

# Day 23 – RESTful APIs & API Middleware

## Theme:

Building fully RESTful APIs for course management with validation and error handling.

---

## Challenge 1 (Easy): CRUD API Setup

### User Story:

As an admin, I want to manage courses using API endpoints.

### Problem Statement:

Create `/api/courses` endpoints for GET, POST, PUT, and DELETE.

### Expected Outcome:

Each route performs the corresponding CRUD operation on an in-memory list.

---

## Challenge 2 (Average): Input Validation

### User Story:

As a developer, I want to validate incoming data before creating a new course.

### Problem Statement:

Use `express-validator` to ensure course name and duration are required fields.

### Expected Outcome:

Invalid data → `{ "error": "Course name is required" }`

---

## Challenge 3 (Difficult): API Rate Limiting

### User Story:

As an admin, I want to prevent API abuse by limiting requests per minute.

### Problem Statement:

Use `express-rate-limit` middleware to restrict users to 5 requests per minute.

### Expected Outcome:

After 5 requests, user receives `{ "error": "Too many requests" }`

---

## **Best Practices for Day 23**

- Always version APIs (`/api/v1/courses`).
  - Centralize error handling in one middleware.
  - Validate all inputs, especially user-facing data.
  - Implement request throttling in production APIs.
-