**2022년도 졸업프로젝트**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **작성자** | 김도현 | **작성일자** | 2022.06.08 |

* **금일 진행 상황**

|  |  |
| --- | --- |
| **진행 상황** | **비고** |
| **질문 및 진행방식에 대한 논의**  **진행 중**  **완료 1. Swagger ui 변경**  @Api(tags="4. 카드")  @ApiOperation(value="1. 카드 생성", notes = "사용자의 아이디와 계좌번호, 카드종류를 받아 카드 생성")  **2. home controller 추가**  @Api(tags = "0. 홈") @RestController @RequiredArgsConstructor @RequestMapping("/users") public class HomeController {  private final MemberService memberService;  private final ResponseService responseService;   @PostMapping(path = "/login")  @ApiOperation(value="1. 로그인", notes = "사용자의 아이디와 비밀번호를 받아 로그인을 합니다.")  public BaseResult authUser(@ApiParam @RequestBody MemberLoginReq memberLoginReq){  // 이전 boolean 통해 오류를 검출하는 방식으로 작동  // refactoring 이후 Exception 통한 예외처리로 로직 변경  // 없는 아이디, 잘못된 비밀번호에서 오류 발생.  try{  return responseService.singleResult(memberService.login(memberLoginReq).toLoginDto());  }catch (Exception e){  return responseService.failResult(  e.getMessage()  );  }  }   // 로그아웃 기능   @PostMapping(path = "/register")  @ApiOperation(value="3. 회원가입", notes = "사용자 정보를 받아 사용자를 저장합니다.")  public BaseResult addUser(@ApiParam @RequestBody MemberRegisterReq memberRegisterReq) {  // 이전 boolean 통해 오류를 검출하는 방식으로 작동  // refactoring 이후 Exception 통한 예외처리로 로직 변경  // 아이디 중복, 이메일 중복, 전화번호 중복, 비밀번호 규칙 검증, 올바르게 비밀번호를 2번 입력했는지 확인  try {  memberService.register(memberRegisterReq);  return responseService.successResult();  } catch (Exception e) {  return responseService.failResult(  e.getMessage()  );  }  }   @PostMapping(path = "/find\_Password")  @ApiOperation(value = "4. 비밀번호 찾기", notes = "사용자의 질문과 정답을 검증해 임시비밀번호를 제공합니다.")  public BaseResult findPassword(@ApiParam @RequestBody MemberFindPasswordReq memberFindPasswordReq){  try {  return responseService.singleResult(  memberService.setTemplatePassword(memberFindPasswordReq));  }catch (Exception e){  return responseService.failResult(  e.getMessage()  );  }  }   @PostMapping(path = "/updateTempPassword")  @ApiOperation(value = "5. 임시 비밀번호를 가진 사용자의 비밀번호 변경",notes = "임시 비밀번호를 제공받은 유저의 새로운 비밀번호로 비밀번호를 변경합니다.")  public BaseResult updateTemplateUserPassword(@ApiParam @RequestBody MemberPasswordUpdateReq memberPasswordUpdateReq){  try {  memberService.updateTemplatePassword(memberPasswordUpdateReq);  return responseService.successResult();  }catch (Exception e){  return responseService.failResult(  e.getMessage()  );  }  }   }  **진행 예정** | re |
| * **특이사항 / 협업 사항** | |