

REST API design using Swagger

Introduction:

This is a comprehensive approach about REST API for the Blog Site, which enabled clients to manage blogposts more easily and efficiently.

API Requirement and definition:

An API, application programming interface, is a set of defined rules that enable different applications to communicate with each other.

Endpoints

1. Get All Blog Posts

Endpoint: GET /api/blog/posts

Description: Retrieve a list of all blog posts.

Parameters: None

Response:

- Status Code: 200 OK
- Body: Array of Blog Post objects

2. Get Blog Post by ID

Endpoint: GET /api/blog/posts/{id}

Description: Retrieve a specific blog post by its ID.

Parameters: id (Path Parameter): ID of the blog post.

Response:

- Status Code: 200 OK
- Body: Blog Post object

3. Create a New Blog Post

Endpoint: POST /api/blog/posts

Description: Create a new blog post.

Request: Body: Blog Post object (excluding the id field)

Response:

- Status Code: 201 Created
- Body: Created Blog Post object

4. Update a Blog Post

Endpoint: PUT /api/blog/posts/{id}

Description: Update an existing blog post by its ID.

Parameters: id (Path Parameter): ID of the blog post to update.

Request: Body: Blog Post object

Response:

- *Status Code: 200 OK*
- *Body: Updated Blog Post object*

5. Delete a Blog Post

Endpoint: DELETE /api/blog/posts/{id}

Description: Delete a blog post by its ID.

Parameters: id (Path Parameter): ID of the blog post to delete.

Response: Status Code: 204 No Content

Data Models

Blog Post Properties:

- id (Long)
- title (String)
- date (String)
- author (String)
- content (String)

Install Swagger editor:

The Swagger Editor can be accessed here: <https://editor.swagger.io/>.

How to create a New Swagger Document

It starts with basic information such as:

Openapi version,

info: (title, description, and version). And

Servers: url.

openapi: 3.0.0

info:

title: Blog Site API

description: REST API for managing blog posts and user registration

version: 1.0.0

The swagger template attached as converted to **json** file includes (see attachment name: **openapi.json**, for the 'Blogpost' and 'UserModel' schemas.