**2022년도 졸업프로젝트**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **작성자** | 김도현 | **작성일자** | 2022.07.05 |

* **금일 진행 상황**

|  |  |
| --- | --- |
| **진행 상황** | **비고** |
| **질문 및 진행방식에 대한 논의**  **진행 중**  **완료 1. JWT 기능 추가**  **JwtAuthenticationFilter :**  **// /login 요청을 하면 로그인 시도를 위해서 실행되는 함수**  **@Override**  **public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response)**  **throws AuthenticationException {**  **System.out.println("로그인시도중");**  **try{**  **ObjectMapper om = new ObjectMapper();**  **MemberLoginReq login = om.readValue(request.getInputStream(),MemberLoginReq.class);**  **System.out.println(login);**  **UsernamePasswordAuthenticationToken authenticationToken =**  **new UsernamePasswordAuthenticationToken(login.getLoginId(), login.getPassword());**  **System.out.println(login.getLoginId() + login.getPassword());**  **//PrincipalDetailsService의 loadUserByUsername()함수가 실행됨**  **Authentication authentication= authenticationManager.authenticate(authenticationToken);**  **PrincipalDetails principalDetails =(PrincipalDetails)authentication.getPrincipal();**  **System.out.println(principalDetails.getMember().getLoginId());**  **System.out.println("1===========================");**  **// 권한 관리를 Security가 해주므로 넘겨준다.**  **return authentication;**  **}catch (IOException e){e.printStackTrace();}**  **System.out.println("2====================================");**  **return null;**  **}**  **// 인증이 정상적으로 완료되면 실행된다.**  **// JWT 토큰을 만들어서 request 요청한 사용자에게 JWT 토큰을 response 해준다.**  **@Override**  **protected void successfulAuthentication(HttpServletRequest request, HttpServletResponse response, FilterChain chain, Authentication authResult)**  **throws IOException, ServletException {**    **System.out.println("successfulAuthentication 실행됨:인증완료");**  **PrincipalDetails principalDetails =(PrincipalDetails)authResult.getPrincipal();**  **//RSA가 아닌 Hash암호방싯**  **String jwtToken=JWT.create()**  **.withSubject((principalDetails.getUsername()))**  **.withExpiresAt(new Date(System.currentTimeMillis()+(60000\*10)))**  **.withClaim("memberId",principalDetails.getMember().getMemberId())**  **.withClaim("username",principalDetails.getMember().getLoginId())**  **.sign(Algorithm.HMAC512("sumung"));**  **response.addHeader("Autorization","Bearer"+jwtToken);**  **}**  **}**  **진행 예정** | re |
| * **특이사항 / 협업 사항** | |