**Blog API Endpoint Design**

A clean, JWT‑based REST API design for a blogging platform.

**Overview**

This spec covers endpoints for:

* **Auth**: register, login (access + refresh), logout, refresh, password flows
* **Users / Profile**: current user, public profile, deletion
* **Posts**: CRUD with filtering, search, pagination
* **Comments**: CRUD & listing per post or per user
* **Likes**: like/unlike, counters, and liked posts for current user

**Base URL & Versioning(v1)**

* **Base**: *https://api.example.com*
* **Base API URL (versioned)**: *https://api.example.com/api/v1*

**Headers**

* **Authorization:** Bearer <access\_token> for protected routes
* **Content-Type:** application/json

**Pagination**

* **Query:** ?page=1&limit=10
* **Response Body:**

{

"data": [ /\* items \*/ ],

"meta": {

"page": 1,

"limit": 10,

"totalItems": 57,

"totalPages": 6

}

}

**Sorting & Search**

* Sort: ?sort=createdAt&order=desc
* Text search: ?q=neural%20networks
* Filtering examples: ?authorId=abc123&tag=tech&published=true&from=2024-01-01&to=2025-08-29

1. **Authentication**

All authenticated endpoints needed Bearer token in Authorization header.

1. **User Registration: *POST*** */api/v1/auth/register*

Request:

{

"username": "sanjeev",

"email": "sanjeev@gmail.com",

"password": "P@ssw0rd!",

"confirmPassword": "P@ssw0rd!"

}

Response:

{

"message": "User created successfully.",

"status": 201,

"data": { "id": "usr\_101", "username": "sanjeev", "email": "sanjeev@gmail.com" }

}

1. **Login: *POST*** */api/v1/auth/login*

Request:

{

"email": "sanjeev@gmail.com",

"password": "P@ssw0rd!"

}

Response:

{

"message": "Login successful",

"status": 200,

"data": {

"accessToken": "<jwt\_access>",

"refreshToken": "<jwt\_refresh>",

"expiresIn": 900

}

}

1. **Logout:** *POST /api/v1/auth/logout*

* Revokes current refresh token (from cookie/header/body based on implementation).
* **Headers**: Authorization: Bearer <accessToken>
* **Response:** { "message": "Logged out", "status": 200 }
* **Refresh Access Token:** *POST /api/v1/auth/refresh*

Request:

{

"refreshToken": "<jwt\_refresh>"

}

Response:

{

"accessToken": "<new\_access>",

"refreshToken": "<rotated\_refresh>",

"expiresIn": 900

}

1. **Password Management**
2. **Change Password (Authenticated):** *POST /api/v1/auth/change-password*

Request:

{

"currentPassword": "old",

"newPassword": "NewP@ss1",

"confirmNewPassword": "NewP@ss1"

}

Response:

{

"message": "Password updated",

"status": 200

}

1. **Forgot Password:** **POST** /api/v1/auth/forgot-password

Body:

{

"email": "string"

}

1. **Reset Password:** **POST** /api/v1/auth/reset-password

Body:

{

"token": "reset-token",

"newPassword": "string"

}

1. **Users**
2. **Get All Users**

**GET** /api/v1/users

Response:

[

{

"id": 1,

"username": "ujjwal",

"email": "ujjwal@example.com"

},

{

"id": 2,

"username": "sanjeev",

"email": "sanjeev@example.com"

}

]

1. **Delete User Account**

**DELETE** /api/v1/users/:id

Response:

{

"message": "User deleted successfully."

}

1. **Profile**
2. **Get Current User Profile**

**GET** /api/v1/me

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"id": 1,

"username": "ujjwal",

"email": "ujjwal@example.com",

"bio": "Tech enthusiast"

}

1. **Update Current User Profile**

**PUT** /api/v1/me

**Headers:** Authorization: Bearer <accessToken>

Request Body:

{

"username": "ujjwal\_updated",

"bio": "Full Stack Developer"

}

Response:

{

"message": "Profile updated successfully.",

"data": {

"id": 1,

"username": "ujjwal\_updated",

"email": "ujjwal@example.com",

"bio": "Full Stack Developer"

}

}

1. **Get Public User Profile**

**GET** /api/v1/users/:id

Response:

{

"id": 2,

"username": "sanjeev",

"bio": "Writer and Blogger"

}

1. **Posts**
2. **List Posts (filtering, search, pagination)**

**GET** /api/v1/posts?authorId=1&tag=tech&page=1&limit=10&search=term

Response:

{

"page": 1,

"limit": 10,

"total": 2,

"data": [

{

"id": 101,

"title": "My First Blog",

"excerpt": "This is my first post!",

"content": "This is the full content...",

"authorId": 1,

"author": { "id": 1, "username": "ujjwal" },

"tags": ["tech", "intro"],

"publishedAt": "2025-08-01T12:00:00Z"

},

{

"id": 102,

"title": "Second Blog",

"excerpt": "Another interesting post",

"content": "Full content...",

"authorId": 1,

"author": { "id": 1, "username": "ujjwal" },

"tags": ["life"],

"publishedAt": "2025-08-05T08:30:00Z"

