

SPRING BOOT

학원관리 웹 포트폴리오

소스코드 : <https://github.com/casatoo/EducationManager>

차례

1. 개발 동기
2. 개발 환경
3. 주요 기능
4. 동작 과정
5. DB 설계
6. 각 기능 소개
7. 추가 될 기능

01

개발 동기

학원에서 사용할 수 있는 학생 관리 프로그램입니다.

02

개발 환경

개발 툴

- STS-4.15.2
- git version 2.35.0.windows.1
- XAMPP Mysql version 10.4.24.MariaDB
- SQLyog community 64
- Maven 4.0.0- chrome
- codepen

기술 스택

- jdbc4 version 1.16
- jdk version 15.0.2
- jquery 3.6.1
- java version 17
- jstl 1.2
- lombok version 1.18.24
- fullcalendar
- tailwind
- ajax
- javax.mail version 1.6.2
- font-awesome 6.2.0
- UNICONS 4.0.0
- TOAST UI editor

프레임 워크

- spring boot starter web 2.7.5
- mybatis version 2.2.2

03

주요 기능

1. 회원 관리 기능

- 회원가입
- 로그인
- 로그아웃
- 회원정보 수정
- 회원탈퇴
- 아이디 찾기
- 비밀번호 찾기
- 인증키 생성 이메일 전송
- 프로필 사진 업로드
- ~~구글, 네이버로 로그인~~
- 개인정보 조회

2. 멀티 게시판 기능

- 공지사항
- 자유게시판
- ~~갤러리~~
- CRUD 기능 구현
- CRUD 권한 부여
- 키워드 검색 기능
- ~~첨부파일 등록~~
- 사진 업로드 기능
- 댓글 작성 기능
- 추천 기능

3. 교육일정 관리 기능

- 캘린더 API
- 날씨 API
- 월간 일정 CRUD
- 일별 일정 CRUD
- 교육일정 조회
- ~~- 출석체크 기능~~

5. 테이블 생성 출력 기능

- ~~- 개인별 기록 테이블 생성~~
- ~~- pdf 출력~~

6. 관리자 권한

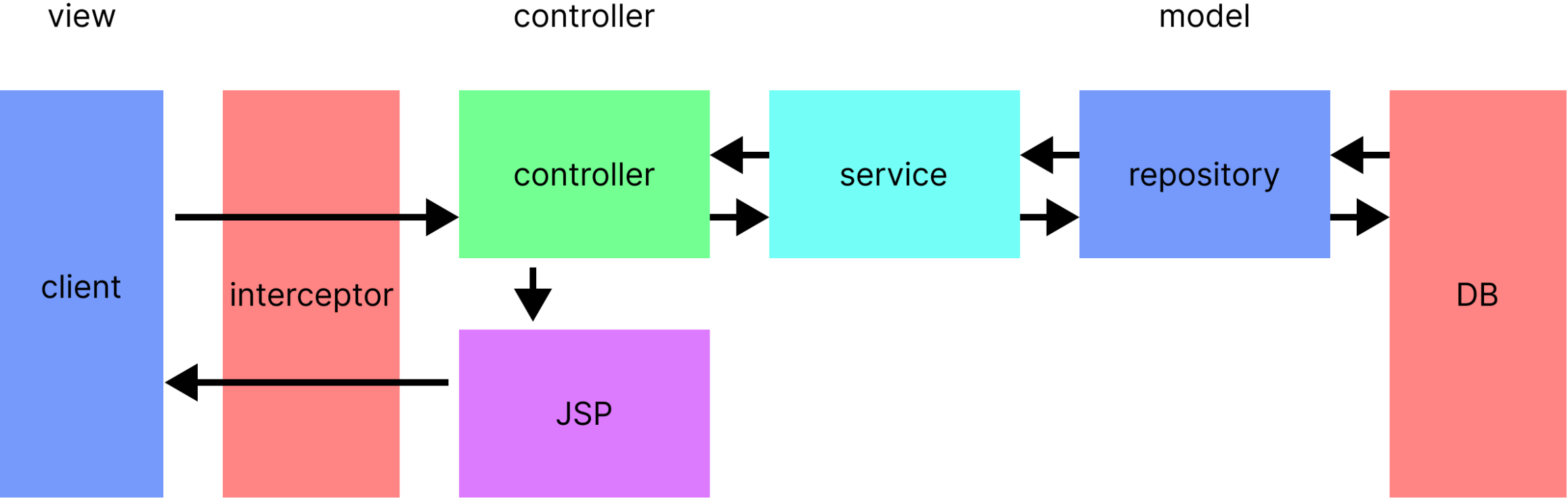
- 회원조회
- 공지사항 권한 부여
- 교육일정 관리 권한

7. 세팅

- 홈페이지 테마 설정

04

동작 과정

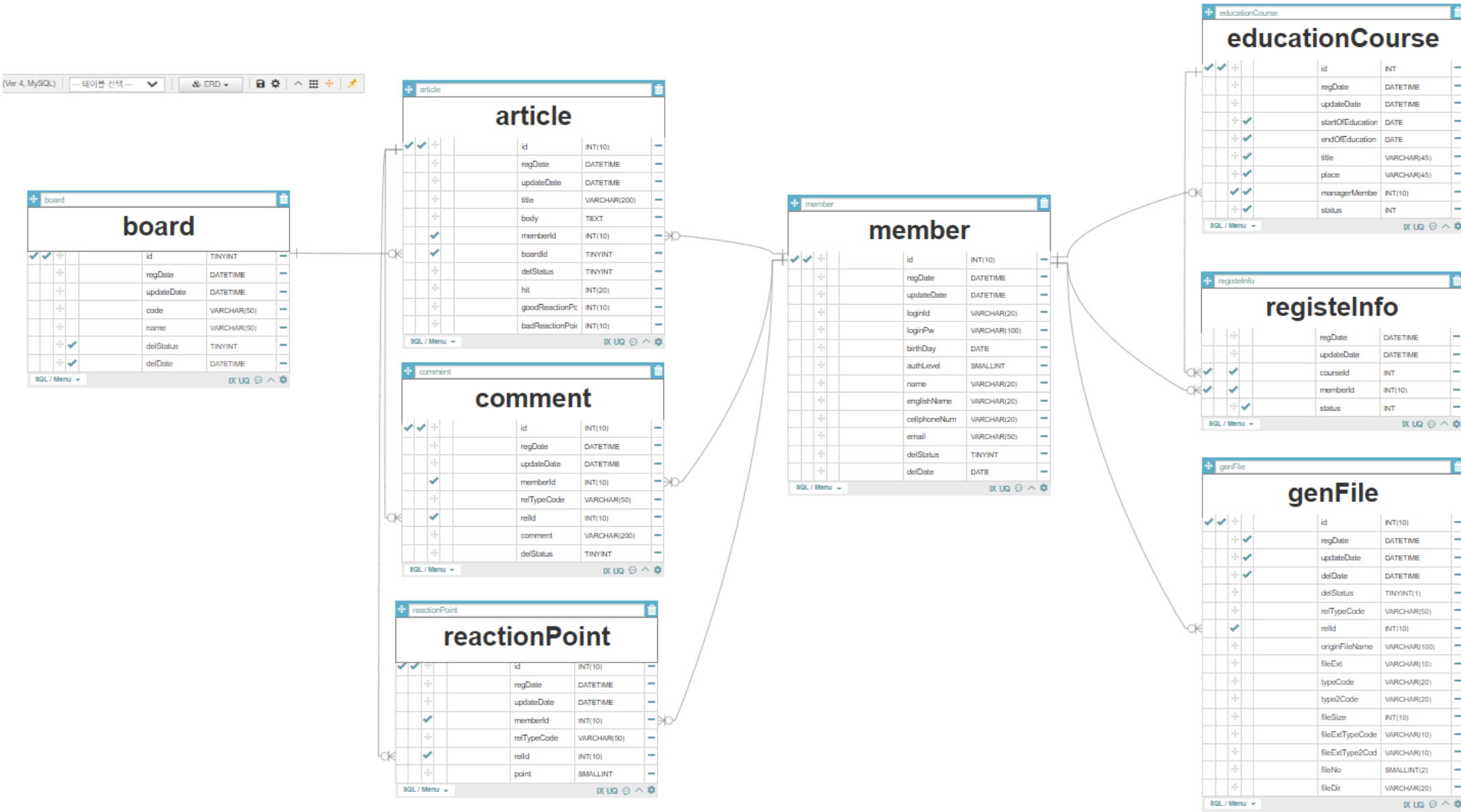


1. client가 controller로 요청을 보낼때 interceptor 에서 rq 객체 불러오기, 로그인에 필요한 요청, 관리자 권한이 필요한 요청에 대해 먼저 작동합니다.
2. interceptor 를 거쳐 도달한 요청은 controller 에서 접수되어 실행되거나 jsp 파일에 필요한 정보를 전달합니다.
3. controller 에서 요청한 데이터는 내부 코드를 숨기기 위해 service단에서 ResultData 형식으로 전달됩니다.
4. service 에서 필요한 데이터는 repository로 요청을 보내어 repository만 데이터베이스에 접근할 수 있도록 합니다.
5. repository는 myBatis 를 사용해 XML 파일과 맵핑되어 값을 주고 받습니다.
6. 계층간 데이터 교환은 VO에서 lombok 을 사용한 getter, setter, 생성자를 만들어줍니다.

