Marx Application Development Exercise

Context

In this exercise, you will be tasked with developing an application called "Marx" using Java 17 for the backend and HTML5 with Angular for the frontend.
The application's primary function is to load data from a CSV file named "Coherency.csv" and manipulate it through a user-friendly interface.
It involves importing, editing, selecting, excluding columns/rows, adding custom fields (e.g., "Comments"), and validating incoming data at multiple levels before persisting it in a database.
The exercise is expected to adhere to the latest web standards and optimization practices.

Detailed Context

Web Development

- Load the "Coherency.csv" file and display its content in a responsive table.
- Allow users to select specific columns and/or rows.
  - Enable the selection to be the sole content in the table.
  - Allow the removal of selected items from the displayed table.
  - Implement CRUD (Create, Read, Update, Delete) operations on the selected items.
- Create a fuzzy-search input textbox to find rows within the table.
- Introduce an "Comments" column as an optional field for user input in the table.
- Enable CRUD operations on rows within the table.

Technical Stack

- Application in Java 17, utilizing its features extensively.
- Dependency management using Maven. Consult Yves for configuring access to Artifactory for dependencies.
- Utilize a real database for the application.
- Implement Spring Boot 3.1.x for the entire application.
- Employ Spring Boot Test for writing unit tests.
- Maintain a test coverage of at least 60%.
- Utilize Spring Boot Data, Spring Boot Rest, and Spring Boot MVC.
- Integrate NodeJs version 20.5 into the project.
- Use Angular 16.2 for building a responsive frontend connected to the backend.
- Implement data validation as required.

Optional Technologies in the Stack

- Consider integrating Swagger for API documentation.
- Use an in-memory database (H2 database) for testing purposes.
- Implement Spring Boot Actuator and create a frontend page to display relevant values.

Constraints

- Ensure adherence to the specified technology stack and versions.
- The application must handle CSV data import, manipulation, and persistence.
- Web components should be developed to be responsive.
- Validation must be implemented at multiple levels to ensure data integrity.
- Test coverage must meet or exceed 60%.
- The exercise should follow industry best practices for both Java and web development.
- Optional technologies are encouraged but not mandatory for completion.

This exercise will assess your proficiency in Java 17, Spring Boot, Angular, and web development, as well as your ability to implement CRUD operations, data validation, and overall code quality.