}

]

}

1. **Get a Post by Id**

**GET** /api/v1/posts/:id

Response:

{

"id": 101,

"title": "My First Blog",

"content": "This is my first post!",

"authorId": 1,

"author": { "id": 1, "username": "ujjwal" },

"tags": ["tech", "intro"],

"publishedAt": "2025-08-01T12:00:00Z"

}

1. **Create a Post**

**POST** /api/v1/posts

**Headers:** Authorization: Bearer <accessToken>

Request Body:

{

"title": "New Blog Post",

"content": "Here is the blog content...",

"tags": ["coding", "tutorial"],

"isPublished": true

}

Response (201):

{

"message": "Post created successfully.",

"data": {

"id": 201,

"title": "New Blog Post",

"content": "Here is the blog content...",

"authorId": 1,

"tags": ["coding", "tutorial"],

"isPublished": true,

"publishedAt": "2025-08-29T10:00:00Z"

}

}

1. **Update a Post**

**PUT** /api/v1/posts/:id

**Headers:** Authorization: Bearer <accessToken>

Request Body:

{

"title": "Updated Blog Post",

"content": "Updated content...",

"tags": ["coding"]

}

Response:

{

"message": "Post updated successfully.",

"data": {

"id": 201,

"title": "Updated Blog Post",

"content": "Updated content...",

"authorId": 1,

"tags": ["coding"]

}

}

1. **Delete a Post**

**DELETE** /api/v1/posts/:id

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"message": "Post deleted successfully."

}

1. **Get Logged-in User’s Posts**

**GET** /api/v1/me/posts

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"data": [

{

"id": 201,

"title": "New Blog Post",

"authorId": 1

}

]

}

1. **Get Posts from a Specific User**

**GET** /api/v1/users/:id/posts

Response:

{

"data": [

{

"id": 101,

"title": "My First Blog",

"authorId": 1

}

]

}

1. **Comments**
2. **Add Comment to a Post**

**POST** /api/v1/posts/:id/comments

**Headers:** Authorization: Bearer <accessToken>

Request Body:

{

"content": "Great post!"

}

Response (201):

{

"message": "Comment added successfully.",

"data": {

"id": 301,

"postId": 201,

"content": "Great post!",

"authorId": 1,

"createdAt": "2025-08-29T10:05:00Z"

}

}

1. **List Comments for a Post**

**GET** /api/v1/posts/:id/comments?page=1&limit=20

Response:

{

"page": 1,

"limit": 20,

"total": 1,

"data": [

{

"id": 301,

"postId": 201,

"content": "Great post!",

"authorId": 1,

"author": { "id": 1, "username": "ujjwal" },

"createdAt": "2025-08-29T10:05:00Z"

}

]

}

1. **Update a Comment**

**PUT** /api/v1/comments/:id

**Headers:** Authorization: Bearer <accessToken>

Request Body:

{

"content": "Updated comment text"

}

Response:

{

"message": "Comment updated successfully.",

"data": {

"id": 301,

"postId": 201,

"content": "Updated comment text",

"authorId": 1

}

}

1. **Delete a Comment**

**DELETE** /api/v1/comments/:id

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"message": "Comment deleted successfully."

}

1. **Get All Comments by Current User**

**GET** /api/v1/me/comments

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"data": [

{

"id": 301,

"postId": 201,

"content": "Great post!",

"createdAt": "2025-08-29T10:05:00Z"

}

]

}

1. **Likes**
2. **Like a Post**

**POST** /api/v1/posts/:id/likes

**Headers:** Authorization: Bearer <accessToken>

Response (200):

{

"message": "Post liked successfully."

}

1. **Unlike a Post**

**DELETE** /api/v1/posts/:id/likes

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"message": "Post unliked successfully."

}

1. **Get Total Likes for a Post**

**GET** /api/v1/posts/:id/likes/count

Response:

{

"postId": 201,

"likes": 25

}

1. **Has Current User Liked a Post**

**GET** /api/v1/posts/:id/likes/me

**Headers:** Authorization: Bearer <accessToken>

Response:

{

"liked": true

}

1. **Get Posts Liked by Current User**

**GET** /api/v1/me/likes

**Headers:** Authorization: Bearer <accessToken>

Response:

[

{

"id": 201,

"title": "New Blog Post",

"authorId": 1

}

]

**#Notes:**

* Authentication endpoints live under: /api/v1/auth
* User endpoints: /api/v1/users and /api/v1/me
* Posts endpoints: /api/v1/posts
* Comments endpoints: /api/v1/posts/:id/comments and /api/v1/comments/:id
* Likes endpoints: /api/v1/posts/:id/likes and /api/v1/me/likes
* Use Authorization: Bearer <accessToken> header for protected routes.
* Pagination defaults: page=1, limit=10 (unless otherwise specified).
* Filtering & search via query params (e.g., ?authorId=1&tag=tech&search=term).