05

DB 설계

04 동작 과정



06

각 기능 소개

05 각 기능 소개

메인페이지 (로그아웃상태)

EduManager

- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정

로그인

Dark Mode



로그아웃 중

교육과정 바로가기

22-6기

2022-11-01~2022-11-30

담당자:직원

진행중

22-5기

2022-09-01~2022-09-30

담당자:직원

진행중

22-4기

2022-07-01~2022-07-30

담당자:관리자

진행중

최근 공지사항

제목	작성자	조회수
공지사항4	직원	0
공지사항3	관리자	1
공지사항2	직원	0
공지사항1	관리자	0

날씨

맑음 4 °C

북동풍

3.6 m/s

05 각 기능 소개

UsrHomeController

```
@Controller
public class UsrHomeController {
    @Autowired
    private EducationCourseService educationCourseService;
    @Autowired
    private ArticleService articleService;
    @Autowired
    private Rq rq;

    @RequestMapping("usr/home/main")
    public String showMain(Model model) {

        /**
         * 최근 작성된 부터 3개 교육과정 가져옴
         */
        List<EducationCourse> educationCourses = educationCourseService.getEducationCourseSomeList(0,3);
        /**
         * 최근 작성된 공지사항 게시물 6개 가져옴
         */
        List<Article> articles = articleService.getForPrintArticles(rq.getLoginedMemberId(), 1, 0, 6, "", "");

        model.addAttribute("educationCourses",educationCourses);
        model.addAttribute("articles",articles);
        return "usr/home/main";
    }

    @RequestMapping("/")
    public String showRoot() {
        return "redirect:usr/home/main";
    }
}
```


05 각 기능 소개

날씨정보 조회

기상청 단기 예보 조회 서비스 오픈 API 사용

필수 요소

- 위도, 경도
- 날짜
- 시간

```
/** 현재 년월일 구하는 함수 */
function getCurrentDate()
{
    var date = new Date();
    var nowDate = new Date();
    var hours = date.getHours();
    var minutes = date.getMinutes();
    var yesterday = new Date(date.setDate(date.getDate() - 1));

    if(hours < 2){
        var year = yesterday.getFullYear().toString();
        var month = yesterday.getMonth() + 1;
        month = month < 10 ? '0' + month.toString() : month.toString();

        var day = yesterday.getDate();
        day = day < 10 ? '0' + day.toString() : day.toString();

        return year + month + day ;
    }
    if(hours = 2 && minutes<10){
        var year = yesterday.getFullYear().toString();
        var month = yesterday.getMonth() + 1;
        month = month < 10 ? '0' + month.toString() : month.toString();

        var day = yesterday.getDate();
        day = day < 10 ? '0' + day.toString() : day.toString();

        return year + month + day ;
    }

    var year = nowDate.getFullYear().toString();
    var month = nowDate.getMonth() + 1;
    month = month < 10 ? '0' + month.toString() : month.toString();

    var day = nowDate.getDate();
    day = day < 10 ? '0' + day.toString() : day.toString();

    return year + month + day ;
}
```

05 각 기능 소개

```
/** 현재 시간을 3시간 단위로 표현 */
function getCurrentTime()
{
    var date = new Date();
    var hours = date.getHours();
    var minutes = date.getMinutes();
    var currentTime = '2300';
    switch (hours){
        case 2 :
            if(minutes < 10){
                break;
            }
        case 3 :
        case 4 :
            currentTime = '0200';
            break;
        case 5 :
            if(minutes < 10){
                currentTime = '0200';
            }
        case 6 :
        case 7 :
            currentTime = '0500';
            break;
        case 8 :
            if(minutes < 10){
                currentTime = '0500';
            }
        case 9 :
        case 10 :
            currentTime = '0800';
            break;
        case 11 :
            if(minutes < 10){
                currentTime = '0800';
            }
        case 12 :
        case 13 :
            currentTime = '1100';
            break;
        case 14 :
```

```
        case 14 :
            if(minutes < 10){
                currentTime = '1100';
            }
        case 15 :
        case 16 :
            currentTime = '1400';
            break;
        case 17 :
            if(minutes < 10){
                currentTime = '1400';
            }
        case 18 :
        case 19 :
            currentTime = '1700';
            break;
        case 20 :
            if(minutes < 10){
                currentTime = '1700';
            }
        case 21 :
        case 22 :
            currentTime = '2000';
            break;
        case 23 :
            if(minutes < 10){
                currentTime = '2000';
            }
            break;
    }
}
```

05 각 기능 소개

```
const API_KEY = 'UDknKWMcEt2j1I10azmpJzieEdhjjaSOMbGQwmF0nEXiBUE2QKGJosC8yLqyllGhvAVgbU3JrtQDakwPQYfE4w%3D%3D';
async function getData() {

  var base_date = getCurrentDate();
  var base_time = getCurrentTime();
  const url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getVilageFcst?serviceKey='+API_KEY+'&numOfRows=10&pageNo=1&base_date='+base_date+'&base_time='+base_time+'&nx=36&ny=127&dataType=JSON';
  const response = await fetch(url);
  const data = await response.json();
  var pty = 0;
  for(var weather of data.response.body.items.item){
    switch(weather.category){
      case 'PTY' :
        switch (weather.fcstValue){
          case '1':
            $('<pty>').empty().append('');
            $('<weatherMsg>').text('비');
            pty = 1;
          case '2':
            break;
          case '3':
            $('<pty>').empty().append('');
            $('<weatherMsg>').text('눈');
            pty = 1;
            break;
          case '4':
            $('<pty>').empty().append('');
            $('<weatherMsg>').text('진눈깨비');
            pty = 1;
            break;
        }
        break;
      case 'SKY' :
        if(pty != 0){
          break;
        }else{
          switch (weather.fcstValue){
            case 6:
            case 7:
            case 8:
              $('<pty>').empty().append('');
            }
        }
    }
  }
}
```

05 각 기능 소개

```
    }  
    }  
    $('.tmp').html(weather.fcstValue);  
    break;  
case 'TMP' :  
    $('.tmp').html(weather.fcstValue);  
    break;  
case 'POP' :  
    $('.pop').html(weather.fcstValue);  
    break;  
case 'VEC' :  
    const vec = parseInt(weather.fcstValue);  
    if(vec >= 337 || vec <= 22){  
        $('.vec').text('북풍');  
    }else if( 23 <= vec <= 67){  
        $('.vec').text('북동풍');  
    }  
    else if( 68 <= vec <= 112){  
        $('.vec').text('동풍');  
    }  
    else if( 113 <= vec <= 157){  
        $('.vec').text('남동풍');  
    }  
    else if( 158 <= vec <= 202){  
        $('.vec').text('남풍');  
    }  
    else if( 203 <= vec <= 247){  
        $('.vec').text('남서풍');  
    }  
    else if( 248 <= vec <= 292){  
        $('.vec').text('서풍');  
    }  
    else if( 293 <= vec <= 336){  
        $('.vec').text('북서풍');  
    }  
    break;  
case 'WSD' :  
    $('.wsd').html(weather.fcstValue);  
    break;  
}
```



날씨



맑음 4 °C

북동풍

3.6 m/s

로그인 화면

→) 로그인

아이디

아이디를 입력해주세요

비밀번호

비밀번호를 입력해주세요

→) 로그인

👤+ 회원가입

👤🔍 아이디 찾기

🔒 비밀번호 찾기

127.0.0.1:8081의 메시지
아이디를 입력해주세요

확인

127.0.0.1:8081의 메시지
비밀번호를 입력해주세요

확인

127.0.0.1:8081 내용:
직원 회원님 환영합니다.

