
      "id": 101,
      "title": "Báo cáo tổng kết tháng 6",
      "description": "Tài liệu báo cáo hoạt động...",
      "source_type": "INTERNAL_UPLOAD",
      "mime_type": "application/pdf",
      "created_at": "2025-07-01T09:00:00Z",
      "author": {
        "id": 5,
        "full_name": "Nguyễn Văn A"
      },
      "folder_id": 1
    }
  ],
}
```

```

"meta": {
  "current_page": 1,
  "per_page": 10, // Giả định giới hạn mặc định là 10
  "total_items": 48,
  "total_pages": 5
}
}

```

3.2. Xem chi tiết và tải một tài liệu (UC-IDV-03, 04)

- **Endpoint:** `GET /api/v1/documents/{id}`
- **Mô tả:** Lấy thông tin chi tiết của một tài liệu. Nếu có query parameter `?download=true`, hệ thống sẽ trả về file để tải xuống.
- **Quyền:** `ROLE >= MEMBER`

Tên	Kiểu	Bắt buộc	Mô tả
<code>download</code>	boolean	Không	Nếu <code>true</code> , trả về file stream để tải.

- **Response (200 OK - Xem chi tiết):**

```

{
  "id": 101,
  "title": "Báo cáo tổng kết tháng 6",
  "description": "Tài liệu báo cáo hoạt động...",
  "source_type": "INTERNAL_UPLOAD",
  "source_location": "/uploads/documents/bao-cao-thang-6.pdf",
  "mime_type": "application/pdf",
  "file_size_bytes": 1024000,
  "status": "PUBLISHED",
  "is_public": true,
  "created_at": "2025-07-01T09:00:00Z",
  "author": { "id": 5, "full_name": "Nguyễn Văn A" },
  "tags": [
    { "id": 1, "name": "Báo cáo" },
    { "id": 7, "name": "Nội bộ" }
  ],
  "folder_id": 1
}

```

- **Response (200 OK - Tải tài liệu):**
 - Trả về file stream với header: `Content-Disposition: attachment; filename="bao-cai-thang-6.pdf"`
- **Error Responses:**
 - **404 Not Found:** `{"error": "Không tìm thấy tài liệu hoặc bạn không có quyền truy cập."}`

3.3. Tạo tài liệu (UC-DM-01)

- **Endpoint:** `POST /api/v1/documents`
- **Mô tả:** Tải lên một tệp và tạo bản ghi tài liệu mới, hoặc chỉ tạo bản ghi với link ngoài.
- **Quyền:** `ROLE >= MEMBER`
- **Request Body (multipart/form-data):**
 - `title`: (string, required)
 - `description`: (string, optional)
 - `is_public`: (boolean, optional, default = false)
 - `status`: (string, optional, default = 'DRAFT', ENUM: 'DRAFT', 'PUBLISHED', 'ARCHIVED')
 - `tags`: (string, optional, chuỗi ID tag cách nhau bởi dấu phẩy, vd: "1,5")
 - `folder_id`: (integer, optional)
 - `file`: (file, conditional required if `source_type` is not `EXTERNAL_LINK`)
 - `source_type`: (string, optional, default = 'INTERNAL_UPLOAD')
 - `source_location`: (string, conditional required if `source_type` is `EXTERNAL_LINK`)
- **Response (201 Created):**

```
{
  "message": "Tài liệu đã được tạo thành công!",
  "document": { ... } // Dữ liệu tài liệu vừa tạo
}
```

- **Error Responses:**
 - **400 Bad Request:** `{"error": "Thiếu các trường bắt buộc."}`

3.4. Cập nhật tài liệu (UC-DM-02, 04, 05, 09)

- **Endpoint:** `PATCH /api/v1/documents/{id}`
- **Mô tả:** Cập nhật một phần thông tin của tài liệu (tiêu đề, mô tả, trạng thái, file, folder, etc.).
- **Quyền:** `ROLE >= MEMBER` (chỉ cho tài liệu của mình), `ROLE >= ADMIN` (cho mọi tài liệu).
- **Request Body (multipart/form-data):** Gửi các trường cần cập nhật.
- **Response (200 OK):**

```
{
```

```
"message": "Cập nhật tài liệu thành công!",
"document": { ... } // Dữ liệu tài liệu đã cập nhật
}
```

- **Error Responses:**
 - **404 Not Found:** {"error": "Không tìm thấy tài liệu."}

3.5. Xóa tài liệu (UC-DM-03)

- **Endpoint:** DELETE /api/v1/documents/{id}
- **Mô tả:** Xóa một tài liệu. Có thể là xóa mềm hoặc xóa vĩnh viễn.
- **Quyền:** ROLE >= MEMBER (chỉ cho tài liệu của mình), ROLE >= ADMIN (cho mọi tài liệu).

Tên	Kiểu	Bắt buộc	Mô tả
force	boolean	Không	Nếu true, xóa vĩnh viễn. Mặc định là xóa mềm. (Chỉ ROLE = ADMIN mới có quyền force=true).

- **Response (204 No Content):** Không có nội dung phản hồi.
- **Error Responses:**
 - **404 Not Found:** {"error": "Không tìm thấy tài liệu."}

II.6 Event management & Event page(Chí quốc)

6.1 Xem danh sách sự kiện

GET /api/v1/events.

Mô tả: Người dùng xem danh sách

Header: content-type: application/json

Authorization: Bearer

Response Success(200):

User:


