**Step-by-Step Instructions**

1. **Add Dependencies** Add the following dependencies to your pom.xml file:

<dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

   <groupId>io.jsonwebtoken</groupId>

   <artifactId>jjwt</artifactId>

   <version>0.9.1</version>

</dependency>

1. **Create JWTUtil Class** Implement a utility class for handling JWT token operations such as creation and validation.
2. **Add Entities** Define the necessary entities for your application:
3. **Create UserRepo** Implement a repository interface for the User entity to interact with the database.
4. **Create MyUserDetails** Create a class that implements UserDetails to represent the authenticated user.
5. **Create MyUserDetailsService** Implement a service class that implements UserDetailsService to load user-specific data.
6. **Create Filter** Implement a filter class to handle JWT token validation and authorization.
7. **Create SecurityConfig** Configure Spring Security by overriding the necessary methods to set up security settings.
8. **Create Security DTOs** Define Data Transfer Objects (DTOs) for handling authentication requests and responses:
9. **Create AuthController** Implement a REST controller for handling authentication-related endpoints.
10. **Create AuthService and Implementation** Define a service interface and its implementation for handling authentication logic.

* **User**: Represents the user entity – (*Not related to security*).
* **Role**: Represents the role entity.
* **LoginRequest**: Represents the login request data.
* **LoginResponse**: Represents the login response data.

**Summary**

This guide provides a structured approach to implementing token authorization in a Spring Boot project. Ensure to follow the steps in order, starting with adding the necessary dependencies and progressively creating and configuring the required classes and components. This will help you set up a secure authentication system using JWT tokens.