**USER AUTHENTICATION**

//com.example.demo

package com.example.demo;

import org.apache.catalina.User;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.demo.service.UserService;

@RestController

public class UserController {

@Autowired

UserService userSer;

@PostMapping("/userExists")

public ResponseEntity<String> login(@RequestBody User user) {

if (userSer.userExistsInDatabase(user)) {

return ResponseEntity.status(HttpStatus.OK).build();

}

return ResponseEntity.status(HttpStatus.BAD\_REQUEST).build();

}

}

//com.example.demo.pojo

package com.example.demo.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity

@Data

@NoArgsConstructor

public class User {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

Long id;

@Column

String username;

@Column

String password;

}

//com.example.demo.repo

package com.example.demo.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.example.demo.pojo.User;

@Repository

public interface UserRepo extends JpaRepository<User, Long>{

boolean existsByUsernameAndPassword(String username, String password);

}

//com.example.demo.service

package com.example.demo.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.example.demo.pojo.User;

import com.example.demo.repo.UserRepo;

@Service

public class UserService {

@Autowired

private UserRepo repo;

public boolean userExistsInDatabase(User user) {

return repo.existsByUsernameAndPassword(user.getUsername(), user.getPassword());

}

public boolean userExistsInDatabase(org.apache.catalina.User user) {

// TODO Auto-generated method stub

return false;

}

}

//JUNIT TEST

package com.example.demo;

import static org.junit.jupiter.api.Assertions.\*;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtendWith;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

import com.example.demo.pojo.User;

import com.example.demo.repo.UserRepo;

import com.example.demo.service.UserService;

@ExtendWith(MockitoExtension.class)

class UserServiceTest {

@Mock

private UserRepo userRepo;

@InjectMocks

private UserService userService;

@Test

void testUserExistsInDatabase() {

// Given

User user = new User();

user.setUsername("naveen");

user.setPassword("1000");

// Mocking the behavior of UserRepo.existsByUsernameAndPassword

when(userRepo.existsByUsernameAndPassword("naveen", "1000")).thenReturn(true);

// When

boolean userExists = userService.userExistsInDatabase(user);

// Then

assertTrue(userExists);

// Verify that the existsByUsernameAndPassword method was called with the correct arguments

verify(userRepo, times(1)).existsByUsernameAndPassword("naveen", "1000"); (//"Naveen", "2000" test pass)

}

}