```
[ { "id": Long, "title": String,  
    "slug": String,  
  
    "content": String,  
  
    "cover_image_url": Url,  
    "location": String,  
  
    "facebook_event_url": Url,  
  
    "start_time": OffsetDateTime,  
  
    "end_time": OffsetDateTime,  
  
    "status": EventStatus  
  
    }  
  
}, ... ]
```

Admin:

```
[ { "id": Long, "title": String,  
    "slug": String,  
  
    "content": String,  
  
    "cover_image_url": Url,
```

"location": String,

"facebook_event_url": Url,

"start_time": OffsetDateTime,

"end_time": OffsetDateTime,

"status": EventStatus,

"is_public": Boolean,

"is_featured": Boolean,

"member_url": Url

}

}, ...]

Response Failure:

- Lỗi không xác định khi truy vấn DB (500)

{

"error": "Lỗi dữ liệu"

}

Yêu cầu quyền truy cập

Role ≥ Guest

6.2 Tìm kiếm sự kiện

Mô tả: Trả về đúng sự kiện mà người dùng yêu cầu

Header: content-type: application/json

Authorization: Bearer

Response Success(200):

GET /api/v1/user/events/?title=...&location=...&status=...

User:

{ "id": Long, "title": String,

 "slug": String,

 "content": String,

 "cover_image_url": Url,

 "location": String,

 "facebook_event_url": Url,

 "start_time": OffsetDateTime,

 "end_time": OffsetDateTime,

 "status": EventStatus

}

GET /api/v1/admin/events/?title=...&location=...&status=...

Admin:

```
{ "id": Long, "title": String,  
  "slug": String,  
  
  "content": String,  
  
  "cover_image_url": Url,  
  "location": String,  
  
  "facebook_event_url": Url,  
  
  "start_time": OffsetDateTime,  
  
  "end_time": OffsetDateTime,  
  
  "status": EventStatus,  
  
  "is_public": Boolean,  
  
  "is_featured": Boolean,  
  
  "member_url": Url  
  
}
```

Response Failure:

- Tham số query sai format(400)

```
{  
  "error": "Lỗi tìm kiếm"  
}
```

Yêu cầu quyền truy cập

Role ≥ Guest

6.3 Tạo sự kiện mới:

POST: /api/v1/events.

Mô tả: Admin tạo một sự kiện mới

Header: content-type: application/json

Authorization: Bearer

Request body:

```
{ "id": Long, "title": String,
```

```
  "slug": String,
```

```
  "content": String,
```

```
  "cover_image_url": Url,
```

```
  "location": String,
```

```
  "facebook_event_url": Url,
```

```
  "start_time": OffsetDateTime,
```

```
  "end_time": OffsetDateTime,
```

```
  "status": EventStatus,
```

“is_public”: Boolean,

“is_featured”: Boolean,

“member_url”: Url

}

Response Success(201):

{

“message”: “Tạo thành công”

}

Response failure:

- Không có token(403)

{

“error” : “Không có token”

}

- Thiếu dữ liệu (400)

{

“error” : “Thiếu dữ liệu”

}

Yêu cầu quyền truy cập

Role ≥ Admin

6.4 Cập nhật thông tin sự kiện

PATCH: /api/v1/events/{id}

Mô tả: Admin cập nhật thông tin của một sự kiện

Header: content-type: application/json

Authorization: Bearer

Request body:

```
{ "id": Long, "title": String,
```

```
    "slug": String,
```

```
    "content": String,
```

```
    "cover_image_url": Url,
```

```
    "location": String,
```

```
    "facebook_event_url": Url,
```

```
    "start_time": OffsetDateTime,
```

```
    "end_time": OffsetDateTime,
```

```
    "status": EventStatus,
```

```
    "is_public": Boolean,
```

```
    "is_featured": Boolean,
```

```
    "member_url": Url
```

```
}
```

Response Success(200):

```
{  
  "message": "Cập nhật thành công"  
}
```

Response failure:

- Thiếu dữ liệu (400)

```
{  
  "error" : "Thiếu dữ liệu"  
}
```

- Không có token(401)

```
{  
  "error" : "Không có token"  
}
```

Yêu cầu quyền truy cập

Role \geq Admin

6.5 Xóa sự kiện

DELETE: /api/v1/events/{id}

Mô tả: Admin xóa một sự kiện

Header: content-type: application/json

Authorization: Bearer

Response Success(200):

```
{  
  "message" : "Xóa thành công"  
}
```


Response failure:

- Thiếu dữ liệu (400)

```
{
```

```
“error” : “Thiếu dữ liệu”
```

```
}
```

- Không có token(401)

```
{
```

```
“error” : “Không có token”
```

```
}
```

Yêu cầu quyền truy cập

Role \geq Admin

III. Các module nội bộ

III.1 Email Service (Nguyễn Anh Kiệt)

III.2 Log Service (Anh Siêu)

Lưu ý

- Viết dựa vào [use case](#) của BA
- Ai chọn module nào sẽ dev module đó luôn.