

Project Brief: Java Spring Boot Todo List API

Objective

Develop a backend API for a Todo list application using Spring Boot. The emphasis is on core Java web development concepts, API design (REST or gRPC), and thorough testing practices.

Time Expectation

This exercise should not take more than 4-5 hours to complete.

Requirements

- **Todo Resource**
 - Define a Todo resource model (properties like id, description, completion status)
- **API Endpoints**
 - **RESTful API (Mandatory)**
 - GET /todos (retrieve all todos)
 - GET /todos/{id} (retrieve a single todo)
 - POST /todos (create a todo)
 - PATCH /todos/{id} (update a todo)
 - DELETE /todos/{id} (delete a todo)
 - **gRPC API (Bonus)**
 - Implement the equivalent list of operations above using gRPC.
- **Persistence**
 - Use an SQLite database for persistence. Use JDBC and SQL for database interactions.
- **Bonus Points**
 - Provide a `Dockerfile` that will run your application in a containerized environment.

Technical Considerations

- **Spring Boot and Spring MVC:** Structure the web application, and use Spring MVC to implement the REST API endpoints.
- **gRPC (Bonus):** If you're pursuing the bonus, use gRPC-Java to define services and generate the necessary code for the gRPC API.
- **Testing**

- **Unit Tests (JUnit):** Thoroughly test API controllers, service layers, and any data access logic.
 - **Integration Tests:** Test the API endpoints and their interaction with the SQLite database.
- **Code Quality:** Clean, well-structured code following Java best practices.
- **Error Handling:** Implement meaningful error handling and response codes within the API.

Evaluation Criteria

- **Functionality:** Does the API meet the specified requirements?
- **API Design:** Is the REST API well-structured? If gRPC is included, is it implemented correctly using gRPC principles?
- **Database Interaction:** Effective use of JDBC and SQL with SQLite.
- **Testing:** Thoroughness of unit and integration tests, ensuring good code coverage.