확인

05 각 기능 소개

memberController

중복 로그인 방지

아이디 입력 체크

비밀번호 입력 체크

서비스 단에 요청하여 보고서 형식으로 받아 결과를 출력.

(아이디 또는 비밀번호 불일치 여부)

afterLoginUri 는 로그인 페이지 접속 전 주소를 기억하여 로그인 후 접속 전 페이지로 돌아가도록 함.

```
* 비밀번호 입력 체크
* 로그인 성공하면 이름 검색해서 메시지 출력
* rq에 로그인 정보 전달
* @param loginId
* @param loginPw
* @param afterLoginUri
* @return String
*/
@RequestMapping("/usr/member/doLogin")
@ResponseBody
public String doLogin(String loginId, String loginPw, @RequestParam(defaultValue = "/")String afterLoginUri) {

    if(rq.isLogged()){
        return Ut.jsReplace(Ut.f("이미 로그인중입니다"),Ut.getUriEncoded(afterLoginUri));
    }
    if(Ut.empty(loginId)) {
        return Ut.jsHistoryBack(Ut.f("아이디를 입력해주세요"));
    }
    if(Ut.empty(loginPw)) {
        return Ut.jsHistoryBack(Ut.f("비밀번호를 입력해주세요"));
    }
    ResultData doLoginRd= memberService.doLogin(loginId,loginPw);

    if(doLoginRd.isFail()) {
        ResultData resultRd = ResultData.from(doLoginRd.getResultCode(),doLoginRd.getMsg());
        return Ut.jsReplace(resultRd.getMsg(), Ut.f("../member/login?afterLoginUri=%s",afterLoginUri));
    }
    Member member = memberService.getMemberByLoginId(loginId);
    rq.login(member);

    return Ut.jsReplace(Ut.f("%s 회원님 환영합니다.",member.getName()),afterLoginUri);
}

@RequestMapping("/usr/member/login")
public String loginForm(String afterLoginUri, HttpServletRequest req, Model model) {

    model.addAttribute("afterLoginUri",afterLoginUri);
    return "/usr/member/login";
}
```

05 각 기능 소개

MemberService

아이디 일치여부 확인

탈퇴한 회원 여부 확인

비밀번호 체크

(입력된 pw 를 sha-256으로 암호화)

실패, 성공 메시지 반환

```
/**
 * 아이디 체크
 * 탈퇴여부 체크
 * 비밀번호 체크
 * 후 로그인
 * @param loginId
 * @param loginPw
 * @return ResultData
 */
public ResultData<Integer> doLogin(String loginId, String loginPw) {

    Member existsMember = memberRepository.getMemberByLoginId(loginId);

    if(existsMember == null) {
        return ResultData.from("F-4", Ut.f("아이디가 틀렸습니다."));
    }
    if(existsMember.getDelStatus() == 1) {
        return ResultData.from("F-5", Ut.f("탈퇴한 회원입니다."));
    }

    if(!existsMember.getLoginPw().equals(Ut.sha256(loginPw))) {
        return ResultData.from("F-6", Ut.f("비밀번호가 틀렸습니다."));
    }
    return ResultData.from("S-1", Ut.f("로그인 성공"));
}
```

05 각 기능 소개

MemberRepository

loginId 조회해서 member 객체 반환

```
<select id="getMemberByLoginId" resultType="Member">
    SELECT *
    FROM
    `member`
    WHERE loginId
    = #{loginId}
</select>
```

Member/login.jsp

중복 로그인 방지 스크립트

```
<script>
    <!-- 로그인 중 로그인 페이지 접속 제한 -->
    if(${rq.isLoggedin()}){
        alert("이미 로그인중입니다.");
        history.back();
    }
</script>
```

VO/rq

세션에 로그인 정보 저장

```
/**
 * 로그인처리
 * 세션에 변수 저장 key : member
 * @param member
 */
public void login(Member member) {
    session.setAttribute("loggedMemberId", member.getId());
}
```


05 각 기능 소개

비밀번호를 입력해주세요

아이디 찾기

이름

관리자

이메일

123123@asd.com

찾기

닫기

일치하는 회원이 없습니다.

비밀번호를 입력해주세요

아이디 찾기

이름

관리자

이메일

casato6666@gmail.com

찾기

닫기

아이디 조회 성공:admin

05 각 기능 소개

Member/login.jsp


비밀번호 찾기

아이디, 이름, 이메일을 입력하면

임시 패스워드를 조합하여 메일로 발송

[EM] 임시 패스워드 발송

받은편지함 x



EM <mliukn@gmail.com>

나에게 ▾

임시 패스워드 : s8yb8q

[로그인 하러가기](#)

← 답장

→ 전달

비밀번호를 입력해주세요

비밀번호 찾기

아이디

admin

이름

관리자

이메일

casato6666@gmail.com

찾기

닫기

127.0.0.1:8081의 메시지

계정의 이메일주소로 임시 패스워드가 발송되었습니다.

확인

05 각 기능 소개

Member/login.jsp

아이디 찾기, 비밀번호 찾기 모달창 작동 함수

아이디 찾기 비동기 처리

```
<!-- 아이디 찾기, 비밀번호 찾기 모달 창 -->
<script>

const findLoginIdModal = () =>{
  $('.findLoginId-modal-bg').removeClass('hidden');
}
const findLoginIdCancleBtn = () =>{
  $('.findLoginId-modal-bg').addClass('hidden');
}
const findPasswordModal = () =>{
  $('.findPassword-modal-bg').removeClass('hidden');
}
const findPasswordCancleBtn = () =>{
  $('.findPassword-modal-bg').addClass('hidden');
}
</script>
<!-- 아이디 찾기 비동기 처리 -->
<script>
const findLoginId = (form) =>{
  $.get('../member/findLoginId', {
    name : form.name.value,
    email : form.email.value,
    ajaxMode : 'Y'
  }, function(data) {
    if(data.resultCode == "F-1"){
      $('.findLoginIdMessage').css('color','red').text(data.msg);
    }
    if(data.resultCode == "S-1"){
      const msg = data.msg + ':' + data.data1;
      $('.findLoginIdMessage').css('color','green').text(msg);
    }
  }, 'json');
}
</script>
```

Member/login.jsp

모달창 외부 클릭 시 모달창이 종료되는 기능 구현

onclick="event.stopPropagation()" (자식요소 클릭시 부모요소가 동작하지 않도록 함)

```
<div class="findLoginId-modal-bg hidden" onclick="findLoginIdCancleBtn()">
  <div class="findLoginId-modal" onclick="event.stopPropagation()">
    <section class="flex justify-center mt-14">
      <div class="w-full w-96">
```

05 각 기능 소개

memberController

아이디 , 비밀번호 찾기

아이디, 이름, 이메일 일치 확인

모두 일치할 경우

이메일로 임시 비밀번호 전송하고
데이터베이스를 임시비밀번호로 변경

```

    @RequestMapping("/usr/member/findLoginId")
    @ResponseBody
    public ResultData findLoginId(String name , String email) {

        ResultData rd = memberService.findLoginId(name,email);

        return rd;
    }
    /**
     * 비밀번호 찾기
     * 이름, 아이디 이메일 검사
     * 임시비밀번호 서비스로 member 전달.
     * @param name
     * @param loginId
     * @param email
     * @param accessCode
     * @return String(afterloginUri)
     */
    @RequestMapping("/usr/member/findLoginPw")
    @ResponseBody
    public String findLoginPw(String name , String loginId, String email , @RequestParam(defaultValue = "/") String afterLoginUri) {

        Member member = memberService.getMemberByLoginId(loginId);

        if (member == null) {
            return Ut.jsHistoryBack("일치하는 회원이 없습니다.");
        }
        if (member.getName() == null) {
            return Ut.jsHistoryBack("이름이 일치하지 않습니다.");
        }

        if (member.getEmail().equals(email) == false) {
            return Ut.jsHistoryBack("이메일이 일치하지 않습니다.");
        }

        ResultData notifyTempLoginPwByEmailRd = memberService.notifyTempLoginPwByEmailRd(member);

        return Ut.jsReplace(notifyTempLoginPwByEmailRd.getMsg(), Ut.f("../member/login?afterLoginUri=%s",afterLoginUri));
    }
}
```

