

Validation Checklist Creation for Udemy Platform



Domain: Online Education (Udemy)

Objective: Create a validation checklist for the main models in the Udemy platform to ensure data integrity and proper input validation.

Jumanah Ibrahim Al-Sulami

Udemy

Udemy is one of the largest online learning platforms in the world that provides access to thousands of courses across various fields such as technology, business, design, and personal development. It allows instructors from around the world to create and share their courses with millions of learners. Users can enroll in courses at any time, learn at their own pace, and receive certificates of completion. Udemy focuses on making education accessible and affordable for everyone, helping individuals improve their skills, advance their careers, and achieve personal goals.

First Model is "Course":

The Course is a model in the system that represents an educational course on the website. It includes information such as the course name, description, price, instructor, and duration.

Validation:

```
@NotEmpty(message = "Course ID cannot be empty")
private String courseId;

@NotEmpty(message = "Title cannot be empty")
@Size(min = 5, max = 30, message = "Title must be between 5 and 30 characters")
private String title;

@NotEmpty(message = "Description cannot be empty")
@Size(min = 30, max = 500, message = "Description must be between 30 and 500 characters")
private String description;

@NotEmpty(message = "Instructor name cannot be empty")
@Size(min = 3, max = 30, message = "Instructor name must be between 3 and 30 characters")
private String instructorName;

@NotNull(message = "Price cannot be null")
@PositiveOrZero(message = "Price must be zero or positive")
private double price;

@DecimalMin(value = "0.0", inclusive = true, message = "Rating cannot be less than 0")
@DecimalMax(value = "5.0", inclusive = true, message = "Rating cannot be more than 5")
private double rating;

@NotNull(message = "Duration cannot be null")
@Positive(message = "Duration must be positive")
private double durationHours;
```

Second Model "Instructor":

The Instructor is a model in the system that represents the person who creates and teaches courses on the website. It includes information such as their name, email, area of expertise, and the number of courses they offer.

Validation:

```
@NotEmpty(message = "Instructor ID cannot be empty")
@Setter(value = @JsonProperty("instructorId"))
@Size(min = 5, max = 15, message = "Instructor ID must be between 5 and 15 characters")
private String instructorId;

@NotEmpty(message = "Full name cannot be empty")
@Setter(value = @JsonProperty("fullName"))
@Size(min = 3, max = 30, message = "Full name must be between 3 and 30 characters")
private String fullName;

@NotEmpty(message = "Email cannot be empty")
@Setter(value = @JsonProperty("email"))
@Email(message = "Email should be valid")
private String email;

@Size(max = 100, message = "Expertise description cannot exceed 100 characters")
@Setter(value = @JsonProperty("expertise"))
private String expertise;

@PositiveOrZero(message = "Total courses must be zero or positive")
@Setter(value = @JsonProperty("totalCourses"))
private int totalCourses;
```

Third Model is " Student " :

The Student is a model in the system that represents the user who enrolls in courses. It includes information such as their name, email, enrolled courses, and enrollment date.

Validation:

```
@NotEmpty(message = "Student ID cannot be empty")
@Setter(value = @JsonProperty("studentId"))
@Size(min = 5, max = 15, message = "Student ID must be between 5 and 15 characters")
private String studentId;

@NotEmpty(message = "Full name cannot be empty")
@Setter(value = @JsonProperty("fullName"))
@Size(min = 3, max = 30, message = "Full name must be between 3 and 30 characters")
private String fullName;

@NotEmpty(message = "Email cannot be empty")
@Setter(value = @JsonProperty("email"))
@Email(message = "Email should be valid")
private String email;

@NotEmpty(message = "Enrolled course ID cannot be empty")
@Setter(value = @JsonProperty("enrolledCourseId"))
private String enrolledCourseId;

@NotNull(message = "Enrollment date cannot be null")
@Setter(value = @JsonProperty("enrollmentDate"))
@PastOrPresent(message = "Enrollment date cannot be in the future")
private LocalDate enrollmentDate;
```

