

# Feedback cho Team Backend - Thiết kế API & CSDL WebHub v1.1

Người gửi: Lead BA Team

Ngày: 16/07/2025

Chào team Backend,

Dưới đây là một số đánh giá và góp ý chi tiết của tôi để chúng ta có thể hoàn thiện hơn nữa sản phẩm.

---

## A. Điểm Tích Cực & Đánh Giá Chung

- Thiết kế Hướng Module Rõ ràng:** API được phân chia thành các module chức năng (Authentication, User, Blog, Project, Document...) rất hợp lý.
- Tài liệu API Chi tiết:** Các endpoint được mô tả rõ ràng về method, request body, response (cả success và failure), và quyền truy cập.
- Luồng nghiệp vụ được cân nhắc kỹ:** Các flow phức tạp như xóa người dùng (chuyển quyền sở hữu nội dung cho Admin hệ thống) hay xử lý upload file/link ngoài cho documents đã được tính đến.

---

## B. Góp ý Chung & Các Điểm Cần Thống Nhất

Đây là những điểm mang tính chiến lược, cần được thống nhất trên toàn bộ hệ thống để đảm bảo tính nhất quán.

- Quy tắc đặt tên (Naming Convention):**
  - Trong CSDL, team đang dùng `snake_case` (ví dụ: `full_name`, `created_at`), rất tốt và chuẩn mực.
  - Tuy nhiên, trong JSON response của API lại chưa thống nhất. Ví dụ: `auth/login` trả về `accessToken` (camelCase), nhưng `users/me` trả về `avatar_url`, `full_name` (snake\_case).
  - Đề xuất:** Thống nhất sử dụng `snake_case` cho tất cả các key trong JSON request và response để đồng bộ với CSDL. Hoặc camelCase chuẩn REST API
- Cấu trúc Phản hồi (Response Structure):**
  - Pagination:** Endpoint `GET /documents` trả về object `pagination`, trong khi `GET /blog-posts` trả về object `meta`.
  - Đề xuất:** Thống nhất sử dụng cấu trúc `meta` cho tất cả các response dạng danh sách. Cấu trúc `meta` nên chuẩn hóa như sau để FE dễ dàng xử lý:

```
"meta": {
```

```

    "current_page": 1,
    "per_page": 10,
    "total_items": 15,
    "total_pages": 2
  }

```

### 3. Cấu trúc Lỗi (Error Structure):

- Tài liệu đang định nghĩa lỗi chung là `{"error": "message"}`. Điều này tốt cho các lỗi đơn giản.
- **Đề xuất:** Với các lỗi xác thực dữ liệu (validation errors) từ form (ví dụ: tạo/sửa bài viết, dự án), nên trả về một cấu trúc chi tiết hơn để FE có thể hiển thị lỗi cho từng trường cụ thể. Ví dụ:

```

// 400 Bad Request
{
  "message": "The given data was invalid.",
  "errors": {
    "title": [
      "Tiêu đề không được để trống."
    ],
    "category_ids": [
      "Phải chọn ít nhất một danh mục."
    ]
  }
}

```

---

## C. Góp ý chi tiết theo từng Module/Endpoint

### 1. Authentication & Authorization (II.1):

- `GET /api/v1/auth/refresh`: Method `GET` thường không được dùng cho các hành động thay đổi trạng thái hoặc tạo ra tài nguyên mới (như tạo access token mới).
- **Đề xuất:** Chuyển endpoint này sang `POST /api/v1/auth/refresh`. Đây là thông lệ chuẩn của OAuth2 và RESTful.

### 2. User Management (II.2):

- `PATCH /api/v1/users/{id}`: Trong tài liệu, phần `multipart/form-data` có trường `user_info: json`. Cách làm này hơi bất tiện cho client và server khi phải parse một chuỗi JSON trong form-data.
- **Đề xuất:** Thay vì `user_info: json`, hãy để các trường `full_name`, `role` là các trường riêng biệt trong `multipart/form-data`, song song với trường `avatar: file`.

### 3. Blog Management (II.3):

- **Phân loại nội dung:** Tài liệu đã đề xuất rất đúng đắn về việc chuyển từ quan hệ 1-nhiều (cột `category_id`) sang nhiều-nhiều để một bài viết có thể có nhiều danh mục. Tuy nhiên, ERD v1.1 đã tiến một bước xa hơn bằng cách dùng bảng `taggables` đã hình, hợp nhất cả Category và Tag vào một hệ thống. Đây là một cải tiến lớn!
- **Đề xuất:** Cần cập nhật lại tài liệu API (II.3) để phản ánh đúng thiết kế này. Bỏ khái niệm "Blog Category" riêng biệt và chỉ sử dụng hệ thống "Tags" chung. Request body khi tạo/sửa bài viết sẽ là `tag_ids: [1, 5, 10]`.

#### 4. Project Management (II.4):

- `POST /api/v1/projects` (Request Body): Cú pháp `"tags": tags [ "id": Long; ]` không phải là JSON hợp lệ.
- **Đề xuất:** Chuẩn hóa request này. Nếu tag đã tồn tại, nên dùng `"tag_ids": [1, 2, 3]`. Nếu cho phép tạo tag mới ngay khi tạo project, có thể dùng `"tags": ["new tag 1", "existing tag 2"]`. Cần làm rõ logic này.

#### 5. Document Management (II.5):

- Response nên được chuẩn hóa lại với cấu trúc `meta` cho pagination như đã góp ý ở mục B.