05 각 기능 소개

MemberService

비동기 아이디 찾기 메시지 전송

```
    }  
    /**  
    * 아이디 찾기  
    * 일치 회원 검색  
    * member가 null 이면 일치회원 없음  
    * 있으면 loginId 를 반환  
    * @param name  
    * @param email  
    * @return  
    */  
    public ResultData findLoginId(String name, String email) {  
        Member existsMember = memberRepository.getMemberByNameAndEmail(name, email);  
        if(existsMember == null) {  
            return ResultData.from("F-1",Ut.f("일치하는 회원이 없습니다."));  
        }  
        return ResultData.from("S-1","아이디 조회 성공","loginId",existsMember.getLoginId());  
    }  
}
```

05 각 기능 소개

MemberService

비밀번호 찾기 후 이메일 전송

임시비밀번호 변경

```
/**
 * 멤버 객체 actor 받음
 * 전역변수 사이트 이름, 주소 설정
 * util.getTempPassword에서 무작위 난수6자리 받음
 * 이메일 내용 작성
 * 메일서비스.send로 메일 전송하고 ResultData 받음
 * 메일 발송 실패 메세지일 경우 결과값 그냥 리턴
 * 임시 비밀번호로 비밀번호 변경
 * @param actor
 * @return ResultData(성공메세지,코드)
 */
public ResultData notifyTempLoginPwByEmailRd(Member actor) {
    String title = "[" + siteName + "] 임시 패스워드 발송";
    String tempPassword = Ut.getTempPassword(6);
    String body = "<h1>임시 패스워드 : " + tempPassword + "</h1>";
    body += "<a href=\"" + siteMainUri + "/usr/member/login\" target=\"_blank\">로그인 하러가기</a>";

    ResultData sendResultData = mailService.send(actor.getEmail(), title, body);

    if (sendResultData.isFail()) {
        return sendResultData;
    }

    setTempPassword(actor, tempPassword);

    return ResultData.from("S-1", "계정의 이메일주소로 임시 패스워드가 발송되었습니다.");
}

/**
 * 멤버와 임시비밀번호 받아서 해당 멤버의 비밀번호 임시비밀번호로 변경
 * @param actor
 * @param tempPassword
 */
private void setTempPassword(Member actor, String tempPassword) {
    memberRepository.doChangePassword(actor.getId(), Ut.sha256(tempPassword));
}
```

05 각 기능 소개

MemberRepository

아이디와 이메일로 일치하는 회원 찾기

```
<select id="getMemberByNameAndEmail" resultType="Member">
    SELECT *
    FROM `member`
    WHERE `name`
        = #{name}
    AND email
        = #{email}
</select>
```

MemberRepository

비밀번호 변경

```
<update id="doChangePassword">
    UPDATE `member` SET
        updateDate = NOW(),
        loginPw = #{loginPw}
    WHERE id = #{memberId}
</update>
```

05 각 기능 소개

회원가입 폼 양식

비동기 아이디 조회

회원가입

아이디

비밀번호

생년월일

년-월-일

이름

영문이름

전화번호

01012341234

이메일

이미지 파일 업로드

파일 선택

선택된 파일 없음

Sign up

아이디

id3

이미 사용중인 아이디 입니다.

아이디

S

2자 이상 입력해주세요

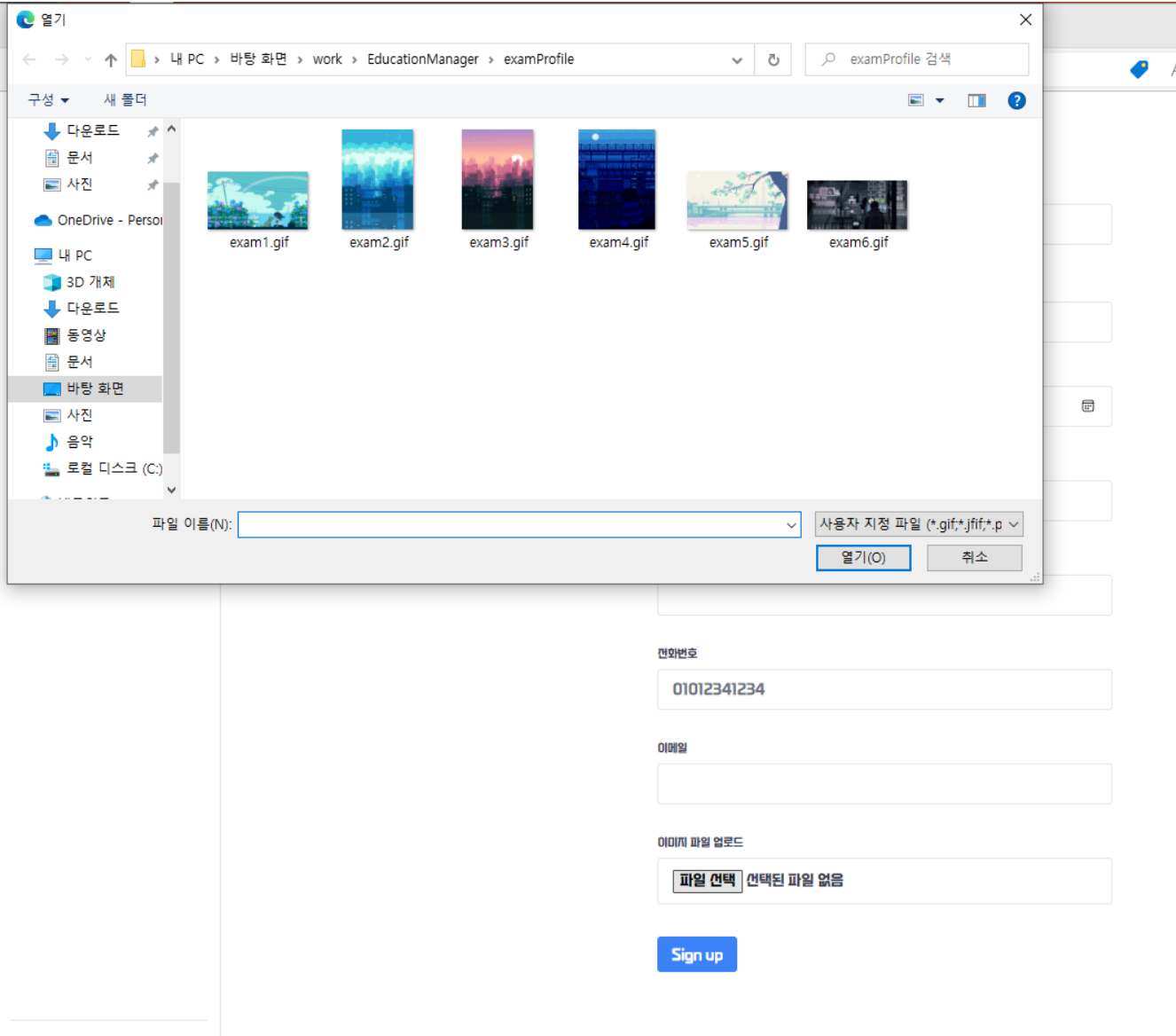
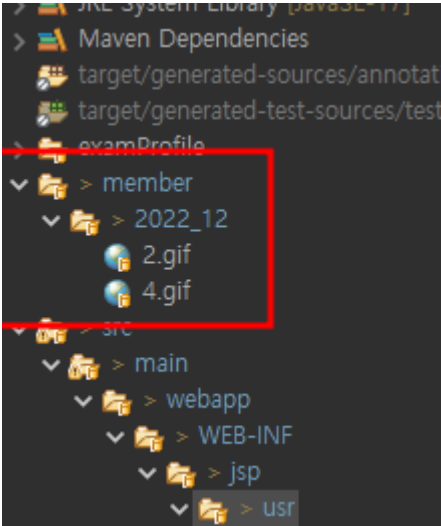
아이디

id13124

사용가능한 아이디 입니다.


