Document and Author Management Web Application

You are tasked with developing a feature-rich web application for managing documents and authors. Each document has a title, body, authors, and references. Authors have a first name and a last name.

Requirements:

- Utilize the Spring Framework to build the project.
- Implement a set of RESTful services for:
  - Adding, deleting, editing, and viewing authors.
  - Adding, deleting, editing, and viewing documents.
- Store all data in a SQL database. An in-memory database like H2 is acceptable for this task.
- Write unit tests to ensure the correctness of your key functionalities.

Additional Requirements:

- Implement efficient and performant code to handle CRUD operations.
- Maintain clarity and readability of your code through proper documentation and coding standards.
- Include error handling mechanisms to provide meaningful feedback to users in case of failures.
- Implement validation for input data to ensure data integrity and security.
- Design and document your API endpoints using clear and consistent naming conventions and proper HTTP methods.

Bonus Requirements:

- Document the APIs using a framework like Swagger for improved API documentation and client integration.
- Implement Authentication and Authorization to secure your endpoints, allowing only authorized users to access and modify data.
- Implement a message queue system using any message broker (e.g., RabbitMQ, Kafka) to publish changes made to authors or documents.
- Develop a consumer application within the web app to consume events from the message queue. When an event is received, delete the specified author and all documents related to it.
- Dockerization for postgress and message-broker

Evaluation Criteria:

Your code will be evaluated based on the following criteria:

- Correctness and completeness of the functionality: Ensure all required features are implemented and work as expected.
- Efficiency and performance of the program: Optimize your code for efficiency and responsiveness.
- Clarity and readability of the code: Write clean, well-structured code with clear documentation and comments where necessary.
- Implementation of bonus requirements: Evaluate the implementation of bonus features such as API documentation, authentication, message

queue integration, and event consumption.

Please share as a public the link for your project on github