

# Lab Day 3: Express.js & Mongoose - Blog API

## Challenge

Build a RESTful API for a blog platform with Express.js and Mongoose. Your task is to create a Post model and implement CRUD operations through API endpoints.

## Setup

1. Create a new project directory and initialize npm
2. Install: `express`, `mongoose`, `morgan`
3. Create your file structure:

blog-api/

```
|— server.js
|
|— models/
|   |— Post.js
|   |— routes/
|       |— postRoutes.js
```

4. Configure your MongoDB connection in `server.js`

## Requirements

### 1. Create Post Model

Design a Mongoose schema for blog posts with the following:

- Title
- Content
- Author (reference to User model using ObjectId)
- Tags (array of strings)
- Comments (embedded documents with text, author reference, and date)

- Created/updated timestamps

## **2. Implement Post Routes**

Create the following endpoints:

- GET /api/posts
- GET /api/posts/:id
- POST /api/posts
- PUT /api/posts/:id
- DELETE /api/posts/:id
- POST /api/posts/:id/comments

## **3. Advanced Features (Choose at least 2)**

- Implement pagination
- Add search functionality by title or content
- Create an endpoint to get all posts by a specific user
- Add validation for post content (min/max length)
- Implement sorting options (by date, title, etc.)
- Add a feature to "like" posts

## **4. Error Handling**

Ensure proper error handling for all routes, including:

## **5. Use morgan for logging**

## **Resources**

- [Mongoose Documentation](#)
- [Express.js Documentation](#)