**2022년도 졸업프로젝트**

**팀이름: 황제펭귄**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **작성자** | 이유민 | **작성일자** | 2022.05.03 |

* **금일 진행 상황**

|  |  |
| --- | --- |
| **진행 상황** | **비고** |
| **질문 및 진행방식에 대한 논의**  **진행 중**  **완료**   1. **인증세션 구현**   -UserLoginControllrer[문제점 해결]  [return response]🡪  [return ResponseEntity.*status*(HttpStatus.*OK*)0.body(auth);  package com.EmperorPenguin.SangmyungBank.api.users.login.controller;  import com.EmperorPenguin.SangmyungBank.api.users.login.domain.loginForm.LoginForm; import com.EmperorPenguin.SangmyungBank.api.users.login.service.UserLoginService; import lombok.RequiredArgsConstructor; import org.springframework.http.HttpCookie; import org.springframework.http.HttpStatus; import org.springframework.http.ResponseEntity; import org.springframework.web.bind.annotation.\*; import com.EmperorPenguin.SangmyungBank.api.users.login.service.SessionService; import com.EmperorPenguin.SangmyungBank.api.users.add.domain.User.User; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import javax.servlet.http.HttpSession;   @RequiredArgsConstructor @RequestMapping("/users") @RestController public class UserLoginController {   private final UserLoginService userLoginService;  private final SessionService sessionService;    @RequestMapping (path="/login",method={RequestMethod.*GET*,RequestMethod.*POST*})  public ResponseEntity<User> authUser(@RequestBody LoginForm loginForm,HttpServletResponse response)  {  User auth = userLoginService.authenticate(loginForm);  if (auth == null) {  return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).body(auth);  }  else {  sessionService.CreateSession(auth, response);  return ResponseEntity.*status*(HttpStatus.*OK*)  .body(auth);  }   } // @PostMapping("/logout") // public String logoutV2(HttpServletRequest request) { // sessionManager.expire(request); // return "redirect:/"; // }  -SessionService  package com.EmperorPenguin.SangmyungBank.api.users.login.service;  import org.springframework.http.HttpStatus; import org.springframework.stereotype.Component;  import javax.servlet.http.Cookie; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import java.net.http.HttpResponse; import java.util.Arrays;  import java.util.Map; import java.util.UUID; import java.util.concurrent.ConcurrentHashMap;  @Component public class SessionService {   public static final String *SESSION\_COOKIE\_Name* = "\_sid";   private Map<String,Object> sessionStore = new ConcurrentHashMap<>();   public void CreateSession(Object value, HttpServletResponse response){  String sessionId = "SB" + UUID.*randomUUID*().toString();  sessionStore.put(sessionId,value);  Cookie SessionCookieId = new Cookie(*SESSION\_COOKIE\_Name*, sessionId);  response.addCookie(SessionCookieId);  response.setStatus(HttpServletResponse.*SC\_OK*);  }  /\*  \* 세션 조회  \*/  public Object getSession(HttpServletRequest request) {  Cookie sessionCookieId = findCookie(request, *SESSION\_COOKIE\_Name*);  if (sessionCookieId == null ) {  return null;  }  return sessionStore.get(sessionCookieId.getValue());  }  /\*  \* 세션 만료  \*/  public void expire(HttpServletRequest request) {  Cookie sessionCookie = findCookie(request, *SESSION\_COOKIE\_Name*);  if (sessionCookie != null) {  sessionStore.remove(sessionCookie.getValue());  }  }   private Cookie findCookie(HttpServletRequest request, String cookieName) {  if (request.getCookies() == null) {  return null;  }  return Arrays.*stream*(request.getCookies())  .filter(cookie -> cookie.getName().equals(cookieName))  .findAny()  .orElse(null);  } }  -PostMan  1.Correct login    2.incorrect login  **진행 예정** |  |
| * **특이사항 / 협업 사항** | |