

## 16. 애플리케이션 설계

수행평가

일시 : 2023.03.23.

이름 : 조주영

### 1. 프로필 페이지 화면 (profile/index)

### 2. 화면 설명

- 사용자의 아이디, 정보, 팔로워 목록, 게시물 등을 볼 수 있는 화면이다.

### 3. 화면 항목 설명

항목명	영문명	설 명	속성	Validation		
				필수여부	자릿수	기타
프로필 사진	profileImg	각 사용자의 프로필 사진이 암호화되어 부여되는 저장명이 입력되는 변수	E	선택	255	
아이디	username	로그인 시 이용하는 개인의 고유 id	R	필수	25	
게시물 갯수	post	현재 프로필 화면에 해당하는 사용자가 게시한 게시물의 갯수	E	선택		
팔로워	myFollower	현재 프로필 화면에 해당하는 사용자를 팔로우하는 사용자의 수	E	선택		
팔로잉	myFollowing	현재 프로필 화면에 해당하는 사용자가 팔로우하는 사용자의 수	E	선택		
소개	profileText	각 사용자가 입력한 프로필 소개	R	선택		
게시물 번호	no	각 게시물에 부여된 고유의 번호	R	필수	11	
썸네일	firstThumb	각 게시물에 업로드된 사진 중 첫 번째 순서, 즉 각 게시물의 썸네일	R	필수	225	
좋아요	like	각 게시물의 좋아요 갯수	R	선택		
댓글 수	commentCount	각 게시물에 달린 댓글 갯수	R	선택		
내 아이디	myName	현재 프로필 화면을 보고 있는 로그인 사용자의 id	H	필수	25	
게시물	article	게시물 목록	R	선택		

#### 4. 화면 이벤트 설명

이벤트 명	이벤트 설명	관련 데이터		관련 Object
화면로딩 (onLoad)	프로필 화면을 표시한다.	S		
		R	profileImg, username, myName, post, myFollower, myFollowing, profileText, posts (no, firstThumb, like, commentCount)	ProfileController - index (GetMapping)
팔로우 버튼 (onClick)	해당 사용자를 보고 있는 사용자가 팔로우한다.	S	userName, myName	ProfileController - follow (PostMapping)
		R		
팔로잉 버튼 (onClick)	현재 프로필 화면의 사용자에게 대한 추가 작업 (친한 친구, 언팔로우) 요청을 할 수 있는 모달창을 띄운다.	S		
		R		
팔로잉 취소 버튼 (onClick)	현재 프로필 화면의 사용자를 보고 있는 사용자가 언팔로우한다.	S	userName, myName	ProfileController - follow (PostMapping)
		R		
프로필 사진 수정 버튼 (onClick)	현재 프로필 화면의 사용자와 보고 있는 사용자가 일치할 때, 프로필 사진을 수정한다.	S	profileImg	ProfileController - upload (PostMapping)
		R	profileImg	
프로필 사진 삭제 버튼 (onClick)	현재 프로필 화면의 사용자와 보고 있는 사용자가 일치할 때, 프로필 사진을 삭제한다.	S	profileImg	ProfileController - upload (GetMapping)
		R		
프로필 변경 (onClick)	현재 프로필 화면의 사용자와 보고 있는 사용자가 일치할 때, 프로필 정보를 수정한다.	S		
		R		profile / modify (페이지 이동)
메시지 보내기 (onClick)	현재 프로필 화면의 사용자에게 보고 있는 사용자가 메시지를 보낸다.	S	userName, myName	ChatController - goChattingRoom (PostMapping) chat / content (페이지 이동)
		R		

## 5. 관련 파일

파일유형	경로명	파일명
html	/templates/profile	index.html
java	/controller	ProfileController.java
java	/dao	ProfileDAO.java
java	/service	ProfileService.java
java	/vo	MemberVO.java
java	/vo	PostVO.java
java	/vo	ImageVO.java
xml	/mappers	profile.xml
css	/static/css	profile.css
js	/static/js	myFollow.js
js	/static/js	profileModal.js
js	/static/js	profileUpload.js

## 1. 프로필 정보 변경 화면 (profile/modify)

### 2. 화면 설명

- 사용자의 아이디, 정보, 팔로워 목록, 게시물 등을 볼 수 있는 화면이다.

