**2022년도 졸업프로젝트[Back-end]**

**팀이름: 황제펭귄**

주제: TLS를 구현을 통한 홈페이지 제작 및 보안 채널 로그인 기능 구현

|  |  |  |  |
| --- | --- | --- | --- |
| **작성자** | 이유민 | **작성일자** | 2022.04.06 |

* **금일 진행 상황**

|  |  |
| --- | --- |
| **진행 상황** | **비고** |
| **질문 및 진행방식에 대한 논의**  **진행 중**  **완료**  **1.새로운 방식의 패키지 구성**  **domain과 web으로 분할**   1. domain하위에는 핵심로직 구현 2. web은 타임리프나 머스태치 rest api같이 다양한 방식으로 교체될수 있기에 web은 domain에 대한 class를 가져올 수 있으나,반대로 domain은 web에 대한 클래스를 가져올 수 없는 단방향 의존성이 형성된다.     **2.회원가입 재구현 및 테스트 완료.**  **1.User Entity Class**  package com.EmperorPenguin.SangmyungBank.api.user.add.domain.User;  import lombok.Builder; import lombok.Getter; import lombok.NoArgsConstructor;  import javax.persistence.\*; import javax.validation.constraints.NotEmpty; import javax.validation.constraints.NotNull;  @Getter @Entity @NoArgsConstructor public class User {  @Id  @GeneratedValue(strategy = GenerationType.*IDENTITY*)  private Long userId;   @Column(columnDefinition = "text", nullable = false)  @NotEmpty  private String loginId;  @NotEmpty  private String password;  @NotEmpty  private String name;   @Column(columnDefinition = "text", unique = true,nullable = false)  @NotEmpty  private String email;   @Column(columnDefinition = "integer", nullable = false)  @NotNull  private int age;   @Column(length = 2, nullable = false)  @NotNull  private char sex;   @Column(length = 15, unique = true, nullable = false)  @NotEmpty  private String phoneNumber;   @Enumerated(EnumType.*STRING*)  @Column(nullable = false)  private Role role;   @Builder  public User(String loginId, String password, String name, String email, int age, char sex, String phoneNumber,Role role)  {  this.loginId = loginId;  this.password = password;  this.name = name;  this.email = email;  this.age = age;  this.sex = sex;  this.phoneNumber = phoneNumber;  this.role = role;  }   public String getRoleKey(){  return this.role.getKey();  } }  **2.UserService class**  package com.EmperorPenguin.SangmyungBank.api.user.add.service;  import com.EmperorPenguin.SangmyungBank.api.user.add.domain.User.Role; import com.EmperorPenguin.SangmyungBank.api.user.add.domain.User.User; import com.EmperorPenguin.SangmyungBank.api.user.add.domain.repository.UserRepository;  import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service;  import java.util.List; import java.util.Optional; import javax.transaction.Transactional;  @Transactional @Service public class UserServiceImpl implements UserService {   private final UserRepository userRepository; // private final PasswordEncoder passwordEncoder;   public UserServiceImpl(UserRepository userRepository ) {  this.userRepository = userRepository;  }    // Id를 통해 DB에 있는지 검색을 하고 있다면 존재한다는 error를 발생  // 아니라면 등록과정을 진행한다.  @Override  public boolean ValidateDuplicateUser(String loginid) {  User user = userRepository.findByLoginId(loginid)  .orElse(null);  if(user == null)  return false;  else  return true;  }   /\*  회원 등록을 위한 작업  \*/  @Override  public User register(String loginId, String password, String name, String email, int age, char sex, String phoneNumber)  { // String encodedPassword = passwordEncoder.encode(password);  User user = User.*builder*()  .loginId(loginId)  .password(password)  .name(name)  .email(email)  .age(age)  .sex(sex)  .phoneNumber(phoneNumber)  .role(Role.*USER*)  .build();   return userRepository.save(user);  }   /\*  회원 로그인을 위한 작업으로 ID, password를 제공하면  password가 맞는지 확인 후 user 데이터를 제공  \*/  @Override  public User authenticate(User loginUser) {  User findUser = userRepository.findByLoginId(loginUser.getLoginId())  .orElse(null);  if(findUser == null)  {  return null;  } // else if(!passwordEncoder.matches(User.getPassword(), user.getPassword())) // { // return null; // }  else if(!findUser.getPassword().equals(loginUser.getPassword())) {  return null;  }  return findUser;  }   @Override  public List<User> findUsers() {  return userRepository.findAll();  }   @Override  public Optional<User> findOne(String userId) {  return userRepository.findByLoginId(userId);  } }  **3.Test**  package com.EmperorPenguin.SangmyungBank.api.user;  import com.EmperorPenguin.SangmyungBank.api.user.add.domain.User.User; import com.EmperorPenguin.SangmyungBank.api.user.add.domain.repository.UserRepository; import com.EmperorPenguin.SangmyungBank.api.user.add.service.UserService; import org.junit.jupiter.api.Test; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.boot.test.context.SpringBootTest;  import javax.transaction.Transactional;  import static org.assertj.core.api.AssertionsForClassTypes.*assertThat*;  @Transactional @SpringBootTest public class addUserTest {   @Autowired  UserService userService;  @Autowired  UserRepository userRepository;   @Test  void 회원가입() {  // given  String loginId = "SpringTest1";  String password = "12345";  String name = "Spring";  String email = "SpringTest@email.com";  char sex = 'M';  int age = 13;  String phoneNumber = "010-1111-1111";  // when  userService.register(loginId,password,name,email,age,sex,phoneNumber);  // then  User findUser = userService.findOne(loginId).get();  *assertThat*(loginId).isEqualTo(findUser.getLoginId());  } }  **진행 예정** |  |
| * **특이사항 / 협업 사항** | |