Task Management System Documentation

Overview

The **Task Management System API** is a **Spring Boot** application designed to manage tasks efficiently through RESTful endpoints. Its architecture ensures a clean separation of concerns with modular layers, and it includes robust testing through the **customTest** package within the test directory.

Technologies Used

- **Spring Boot**: Framework for building the API.
- **Spring Data JPA**: Simplifies database interactions.
- **H2 Database**: Relational database for storing task data.
- **Maven**: Dependency management and project building.
- **Postman**: API testing tool.
- **JUnit**: Unit testing framework.
- **Spring Security**: Secures endpoints using Basic Authentication

Software used

1. **Java Development Kit (JDK 17+)**
2. **Apache Maven (3.9.9+)**
3. **IDE**: Spring Tool Suite (STS).
4. **H2 Database**: Database management.
5. **Postman**: For API testing and verification.

System Architecture

The architecture includes the following layers:

1. **Controller Layer**:
   - Handles HTTP requests and delegates them to the service layer.
   - Prepares HTTP responses for the client.
   - Located in com.example.www.controller.
2. **Service Layer**:
   - Implements core business logic.
   - Communicates with the repository layer to handle data operations.
   - Located in com.example.www.service.
3. **Repository Layer**:
   - Manages database interactions using Spring Data JPA.
   - Located in com.example.www.repository.
4. **Model Layer**:
   - Represents the structure of entities (e.g., Task) that map directly to database tables.
   - Located in com.example.www.entity.
5. **Mapper Layer**:
   - Converts entities to DTOs and vice versa, ensuring separation between persistence and client-facing data.
   - Located in com.example.www.mapper.
6. **DTO Layer**:
   - Simplifies and formats data exchanged between the API and clients.
   - Located in com.example.www.dto.
7. **Security Layer**:
   - Configures authentication and authorization for secure access to API endpoints.
   - Located in com.example.www.security.
8. **Testing Layer (customTest)**:
   - Located in the test directory as com.example.www.customTest.
   - Includes unit tests for individual components and integration tests to validate end-to-end functionality.
9. **Exception Layer (exception)**:
   - Located at com.example.www.exception.
   - Handle the Global and Custom Exception

EndPoint OverView

1. **Create Task**
   - **Endpoint:** POST /tasks
   - **Description:** Creates a new task.
   - **Request Body:**

     json
     Copy code
     {
      "title": "Task Title",
      "description": "Task Description",
      "dueDate": "2024-11-21"
     }

2. **Get All Tasks**
   - **Endpoint:** GET /tasks
   - **Description:** Retrieves a list of all tasks.
   - **Response Body:**
   -
3. **Get Task by ID**
   - **Endpoint:** GET /tasks/id
   - **Description:** Retrieves a specific task by its ID.

4. **Update Task**
   - **Endpoint:** PUT /tasks/id
   - **Description:** Updates an existing task.
   - **Request Body:**

     json
     Copy code
     {
      "title": "Updated Task Title",
      "description": "Updated Task Description",
      "dueDate": "2024-11-25"
     }

5. **Delete Task**
   - **Endpoint:** DELETE /tasks/id
   - **Description:** Deletes a specific task by its ID.
   - **Response Body:**

6. **Search Tasks by Id**
    - **Endpoint:** GET /tasks/id
    - **Description:** Searches tasks by title.
    - **Query Parameter:** title (the title to search for).
    -

Authentication and Authorization

- Username: admin
- **Password: admin**
- **Role: admin**