### 3. 화면 항목 설명

항목명	영문명	설 명	속성	Validation		
				필수여부	자릿수	기타
프로필 사진	profileImg	각 사용자의 프로필 사진이 암호화되어 부여되는 저장명이 입력되는 변수	E	선택	255	
사용자 이름	username	로그인 시 이용하는 개인의 고유 id	E	필수	25	
이름	name	사용자의 이름	E	필수	20	
이메일	email	사용자의 이메일	E	필수	45	
전화번호	profilePhone	사용자의 전화번호	E	선택	13	
소개	profileText	사용자 소개	R	선택		
성별	gender	사용자 성별	R	선택	2	

### 4. 화면 이벤트 설명

이벤트 명	이벤트 설명	관련 데이터		관련 Object
화면로딩 (onLoad)	프로필 변경 화면을 표시한다.	S		
		R	profileImg, username, name, email, profilePhone, profileText, gender	ProfileController - modify (GetMapping)
제출 버튼 (onClick)	해당 사용자의 프로필 정보를 변경한다.	S	username, name, email, profilePhone, profileText, gender	ProfileController - modify (PostMapping)
		R		
프로필 사진 변경 버튼 (onClick)	프로필 사진을 수정한다.	S	profileImg	ProfileController - upload (PostMapping)
		R	profileImg	
프로필 사진 삭제 버튼 (onClick)	프로필 사진을 삭제한다.	S	profileImg	ProfileController - upload (GetMapping)
		R		

## 5. 관련 파일

파일유형	경로명	파일명
html	/templates/profile	modify.html
java	/controller	ProfileController.java
java	/dao	ProfileDAO.java
java	/service	ProfileService.java
java	/vo	MemberVO.java
java	/vo	ImageVO.java
xml	/mappers	profile.xml
css	/static/css	profileModify.css
js	/static/js	profileModal.js
js	/static/js	profileUpload.js

## 1. 클래스명

항목명	ProfileController		PACKAGE	kr.co.photostagram.controller	
EXTENDS	N/A				
IMPLEMENTS	N/A				
IMPORT	kr.co.photostagram.service.MainService; kr.co.photostagram.service.ProfileService; kr.co.photostagram.utils.JSFunction; kr.co.photostagram.vo.MemberVO; kr.co.photostagram.vo.NoticeVO; kr.co.photostagram.vo.PostVO; kr.co.photostagram.vo.SearchListVO; lombok.extern.slf4j.Slf4j; org.springframework.beans.factory.annotation.Autowired; org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder; org.springframework.stereotype.Controller; org.springframework.ui.Model; org.springframework.web.bind.annotation.GetMapping; org.springframework.web.bind.annotation.PostMapping; org.springframework.web.bind.annotation.RequestParam; org.springframework.web.bind.annotation.ResponseBody; org.springframework.web.multipart.MultipartFile; javax.servlet.http.HttpServletResponse; javax.transaction.Transactional; java.security.Principal; java.util.*;				
CLASS설명	profile에 속한 모든 페이지의 기능을 담당하는 CLASS				
[속성]					
Name	Visibility	Type		Default Value	설명

## 2. 메소드 명

### 1) modify @GetMapping

Name	modify
Visibility	public
Parameter	Principal principal, Model model
Return	String
설명	Spring Security 내 사용자 정보를 불러온 다음 프로필 변경 화면으로 넘긴다.

## 2) index @GetMapping

Name	index
Visibility	public
Parameter	Principal principal, Model model, String username
Return	String
설명	<p>Spring Security 내 사용자 정보, 그리고 username 값을 이용해 각 사용자 VO를 불러온다. (service.selectMember 이용, 각각 member와 user로 명명한다.) member와 user의 고유 번호를 호출해 myNo, pageNo로 명명한다. pageNo를 이용해 사용자 게시물 목록을 불러온다. (service.selectPosts)</p> <p>pageNo와 게시물 번호를 이용해 각 게시물의 썸네일을 불러온다. (service.selectThumb) 최신 순으로 정렬하여 화면에 넘길 준비를 한다.</p> <p>사용자의 게시물, 팔로워, 팔로잉 수를 불러온다. (service.selectCountPost / service.selectCountFollower / service.selectCountFollowing)</p> <p>팔로워, 팔로잉 목록을 불러온다. (service.selectFollowers / service.selectFollowings)</p> <p>팔로잉, 팔로워 목록의 유저들이 프로필 페이지를 보고 있는 사용자와 팔로우, 팔로잉 중인지 검색을 수행한다. (service.selectMember -&gt; service.searchFollowing)</p> <p>aside에서 이용할 검색 기록을 요청한다. (mainService.selectSearchItemRecent) aside에서 이용할 알림을 불러온다. (mainService.selectNotices)</p>

