

2

Introduction to Spring

Objective

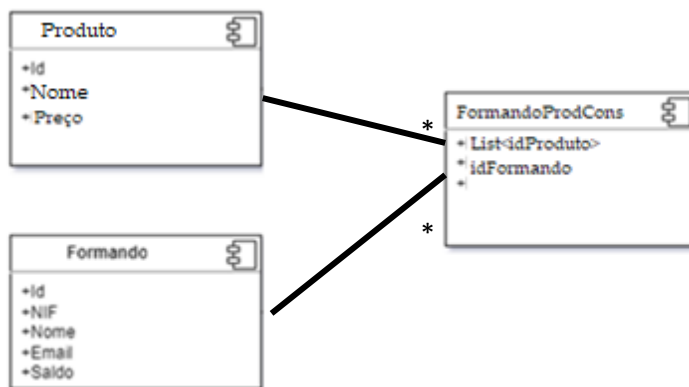
Building a Complex Application with CRUD Operations for Products and Students in Spring Boot Using REST and Exception Handling

Introduction:

In this essay, we will discuss the process of creating a complex application using Spring Boot, REST, and exception handling. Our application will involve CRUD operations for managing Products and Students. By leveraging the power of Spring Boot, we will build a robust and efficient solution.

Specific tasks:

1. Setting up the Development Environment:
2. Designing the Data Model



3. Implementing the Repository Layer (based on csv, in-memory,...)
4. Developing the Service Layer to communicate with the repository in 3.
5. Implementing the REST API
6. Exception Handling
7. Running the Application
8. Testing the Application in a browser
 - a. Through Postman
 - b. Through JUnit and Curl

Continue in the next page...

Minimum REST API Calls:

1. Define the resource **GET: /api/produtos** to get all existing products in the bar;
2. Define the resource **GET: /api/produto/{id}** to get a specific product in the bar;
3. Define the resource **GET: /api/formandos** to obtain all students;
4. Define the resource **GET: /api/formando/{id}** to get a specific student;
5. Define the resource **GET: /api/formando/{id}/bar** to get all products that a specific student consumed;
6. Define the resource **GET: /api/formando/{id}/produto/{id}** to get a specific product that a specific student consumed;
7. Define the resource **GET: /api/formando/{id}/saldo** to get the funds available by a student;
8. Define the resource **POST: /api/formando/{id}/compra/{id}** that allow to buy one product of the bar, updating the funds of the student;
9. Other resources that consider relevant for the application.