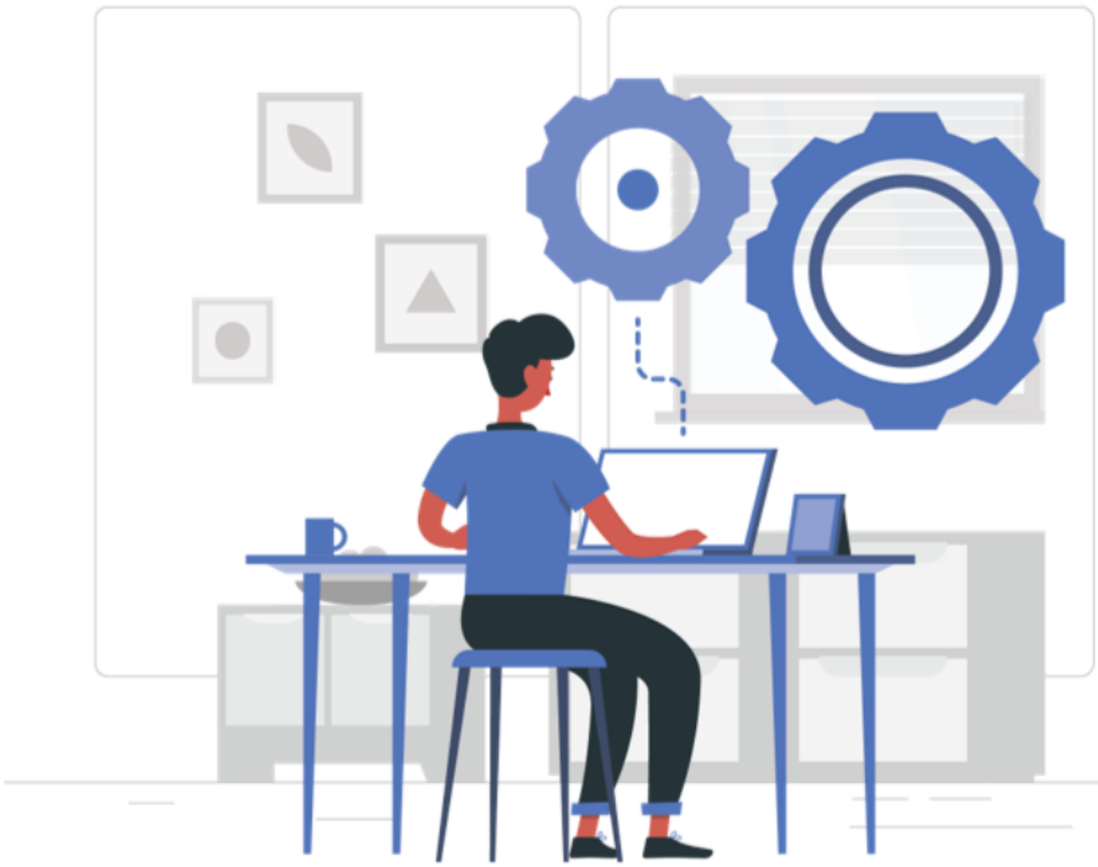


Challenge

Authenticate a Backend Application by Using JASON Web Token (JWT)

User Authentication

Users of an application must be registered users of the application and authenticated at login.

Only registered and authenticated users can log in.

Create a Spring Boot application with an entity class User with userId, userName, userPassword, and userAddress. Implement authentication using JWT.

CHALLENGE

Instructions for the Challenge

- Click on the boilerplate.
- Fork the boilerplate using the fork button 

- Select your namespace to fork the project.
- Clone the project into your local system.
- Open the project in the IntelliJ IDE.
- Execute the test cases given in the test folder.
- In the application.properties file there are two configurations to execute the application, one for local executions and other for Hobbes execution.
- When executing the application on local machine, comment the hobbes configuration and uncomment the local configuration and change username and password to connect to database as per your local config.
- Before pushing the solution to the repository comment the local configuration and uncomment the hobbes configuration.
- Push the solution to git.

