Handling User Authentication

Writeup

Steps

1. Start
2. Create a Spring boot application with Maven based on J2SE 1.8
3. Add Spring Data JPA, MySQL Driver, Spring Boot DevTools Dependencies
4. Create a POJO of name **UserEntity** with attributes **username (Primary Key – Id), password** and **email.**
5. Create an interface **UserRepo** which extends **JpaRepository** interface.
6. Create queries and methods in the UserRepo interface
7. Create a **Service** class – **UserAuthService** which has an autowired entity of UserRepo.
8. Create methods in UserAuthService which handle login and register operations.
9. **login() method:**
    - The login() method takes username and password as parameters.
    - It returns the UserEntity object if present otherwise null is returned.
    - If the returned object is null, it is understood that either password/username or both are incorrect.
10. **register() method:**
    - The register() method takes an UserEntity object.
    - Checks are made to ensure that the username and email ID entered are already not in use.
    - The method returns string which tells whether the username or email ID already exists.
    - If the username and email ID are note already in use, the user object is saved and fed into the database.
    - A string stating successful account creation is returned.

11. A JUnit 5 test unit - **AuthenticationTest** is created in the project.
    - It has methods to test Successful and Unsuccessful User Registration and Login.
    - It has an autowired object of the UserAuthService class.
    - Methods of the object are called and assertEqual(), assertNotNull() and asserNull() methods are used in the test methods of the AuthenticationTest Class.
    - Messages are also printed in the console to understand the flow.
12. End