

# Junior Backend Developer Test (Laravel)

## USE CASE

You are responsible for building an API of an e-learning application that will be used by many schools to manage their courses, students, and teachers.

Your application is basically offering the following:

- Teachers can manage courses and groups.
- Students can enroll in courses.
- Groups gather teachers and students. Teachers and students can join multiple Groups.
- Teachers and Students can add posts to a group.
- Each school can only access its own courses, teachers, students

## Requirements

### 1) Build your database

Write the [Laravel Migrations](#) that will satisfy the specifications:

*n.b. Add any other attribute or entity if needed*

Entity	Attributes
School	Name
User (Teacher or Student)	Name, Email (unique), Password
Course	Title, Description
Group	Name
Post	Content, Creation Date

### 2) Generate sample data to fill your database

Write a [Laravel Command](#) that will generate sample data. The command takes as input the number of schools to generate. Each generated school comes with the following:

- 5 Teachers (per school)
- 1 Course (per teacher)
- 10 Students (per school) (not enrolled in any course)

Example for the command signature: `php artisan generate:schools {number}`

Good Luck!

### 3) Build your RESTful API endpoints

*n.b. Add any other endpoint if needed*

Make the following endpoints:

- Register a student in a school. Use [Laravel Validation](#) to validate the following:
  - ◆ Name is required and less than 20 characters
  - ◆ Email format is correct and unique within the selected school
  - ◆ Password is at least 8 characters long, has at least one alphabetic character and one numeric character
- ◆ *POST: api/schools/{school\_id}/register*
- Login as a student or teacher.
  - ◆ *POST: api/schools/{school\_id}/login*
- Enroll in a course. The authenticated user should be a student and member of the school.
  - ◆ *POST: api/schools/{school\_id}/courses/{course\_id}*
- List available courses. If the authenticated user is a student, return only courses he is enrolled in. If authenticated user is a teacher return all courses
  - ◆ *GET: api/schools/{school\_id}/courses*