## 3) post @PostMapping

Name	post
Visibility	public
Parameter	String username, int pg
Return	Map <Integer, PostVO>
설명	<p>프로필 페이지 사용자 정보를 불러온다. (service.selectMember) 사용자 게시물을 원하는 만큼 불러와 최신순으로 정렬한다. (service.selectPosts -&gt; service.selectThumb -&gt; data.put)</p>



#### 4) modify @PostMapping

Name	modify
Visibility	public
Parameter	HttpServletResponse resp, MemberVO vo
Return	void
설명	성별 입력이 유효성 검사에서 탈락할 경우 null 처리를 한다. (vo.setGender(null)) service.updateMember 후 alert창을 띄우고 modify로 돌아온다.

#### 5) changePass @GetMapping

Name	changePass
Visibility	public
Parameter	Principal principal, Model model
Return	String
설명	Spring security로 멤버 정보를 불러온 후 비밀번호 변경 페이지를 띄운다.

#### 6) changePass @PostMapping

Name	changePass
Visibility	public
Parameter	HttpServletResponse resp, Principal principal, String prePass, String pass1, String pass2
Return	void
설명	if Spring security 방식의 password로 받아온 prePass를 변환하여 현재 비밀번호와 이전 비밀번호가 일치하는지 확인 if pass1과 pass2가 일치하는지 확인 확인하여 일치할 경우 변경 수행. (service.updatePassword) if 비밀번호 변경에 성공했을 경우 로그아웃 처리 else - 일치하지 않을 경우 유효성에서 탈락

#### 7) deleteUser @GetMapping

Name	deleteUser
Visibility	public
Parameter	Principal principal, Modal modal
Return	String
설명	Spring security로 멤버 정보를 불러온 후 탈퇴 페이지를 띄운다.

## 8) deleteUser @PostMapping

Name	deleteUser
Visibility	public
Parameter	HttpServletResponse resp, Principal principal, int no, String password
Return	void
설명	if - 입력한 비밀번호가 비밀번호와 같은 경우 service.deleteMember (멤버 정보를 삭제) service.deleteMemberData (멤버 활동 기록을 삭제) 로그아웃되어 초기 화면으로 이동. else - 비밀번호를 다시 확인해달라는 알림창

## 9) upload @GetMapping

Name	upload
Visibility	public
Parameter	Principal principal, String type
Return	Map <String, Integer>
설명	member의 번호와 profileImg를 불러온 후, (service.selectMember -> .getNo(), .getProfileImg()) service.uploadProfilePhoto로 프로필 사진 이름을 DB에서 삭제 service.deleteProfilePhotoFile로 사진 파일을 서버에서 삭제

## 10) upload @PostMapping

Name	upload
Visibility	public
Parameter	@RequestParam("file") MultipartFile file, Principal principal
Return	Map <String, Integer>
설명	member 번호를 불러온 후, (service.selectMember.getNo()) if - 이전 사진이 있는 경우 DB에서 사진 파일을 삭제한다. service.deleteProfilePhotoFile 이후 새 사진을 업로드한다. service.uploadProfilePhoto

## 11) follow @GetMapping

Name	follow
Visibility	public
Parameter	String type, String userName, String myName
Return	String
설명	userName과 myName을 이용해 사용자 번호를 추출한 후 if - 받은 type이 "insert"인 경우 (service.insertFollowing) else - 받은 type이 "delete"인 경우 (service.deleteFollowing)

12) follow @PostMapping

Name	upload
Visibility	public
Parameter	Principal principal, @RequestParam("type") String type, @RequestParam("userName") String userName
Return	Map <String, Integer>
설명	<p>userName과 myName을 이용해 사용자 번호를 추출한 후</p> <p>if - 받은 type이 "insert"인 경우 (service.insertFollowing)</p> <p>else if - 받은 type이 "delete"인 경우 (service.deleteFollowing)</p> <p>else if - 받은 type이 "myDelete"인 경우 (service.deleteFollowing)</p> <p>myDelete의 경우는 내가 팔로우의 객체인 경우이다. 내 팔로워를 삭제.</p>