05 각 기능 소개

프로필 사진 업로드



05 각 기능 소개

프로필 이미지를 등록한 회원의 프로필
이미지 노출



테스트 글 작성


직원

글번호: 9 글 작성일: 2022-12-12 22:58:45 글 수정일: 2022-12-12 22:58:45 조회수: 1


좋아요👍 0

싫어요👎 0

테스트 글 작성

 직원님 접속 중

프로필 이미지



아이디

id1

회원 등급

관리자 회원

가입일자

2022-12-12 22:51

비밀번호 변경

이름

직원

영문이름

user1

생년월일

1997-02-03

PHONE NUMBER

01012356781

E-MAIL

a333d@gmail.com

프로필 이미지 변경

파일 선택

선택된 파일 없음

회원정보 수정

탈퇴하기

05 각 기능 소개

Member/join

아이디 중복 체크 비동기 조회

아이디를 2글자 이상 입력받도록 유도

Lodash 를 사용하여 0.5초 이하의 입력
속도에는 동작하지 않도록 함

```
<script>
function checkLoginIdDup() {
  if ($('#loginId').val().length > 2) {
    $.get('../member/checkLoginId', {
      loginId : $('#loginId').val(),
      ajaxMode : 'Y'
    }, function(data) {

      if (data.resultCode == 'F-1') {
        $('#check-loginId').css("color", "red");
      }
      if (data.resultCode == 'S-1') {
        $('#check-loginId').css("color", "green");
      }
      $('#check-loginId').empty().html(data.msg);
    }, 'json');
  } else {
    $('#check-loginId').empty().html("2자 이상 입력해주세요")
      .css("color", "red");
  }
}
const checkLoginIdDupDebounce = _.debounce(checkLoginIdDup, 500);
</script>

<script>
let memberJoin__submitFormDone = false;
function memberJoin__submitForm(form) {
  alert(form.loginId.value.length);
  if (memberJoin__submitFormDone) {
    return;
  }
  form.loginId.value = form.loginId.value.trim();

  if (form.loginId.value.length <= 2) {
    alert('아이디는 2자 이상입니다. ');
    form.loginId.focus();
    return;
  }
  memberJoin__submitFormDone = true;
  form.submit();
}
```

05 각 기능 소개

Image size check

```
<script>
  let memberJoin__submitFormDone = false;
  function memberJoin__submitForm(form) {
    alert(form.loginId.value.length);
    if (memberJoin__submitFormDone) {
      return;
    }
    form.loginId.value = form.loginId.value.trim();

    if (form.loginId.value.length <= 2) {
      alert('아이디는 2자 이상입니다. ');
      form.loginId.focus();
      return;
    }

    const maxSizeMb = 10;
    const maxSize = maxSizeMb * 1204 * 1204;
    const profileImgFileInput = form["file__member__0__extra__profileImg__1"];
    if (profileImgFileInput.value) {
      if (profileImgFileInput.files[0].size > maxSize) {
        alert(maxSizeMb + "MB 이하의 파일을 업로드 해주세요.");
        profileImgFileInput.focus();
        return;
      }
    }
    memberJoin__submitFormDone = true;
    form.submit();
  }
</script>
```

05 각 기능 소개

CommonGenFileController

업로드

다운로드

```
@RequestMapping("/common/genFile/doUpload")
@ResponseBody
public ResultData doUpload(@RequestParam Map<String, Object> param, MultipartRequest multipartRequest) {
    return genFileService.saveFiles(param, multipartRequest);
}

@GetMapping("/common/genFile/doDownload")
public ResponseEntity<Resource> downloadFile(int id, HttpServletRequest request) throws IOException {
    GenFile genFile = genFileService.getGenFile(id);

    if (genFile == null) {
        throw new GenFileNotFoundException();
    }

    String filePath = genFile.getFilePath(genFileDirPath);

    Resource resource = new InputStreamResource(new FileInputStream(filePath));

    // Try to determine file's content type
    String contentType = request.getServletContext().getMimeType(new File(filePath).getAbsolutePath());

    if (contentType == null) {
        contentType = "application/octet-stream";
    }

    return ResponseEntity.ok()
        .header(HttpHeaders.CONTENT_DISPOSITION, "attachment; filename=\"" + genFile.getOriginFileName() + "\"")
        .contentType(MediaType.parseMediaType(contentType)).body(resource);
}
```

05 각 기능 소개

MemberController

```
*/
@RequestMapping("/usr/member/dojoin")
@ResponseBody
public String doJoin(String loginId, String loginPw, String birthDay, String name, String englishName,
    String cellphoneNum, String email, Model model, @RequestParam(defaultValue = "/") String afterLoginUri, MultipartRequest multipartRequest) {

    ResultData doJoinRd = memberService.doJoin(loginId, loginPw, birthDay, name, englishName, cellphoneNum, email);

    if (doJoinRd.isFail()) {
        if (doJoinRd.getResultCode().equals("F-1")) {
            return Ut.jsHistoryBack(Ut.f("이미 사용중인 아이디 입니다."));
        }
        if (doJoinRd.getResultCode().equals("F-2")) {
            return Ut.jsHistoryBack(Ut.f("이미 가입된 회원입니다."));
        }
    }
    int newMemberId = (int) doJoinRd.getBody().get("id");

    Map<String, MultipartFile> fileMap = multipartRequest.getFileMap();

    for (String fileInputName : fileMap.keySet()) {
        MultipartFile multipartFile = fileMap.get(fileInputName);

        if (multipartFile.isEmpty() == false) {
            genFileService.save(multipartFile, newMemberId);
        }
    }

    return Ut.jsReplace(Ut.f("회원가입 성공!"),
        Ut.f("../member/login?afterLoginUri=%s", Ut.getUriEncoded(afterLoginUri)));
}
```

05 각 기능 소개

MemberService

아이디 체크

이름, 이메일 중복체크

해당되지 않을 경우

비밀번호를 암호화 해서 회원가입

```
/**
 * 회원가입
 * @param loginId
 * @param loginPw
 * @param birthDay
 * @param name
 * @param englishName
 * @param cellphoneNum
 * @param email
 * @return
 */
public ResultData doJoin(String loginId, String loginPw, String birthDay, String name, String englishName, String cellphoneNum, String email) {

    Member existsMember = memberRepository.getMemberByLoginId(loginId);

    if(existsMember != null) {
        return ResultData.from("F-1",Ut.f("이미 사용중인 아이디 %s 입니다.",loginId));
    }
    existsMember = memberRepository.getMemberByNameAndEmail(name, email);

    if(existsMember != null) {
        return ResultData.from("F-2",Ut.f("이미 가입된 회원입니다. %s, %s",name, email));
    }

    loginPw = Ut.sha256(loginPw);
    memberRepository.doJoin(loginId, loginPw, birthDay, name, englishName, cellphoneNum, email);

    return ResultData.from("S-1","회원가입 성공");
}
```

05 각 기능 소개

memberRepository

같은 아이디 가진 멤버 조회.

이름과 이메일이 일치하는 멤버 조회

회원 가입

```
<select id="getMemberByLoginId" resultType="Member">
    SELECT *
    FROM
    `member`
    WHERE loginId
    = #{loginId}
</select>
```

```
<select id="getMemberByNameAndEmail" resultType="Member">
    SELECT *
    FROM `member`
    WHERE `name`
    = #{name}
    AND email
    = #{email}
</select>
```

```
<insert id="doJoin">
    INSERT INTO `member`
    SET regDate = NOW(),
    updateDate = NOW(),
    loginId = #{loginId},
    loginPw = #{loginPw},
    birthDay = #{birthDay},
    `name` = #{name},
    englishName = #{englishName},
    cellphoneNum = #{cellphoneNum},|
    email = #{email}
</insert>
```


