

Blog API Enhancement Lab: Adding Comments with PostgreSQL & Prisma

Overview

In this lab, you will enhance the Express/Prisma/PostgreSQL blog API by adding a comment system with full CRUD functionality. The lab will reinforce your understanding of Prisma schema design, relations, authentication middleware, and API development.

Lab Tasks

0. Finish all crud operations for posts

1. Comment Schema Design

Add a Comment model to your Prisma schema with the following requirements:

- Each comment must be associated with a user (author)
- Each comment must be associated with a post
- Comments should have content, timestamps (created/updated)
- Comments can optionally have a parent comment (for nested replies)

Update your schema.prisma file and run the appropriate Prisma commands to apply these changes.

2. Comment Controller Implementation

Create a comment controller (controllers/comment.controller.js) with the following functions:

- `createComment`: Create a new comment on a post
- `getAllComments`: Get all comments for a specific post
- `getCommentById`: Get a single comment by ID
- `updateComment`: Update a comment (only by the comment author)
- `deleteComment`: Delete a comment (only by the comment author or post author)

3. Comment Routes Implementation

Create comment routes (routes/comment.routes.js) and connect them to your Express application:

- POST /posts/:postId/comments - Create comment on a post
- GET /posts/:postId/comments - Get all comments for a post
- GET /comments/:id - Get a specific comment
- PATCH /comments/:id - Update a comment
- DELETE /comments/:id - Delete a comment

Ensure your routes are protected with authentication middleware where appropriate.

4. Update Post Controller

Enhance your existing post controller with:

- Include comments when getting post details
- Add functionality to count comments on posts
- Update post deletion to cascade delete associated comments

Extension Challenges

Complete these for extra points:

1. Comment Pagination

Implement pagination for comments, allowing clients to specify page size and number.

2. Comment Search

Add a search endpoint for comments based on content.

Deliverables

- Updated Prisma schema with Comment model
- Complete implementation of comment CRUD functionality
- Updated Post controller with comment-related enhancements

Hints

- Remember to run `npm run prisma generate` after updating your schema
- Use middleware to extract the authenticated user's ID from the request
- Test cascading deletes carefully to ensure data integrity

Resources

- [Prisma Relations Documentation](#)
- [Express.js Routing Guide](#)
- [REST API Best Practices](#)
- [JWT Authentication Tutorial](#)