

Day 20 – Basic Routing & Route Middleware

Theme:

Building the foundational routes for the SkillSphere LMS backend to manage courses and users.

Challenge 1 (Easy): Basic Route Setup

User Story:

As a visitor, I want to see a welcome message when I access the home page so that I can verify the server is running.

Problem Statement:

Create an Express server with a root route / returning "Welcome to SkillSphere LMS API".

Expected Outcome:

Visiting <http://localhost:4000/> → Displays the welcome message.

Challenge 2 (Average): Dynamic Routing

User Story:

As a user, I want to view course details using the course ID in the URL so that I can get specific course information.

Problem Statement:

Create a route `/courses/:id` that returns a JSON object `{ id: <id>, name: "React Mastery", duration: "6 weeks" }`.

Expected Outcome:

`/courses/101` → `{ "id": "101", "name": "React Mastery", "duration": "6 weeks" }`

Challenge 3 (Difficult): Route Middleware for Validation

User Story:

As an admin, I want to validate that course IDs are numeric before proceeding so that invalid requests are handled early.

Problem Statement:

Create a route middleware `validateCourseId` that checks if `req.params.id` is numeric before reaching the main route.

Expected Outcome:

- `/courses/101` → Course data

- `/courses/abc` → `{ "error": "Invalid course ID" }`
-

Best Practices for Day 20

- Organize routes in separate files (e.g., `routes/courses.js`).
 - Always validate URL parameters and inputs.
 - Use route-level middleware to isolate logic and improve maintainability.
 - Prefer returning JSON responses over plain text for consistency.
-