05 각 기능 소개

Common / sidebar

로그인 후 authlevel 이 관리자 권한일
경우 관리자 페이지를 보여줌

로그인 상태인지 로그아웃 상태인지에
따라 로그인과 로그아웃 메뉴 보여줌

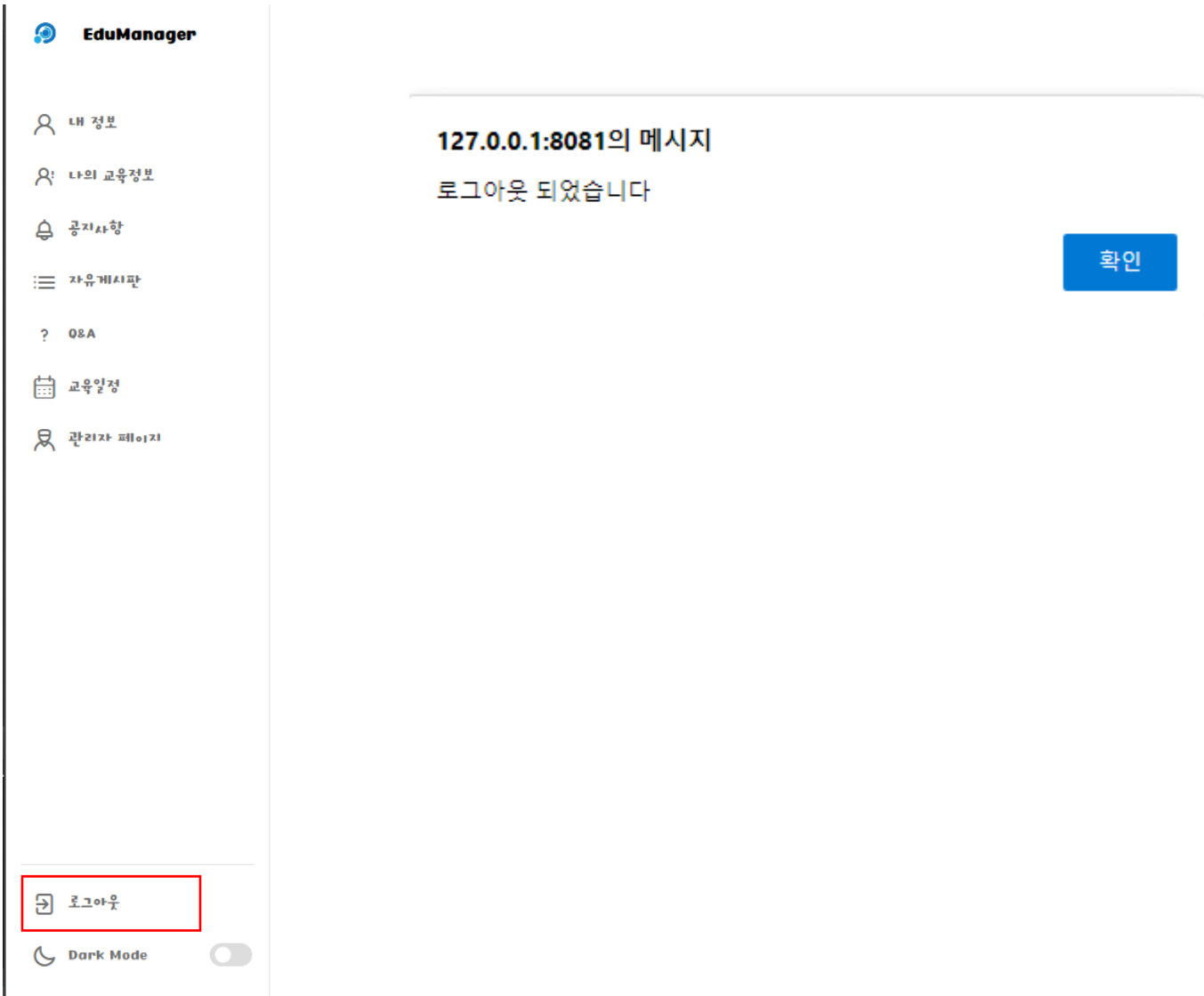
```
<c:if test="${rq.loginedMember.authLevel == 1 || rq.loginedMember.authLevel == 2}">
  <li><a href="../member/administrator">
    <i class="uil uil-user-md"></i>
    <span class="link-name">관리자 페이지</span>
  </a></li>
</c:if>

</ul>

<ul class="logout-mode">
  <c:if test="${rq.isLogined()}">
    <li><a href="${rq.logoutUri}">
      <i class="uil uil-signout"></i>
      <span class="link-name">로그아웃</span>
    </a></li>
  </c:if>
  <c:if test="${!rq.isLogined()}">
    <li><a href="${rq.loginUri}">
      <i class="uil uil-signin"></i>
      <span class="link-name">로그인</span>
    </a></li>
  </c:if>
</ul>
```

05 각 기능 소개

Logout 기능



05 각 기능 소개

memberController

로그인이 되지 않았을때는 작동하지 않도록 함

Rq 객체에 로그아웃 명령 실행

```
/**
 * 로그아웃
 * rq에 로그아웃 명령
 * 로그아웃 후 uri로 이동
 * @param afterLogoutUri
 * @return
 */
@RequestMapping("/usr/member/doLogout")
@ResponseBody
public String doLogout(@RequestParam(defaultValue = "/")String afterLogoutUri) {
    if(rq.isNotLoggedIn()){
        return Ut.jsReplace(Ut.f("로그인이 필요한 서비스입니다."), Ut.getUriEncoded(afterLogoutUri));
    }
    rq.logout();

    return Ut.jsReplace("로그아웃 되었습니다", afterLogoutUri);
}
```

VO/rq

세션에서 값 제거

```
/**
 * 로그아웃
 * 세션에서 값 제거 Key
 */
public void logout() {
    session.removeAttribute("loggedInMemberId");
}
```

05 각 기능 소개

회원 정보 페이지

비밀번호 변경, 회원정보 수정, 탈퇴 기능

프로필 이미지

회원 등급

관리자 회원

가입일자

2022-12-12 22:51

아이디

id1

비밀번호 변경

이름

직원

영문이름

user1

생년월일

1997-02-03

PHONE NUMBER

01012356781

E-MAIL

a333d@gmail.com

프로필 이미지 변경

파일 선택

선택된 파일 없음

회원정보 수정

탈퇴하기

05 각 기능 소개

비밀번호 변경 버튼 클릭 시

나타나는 비밀번호 입력 창

현재 비밀번호, 새로운 비밀번호를 입력받아
변경할 수 있다.

입력 받은 값이 없을 경우 나오는 메시지

관리자 회원

가입일자

2022-12-05 11:26

아이디

id1

현재 비밀번호

새 비밀번호

비밀번호 변경

취소

현재 비밀번호


*

새 비밀번호

|

비밀번호 변경

취소

 이 입력란을 작성하세요.

05 각 기능 소개

현재 비밀번호를 잘못 입력 했을 때

127.0.0.1:8081 내용:
비밀번호가 틀렸습니다.

확인

비밀번호 수정이 성공했을 때

127.0.0.1:8081 내용:
비밀번호가 수정되었습니다!

확인

05 각 기능 소개

회원정보 수정 버튼을 눌렀을 때 활성화 되는 입력창

이름은 수정할 수 없다.

프로필 이미지

회원 등급

관리자 회원

가입일자

2022-12-12 22:51

아이디

id1

비밀번호 변경

이름

직원

영문이름

user1

생년월일

1997-02-03

PHONE NUMBER

01012356781

E-MAIL

a333d@gmail.com

프로필 이미지 변경

파일 선택

선택된 파일 없음

수정 완료

취소

탈퇴하기

05 각 기능 소개

입력란이 공백인 경우 나오는 메시지

이름

직원

영문이름

생년월일

1997-02-03

이 입력란을 작성하세요.

PHONE NUMBER

0101235678

E-MAIL

a333d@gmail.com

수정 완료

취소

탈퇴하기

개인정보 수정 완료 메시지

127.0.0.1:8081 내용:
개인정보 수정완료

확인

05 각 기능 소개

탈퇴하기 버튼 클릭 시 확인 메시지

127.0.0.1:8081 내용:
정말 탈퇴하시겠습니까?

확인

취소

탈퇴 처리 메시지

127.0.0.1:8081 내용:
탈퇴 성공

확인

탈퇴 후 로그인 시도 메시지

127.0.0.1:8081 내용:
탈퇴한 회원입니다.

확인

05 각 기능 소개

Member/userInfo.jsp

비밀번호 변경 또는 회원정보 수정 시
인증코드 생성

```
<!-- 인증코드 생성 -->
<script>
const createAuthKey = () => {
$.get('../member/createAuthKey',{
  ajaxMode : 'Y'
}, function(AuthKey){
  $('#memberModifyAuthKey').val(AuthKey);
  $('#memberPasswordAuthKey').val(AuthKey);
});
}
</script>
```

Member/userInfo.jsp

비밀번호 수정 버튼, 회원정보 수정 버튼
클릭 시 폼을 변경

```
<!-- 회원정보 수정 -->
<script>
const memberInfoModify =()=>{
  $('#birthDay').removeAttr('disabled').removeClass('bg-gray-300');
  $('#englishName').removeAttr('disabled').removeClass('bg-gray-300');
  $('#cellphoneNum').removeAttr('disabled').removeClass('bg-gray-300');
  $('#email').removeAttr('disabled').removeClass('bg-gray-300');
  $('#modifyBtn').addClass('hidden');
  $('#doModify').removeClass('hidden');
  $('#cancleModify').removeClass('hidden');
}
</script>
<!-- 비밀번호 변경 -->
<script>
const passwordChange = () =>{
  $('#password-check-label').removeClass('hidden');
  $('#new-password-label').removeClass('hidden');
  $('#loginPwCheck').attr("type","password").removeAttr('disabled').val("");
  $('#loginPw').attr("type","password").removeAttr('disabled').val("");
  $('#changePasswordBtn').addClass('hidden');
  $('#changePasswordConfirm').removeClass('hidden');
  $('#changePasswordCancle').removeClass('hidden');
}
</script>
```