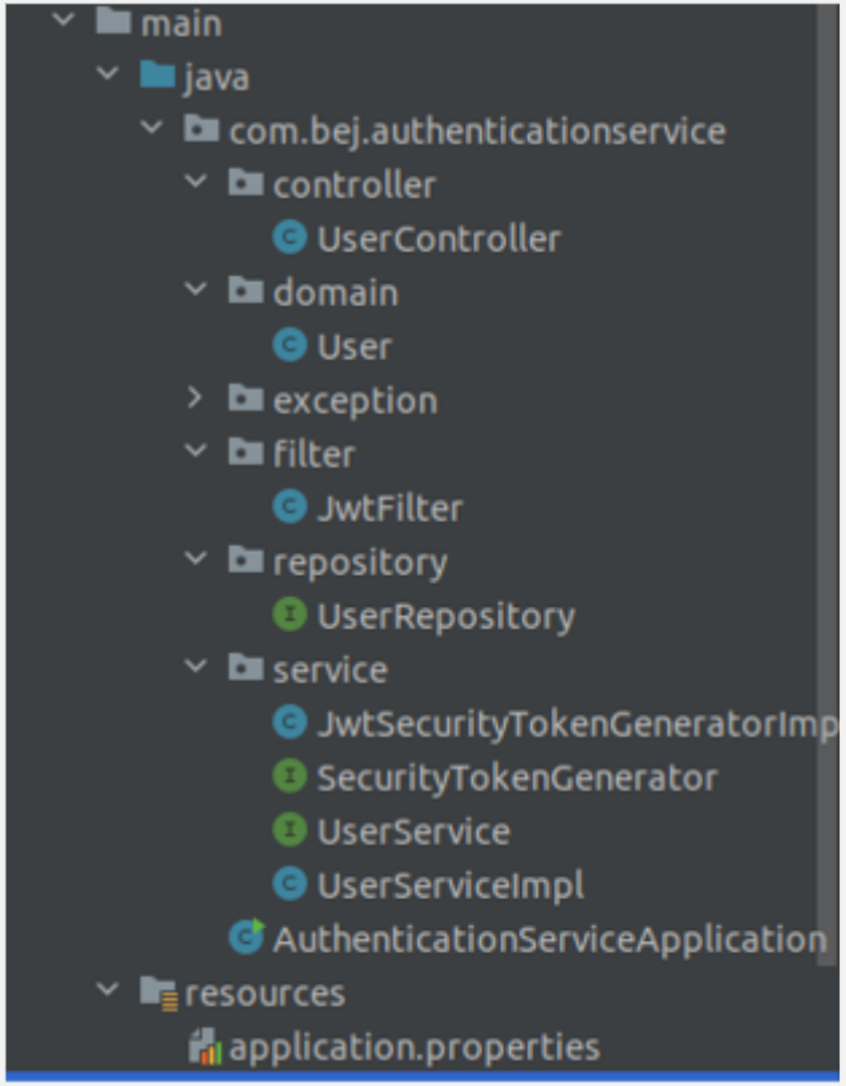
Implementation Environment

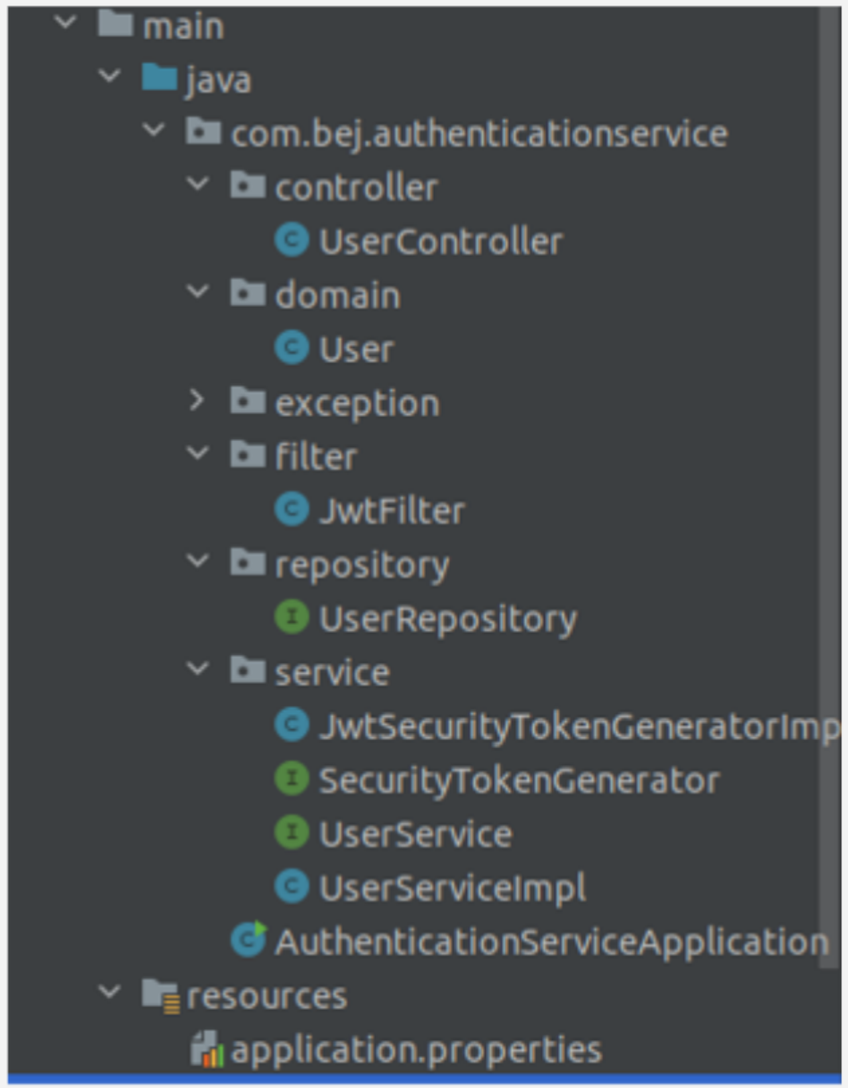
- Create a Spring Boot application from the Spring Initializr.
- Add the necessary dependencies in pom.xml.
- Download the project into your local machine.
- Extract the zip file.
- Export the project in your local IDE.
- `io.jsonwebtoken` needs to be taken from maven repository.

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt</artifactId>
    <version>0.9.1</version>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
</dependency>
```

Task: Challenge 1

- Create all the packages and classes with the same structure as shown.
- Controller will have handler methods for `login()`, `register()`, `getAllUsers()`, and `deleteUser()`.
- The register should save all the user details in the database.
- While logging in with the correct `userName` and `userPassword`, the application should generate a JWT token and send it as response.
- Thd subsequent request to `getAllUsers()` and `deleteUser`() should have a token with it in the Authorization header.
- If these request comes without token then a message should be displayed `"Invalid or Missing Token"`.





Task: Challenge 1 (contd.)

- Service will have two implementation classes: `UserServiceImpl` and other is `SecurityToken Generator`.
- The `SecurityTokenGenerator` class will generate the token.
- Create the domain and exception classes.
- Implement filter class to Verify the JWT token.
- Add the `FilterRegistrationBean` class in the main method.
- In the application.properties file add all the configuration for the MySql Database.

Submission Instructions

- Submit the practice or challenge on hobbes.
- Login to hobbes using your credentials.
- Click on **Submission** in the left navigation bar.
- The **Submit for evaluation** page is opened.
- Select the solution repository `bej-autheticate-using-JWT-mc-1-user-authentication` against which your submission will be evaluated, under **Assignment Repository**
- Select your solution repository `mc-1-user-authentication` under **Search Submission Repo**
- Click on **Submit**.
- The results can be viewed in the **Past Submissions** screen.