## 1. 클래스 명

항목명	ProfileService		PACKAGE	kr.co.photostagram.service	
EXTENDS	N/A				
IMPLEMENTS	N/A				
IMPORT	import kr.co.photostagram.dao.ProfileDAO; import kr.co.photostagram.vo.MemberVO; import kr.co.photostagram.vo.PostVO; import lombok.extern.slf4j.Slf4j; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.beans.factory.annotation.Value; import org.springframework.stereotype.Service; import org.springframework.web.multipart.MultipartFile; import java.io.File; import java.io.IOException; import java.util.List; import java.util.UUID;				
CLASS설명	profile에 관련된 모든 서비스를 담당하는 CLASS				
[속성]					
Name	Visibility	Type	Default Value	설명	
uploadPath	private	String		파일이 upload되는 경로	

## 2. 메소드 명

### 1) selectMember

Name	selectMember
Visibility	public
Parameter	String username
Return	MemberVO
설명	멤버 한 명의 정보를 불러온다.

### 2) updateMember

Name	updateUser
Visibility	public
Parameter	MemberVO vo
Return	int
설명	멤버 정보를 수정한다.

### 3) deleteMember

Name	deleteMember
Visibility	public
Parameter	int no
Return	int
설명	멤버 정보를 모두 삭제한다.

### 4) deleteMemberData

Name	deleteMemberData
Visibility	public
Parameter	int no
Return	int
설명	특정 멤버가 관련된 모든 데이터를 삭제한다.

### 5) updatePassword

Name	updatePassword
Visibility	public
Parameter	String newPass, int no
Return	int
설명	비밀번호를 변경한다.

### 6) selectPosts

Name	selectPosts
Visibility	public
Parameter	int no, int pg
Return	List <PostVO>
설명	프로필 페이지에 띄울 게시물을 일정 수만큼 불러온다.

### 7) selectThumb

Name	selectThumb
Visibility	public
Parameter	int pageNo, int postNo
Return	PostVO
설명	특정 게시물의 첫 번째 이미지 (썸네일)를 불러온다.

## 8) selectFollowers

Name	selectFollower
Visibility	public
Parameter	int no
Return	List <MemberVO>
설명	특정 멤버를 팔로우하는 멤버들을 불러온다.

## 9) selectFollowings

Name	selectFollowings
Visibility	public
Parameter	int no
Return	List <MemberVO>
설명	특정 멤버가 팔로우하는 멤버들을 불러온다.

## 10) selectCountPost

Name	selectCountPost
Visibility	public
Parameter	int no
Return	int
설명	특정 멤버의 게시물 수를 센다.

## 11) selectCountFollower

Name	selectCountFollower
Visibility	public
Parameter	int no
Return	int
설명	특정 멤버를 팔로우하는 멤버의 수를 센다.

## 12) selectCountFollowing

Name	selectCountFollowing
Visibility	public
Parameter	int no
Return	int
설명	특정 멤버가 팔로우하는 멤버의 수를 센다.

## 13) insertFollowing

Name	insertFollowing
Visibility	public
Parameter	int follower, int following
Return	int
설명	특정 멤버가 특정 멤버를 팔로우한다.

## 14) deleteFollowing

Name	deleteFollowing
Visibility	public
Parameter	int follower, int following
Return	int
설명	특정 멤버가 특정 멤버를 언팔로우한다.

## 15) searchFollowing

Name	searchFollowing
Visibility	public
Parameter	int follower, int following
Return	int
설명	특정 멤버가 특정 멤버를 팔로우하는지 검색한다.

## 16) uploadProfilePhoto

Name	uploadProfilePhoto
Visibility	public
Parameter	String type, MultipartFile file, int no
Return	int
설명	if - type이 delete라면 dao.deleteProfilePhoto 수행. (DB 삭제) else - 경로를 구해 UUID로 파일 명을 암호화, dao.updateProfilePhoto 수행.

## 17) deleteProfilePhotoFile

Name	deleteProfilePhotoFile
Visibility	public
Parameter	String name
Return	void
설명	이전 프로필 사진이 있을 경우 file.delete() (서버 삭제)