05 각 기능 소개

Member/userInfo.jsp

비밀번호 변경 취소, 회원정보 수정 취소 버튼 클릭 시 폼 숨기기, 비활성화

```
<!-- 비밀번호 변경 취소-->

<script>
const passwordChangeCancle = () =>{
  $('#password-check-label').addClass('hidden');
  $('#new-password-label').addClass('hidden');
  $('#loginPwCheck').attr("type","hidden").attr("disabled", true);
  $('#loginPw').attr("type","hidden").attr("disabled", true);
  $('#changePasswordBtn').removeClass('hidden');
  $('#changePasswordConfirm').addClass('hidden');
  $('#changePasswordCancle').addClass('hidden');
}
</script>

<!-- 회원정보 수정 취소 -->

<script>
const memberCancleModify =()=>{
  $('#birthDay').attr("disabled", true).val('${member.birthDay}').addClass('bg-gray-300');
  $('#englishName').attr("disabled", true).val('${member.englishName}').addClass('bg-gray-300');
  $('#cellphoneNum').attr("disabled", true).val('${member.cellphoneNum}').addClass('bg-gray-300');
  $('#email').attr("disabled", true).val('${member.email}').addClass('bg-gray-300');
  $('#modifyBtn').removeClass('hidden');
  $('#doModify').addClass('hidden');
  $('#cancleModify').addClass('hidden');
}
</script>
```

05 각 기능 소개

memberController

Form 에서 보낸 인증코드 확인

기존 비밀번호 일치 여부 확인

비밀번호 변경

```
/**
 * 인증코드 확인
 * 기존 비밀번호 일치 확인
 * 비밀번호 수정
 * @param loginPwCheck
 * @param loginPw
 * @param memberPasswordAuthKey
 * @return String
 */
@RequestMapping("/usr/member/doChangePassword")
@ResponseBody
public String doChangePassword(String loginPwCheck, String loginPw, String memberPasswordAuthKey) {

    if(memberPasswordAuthKey == null) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 생성되지 않았습니다."));
    }

    Attr getAttr = attrService.get("member", rq.getLoginedMemberId(), "extra", "memberModifyAuthKey");

    if(getAttr==null) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 만료되었습니다. 다시 시도해주세요"));
    }

    if(!getAttr.getValue().equals(memberPasswordAuthKey)) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 일치하지 않습니다. 다시 시도해주세요"));
    }

    Member member = memberService.getMemberById(rq.getLoginedMemberId());

    if(!member.getLoginPw().equals(Ut.sha256(loginPwCheck))) {
        return Ut.jsReplace(Ut.f("비밀번호가 틀렸습니다."), "../member/userInfo");
    }

    ResultData doChangePasswordRd = memberService.doChangePassword(member.getId() , loginPw);

    return Ut.jsReplace(Ut.f("비밀번호가 수정되었습니다!"), "../member/userInfo");
}
```

05 각 기능 소개

MemberService

입력받은 암호 sha-256으로 암호화 후

Repository에 데이터베이스 변경 요청

```
/**
 * 비밀번호 변경
 * @param memberId
 * @param loginPw
 * @return ResultData
 */
public ResultData doChangePassword(int memberId ,String loginPw) {
    memberRepository.doChangePassword(memberId,Ut.sha256(loginPw));
    return ResultData.from("S-1","비밀번호 수정 성공");
}
```

MemberRepository

Update 쿼리

```
<update id="doChangePassword">
    UPDATE `member` SET
    updateDate = NOW(),
    loginPw = #{loginPw}
    WHERE id = #{memberId}
</update>
```

05 각 기능 소개

memberController

인증코드 확인

입력값이 공백이거나 없는 경우 확인

Service 단에 정보 변경 요청

실패 또는 성공 메시지 출력

```
@RequestMapping("/usr/member/doModify")
@ResponseBody
public String doModify(String birthDay, String name, String englishName, String cellphoneNum, String email, String memberModifyAuthKey, Model model) {

    if(memberModifyAuthKey == null) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 생성되지 않았습니다."));
    }

    Attr getAttr = attrService.get("member", rq.getLoginedMemberId(), "extra", "memberModifyAuthKey");

    if(getAttr==null) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 만료되었습니다. 다시 시도해주세요"));
    }
    if(!getAttr.getValue().equals(memberModifyAuthKey)) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 일치하지 않습니다. 다시 시도해주세요"));
    }
    birthDay = birthDay.replaceAll("-", "");

    if(Ut.empty(birthDay)) {
        return Ut.jsHistoryBack(Ut.f("생년월일을 입력해주세요"));
    }
    if(Ut.empty(cellphoneNum)) {
        return Ut.jsHistoryBack(Ut.f("전화번호를 입력해주세요"));
    }
    if(Ut.empty(englishName)) {
        return Ut.jsHistoryBack(Ut.f("영문이름을 입력해주세요"));
    }
    if(Ut.empty(email)) {
        return Ut.jsHistoryBack(Ut.f("이메일을 입력해주세요"));
    }

    ResultData doModifyRd = memberService.doModify( birthDay, name, englishName, cellphoneNum, email, rq.getLoginedMemberId());

    if(doModifyRd.isFail()) {
        if(doModifyRd.getResultCode().equals("F-1")) {
            return Ut.jsHistoryBack(doModifyRd.getMsg());
        }
    }
    Member member = memberService.getMemberById((int) doModifyRd.getData1());
    ResultData resultRd = ResultData.newData(doModifyRd, "member", member);
    return Ut.jsReplce("개인정보 수정완료", "../member/userInfo");
}
```

05 각 기능 소개

MemberService

이름과 이메일이 기존회원과 중복되는
경우
실패 메시지와 함께 종료

Repository 에 정보 수정 요청

```
/**
 * 개인정보 수정
 *
 * @param name
 * @param nickname
 * @param cellphoneNum
 * @param email
 * @param memberId
 * @return ResultData(memberId)
 */
public ResultData doModify(String birthDay, String name, String englishName, String cellphoneNum, String email, int memberId) {

    Member existsMember = memberRepository.getMemberByNameAndEmail(name, email);
    if(existsMember != null) {
        return ResultData.from("F-1", Ut.f("중복된 회원 정보입니다. %s, %s", name, email));
    }
    memberRepository.doModify(birthDay, englishName, cellphoneNum, email, memberId);
    return ResultData.from("S-1", "회원정보 수정 성공", "memberId", memberId);
}
```

MemberRepository

회원 정보 수정 업데이트 쿼리

```
<update id="doModify">
    UPDATE `member` SET
    updateDate = NOW(),
    birthDay = #{birthDay},
    englishName = #{englishName},
    cellphoneNum = #{cellphoneNum},
    email = #{email}
    WHERE id = #{memberId}
</update>
```

05 각 기능 소개

멀티 게시판

공지사항, 자유게시판, Q&A 게시판

글 작성 버튼은 로그인 하지 않으면 보이지 않는다.

공지사항 게시판

게시글: 4개

검색조건 ▾

검색어를 입력해주세요

검색

번호	작성일시	제목	작성자	조회수	추천
4	12-09 16:16	공지사항4	직원	0	<div><div>👍 0</div><div>👎 0</div></div>
3	12-09 16:16	공지사항3	관리자	0	<div><div>👍 0</div><div>👎 0</div></div>
2	12-09 16:16	공지사항2	직원	0	<div><div>👍 0</div><div>👎 0</div></div>
1	12-09 16:16	공지사항1	관리자	0	<div><div>👍 0</div><div>👎 0</div></div>

글 작성

1

자유 게시판

게시글: 2개

검색조건 ▾

검색어를 입력해주세요

검색

번호	작성일시	제목	작성자	조회수	추천
6	12-09 16:16	제목4	관리자	0	<div><div>👍 0</div><div>👎 0</div></div>
5	12-09 16:16	제목3	사용자2	0	<div><div>👍 0</div><div>👎 0</div></div>

글 작성

1

Q&A 게시판

게시글: 2개

검색조건 ▾

검색어를 입력해주세요

검색

번호	작성일시	제목	작성자	조회수	추천
8	12-09 16:16	제목5	관리자	0	<div><div>👍 0</div><div>👎 0</div></div>
7	12-09 16:16	제목5	사용자2	0	<div><div>👍 0</div><div>👎 0</div></div>

글 작성

1

05 각 기능 소개

글 검색 기능

조건을 설정하지 않았을 때 제목과 내용으로 검색

검색 조건은 제목, 내용, 작성자 로 검색 가능

공지사항 게시판

게시글: 1개

검색조건 ▾ 3

검색

번호	작성일시	제목	작성자	조회수	추천
3	12-09 16:16	공지사항3	관리자	0	<div>👍 0</div> <div>👎 0</div>

글 작성

1

검색조건 ▾

내

검색

검색조건

제목

내용

작성자

작성일시	작성자	조회수	추천
34	직원	0	<div>👍 0</div> <div>👎 0</div>

페이징 처리

공지사항 게시판

게시글: 204개

검색조건 ▾

검색어를 입력해주세요

검색

번호	작성일시	제목	작성자	조회수	추천	
244	12-09 16:57	공지사항4	직원	0	0	0
243	12-09 16:57	공지사항3	관리자	0	0	0
242	12-09 16:57	공지사항2	직원	0	0	0
241	12-09 16:57	공지사항1	관리자	0	0	0
236	12-09 16:57	공지사항4	직원	0	0	0
235	12-09 16:57	공지사항3	관리자	0	0	0
234	12-09 16:57	공지사항2	직원	0	0	0
233	12-09 16:57	공지사항1	관리자	0	0	0
228	12-09 16:57	공지사항4	직원	0	0	0
227	12-09 16:57	공지사항3	관리자	0	0	0

글 작성

05 각 기능 소개

Article/list

```
<div class="flex justify-center mb-11">

  <div class="mt-4">
    <c:set var="pageMenuLen" value="4" />
    <c:set var="startPage"
      value="${page-pageMenuLen >= 1 ? page-pageMenuLen : 1}" />
    <c:set var="endPage"
      value="${page + pageMenuLen <= pageCount ? page + pageMenuLen : pageCount}" />
    <c:set var="pageBaseUri" value="?boardId=${boardId}" />
    <c:set var="pageBaseUri"
      value="${pageBaseUri}&searchWord=${param.searchWord}" />
    <c:set var="pageBaseUri"
      value="${pageBaseUri}&searchFrom=${param.searchFrom}" />

    <c:if test="${startPage > 1}">
      <button class="page-move-btn"
        onclick="location.href='../article/list${pageBaseUri}&page=1';">1</button>
    </c:if>
    <c:if test="${startPage > 2}">
      <button class="page-move-btn page-move-btn-not-active">...</button>
    </c:if>
    <c:forEach var="pageNum" begin="${startPage}" end="${endPage}">
      <button
        class="page-move-btn ${page == pageNum ? 'page-move-btn-active' : ''}"
        onclick="location.href='../article/list${pageBaseUri}&page=${pageNum}';">${pageNum}</button>
    </c:forEach>
    <c:if test="${endPage < pageCount-1}">
      <button class="page-move-btn page-move-btn-not-active">...</button>
    </c:if>
    <c:if test="${endPage < pageCount}">
      <button class="page-move-btn"
        onclick="location.href='../article/list${pageBaseUri}&page=${pageCount}';">${pageCount}</button>
    </c:if>
  </div>
</div>
```

05 각 기능 소개

articleController

게시판 조회

조건에 맞는 글을 서비스에 요청

검색기록 유지와 페이징처리를 위해
필요한 값을 model 에 추가

```
/**
 * 게시판 종류, 검색결과에 따른 리스트 출력
 * @param model
 * @param boardId
 * @param page
 * @param searchWord
 * @param searchFrom
 * @return String
 */
@RequestMapping("/usr/article/list")
public String showList( Model model,
    @RequestParam(defaultValue = "1")Integer boardId,
    @RequestParam(defaultValue = "1")Integer page,
    @RequestParam(defaultValue = "")String searchWord,
    @RequestParam(defaultValue = "")String searchFrom) {
    if(boardId == null) {
        return rq.jsHistoryBackOnView("지정되지 않은 게시판");
    }
    Board board = boardService.getBoardById(boardId);
    if(board == null) {
        return rq.jsHistoryBackOnView("게시판이 존재하지 않습니다.");
    }
    int itemsInAPage = 10;
    int limitFrom = (page - 1) * itemsInAPage;

    List<Article> articles = articleService.getForPrintArticles(rq.getLoginedMemberId(),boardId,limitFrom,itemsInAPage, searchWord, searchFrom);
    int getTotalArticle = articleService.getTotalArticle(boardId, searchWord, searchFrom);
    int pageCount = (int) Math.ceil(((double)getTotalArticle/itemsInAPage);

    model.addAttribute("searchFrom",searchFrom);
    model.addAttribute("searchWord",searchWord);
    model.addAttribute("getTotalArticle",getTotalArticle);
    model.addAttribute("page",page);
    model.addAttribute("pageCount",pageCount);
    model.addAttribute("boardId",boardId);
    model.addAttribute("board", board);
    model.addAttribute("articles", articles);
    return "usr/article/list";
}
```

05 각 기능 소개

MemberService

Article Repository에

조건에 맞는 게시글과

조건에 맞는 전체 게시글 수 요청

```
public List<Article> getForPrintArticles(int actorId, int boardId, int limitFrom, int itemsInAPage, String searchWord, String searchFrom) {  
    List<Article> articles = articleRepository.getArticles(boardId, limitFrom, itemsInAPage, searchWord, searchFrom);  
  
    for (Article article : articles) {  
        updateForPrintData(actorId, article);  
    }  
  
    return articles;  
}
```

```
public int getTotalArticle(int boardId, String searchItem, String searchFrom) {  
    return articleRepository.getTotalArticle(boardId, searchItem, searchFrom);  
}
```

05 각 기능 소개

Article repository

조건에 맞는 게시글 수를 조회하는 쿼리

```
<select id="getTotalArticle" resultType="int">
  SELECT COUNT(*)
  FROM article AS A
  INNER JOIN `member` AS M
  ON A.memberId
  = M.id
  <if test="boardId !=0">
    AND A.boardId = #{boardId}
  </if>
  <if test="searchWord !=''">
    <choose>
      <when test="searchFrom != ''">
        AND ${searchFrom} LIKE '%${searchWord}%'
      </when>
      <otherwise>
        AND (
          A.title LIKE '%${searchWord}%'
          OR A.body LIKE
            '%${searchWord}%'
        )
      </otherwise>
    </choose>
  </if>
</select>
```

Article repository

조건에 맞는 게시글을 조회하는 쿼리

```
<select id="getArticles" resultType="Article">
  SELECT A.*, M.name AS
  extra__writerName
  FROM article AS A
  INNER
  JOIN
  `member` AS M
  ON A.memberId
  = M.id
  <if test="boardId !=0">
    AND A.boardId = #{boardId}
  </if>
  AND A.delStatus = 0
  <if test="searchWord !=''">
    <choose>
      <when test="searchFrom != ''">
        AND ${searchFrom} LIKE '%${searchWord}%'
      </when>
      <otherwise>
        AND (
          A.title LIKE '%${searchWord}%'
          OR A.body LIKE
            '%${searchWord}%'
        )
      </otherwise>
    </choose>
  </if>
  ORDER BY A.id DESC
  LIMIT #{limitFrom} , #{itemsInAPage}
</select>
```

05 각 기능 소개

글 작성 폼

TOAST UI editor 를 사용

EduManager

- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정
- 관리자 페이지

- 로그아웃
- Dark Mode



공지사항 ▼ 테스트글 작성

H B I A[•] S | — 66 | ≡ 1≡ ☑ ≡ ≡ | 📄 🖼️ 🔗 | </> CB | Scroll ☐

게시판 테스트글 작성

게시글 작성

1. 글 넘버링

2. 팔 됩니다.

글 작성 팔 됩니다.

| column1 | column2 | column3 |

| — | — | — | — |

| row1 | 111 | 222 | 333 |

| row2 | 123 | 456 | 789 |

| row3 | 테이블 | 작성 | 테스트 |

게시판 테스트글 작성

게시글 작성

1. 글 넘버링

2. 팔 됩니다.

글 작성 팔 됩니다.

	column1	column2	column3
row1	111	222	333
row2	123	456	789
row3	테이블	작성	테스트

Markdown WYSIWYG

글 작성



직원님 접속 중

05 각 기능 소개

글 작성 완료 메시지

127.0.0.1:8081 내용:
409번 게시물 작성

확인

글 작성 완료 후 작성된 글로 이동

EduManager

내 정보

나의 교육정보

공지사항

자유게시판

Q&A

교육일정

관리자 페이지

로그아웃

Dark Mode

직원님 접속 중

테스트 글 작성

작성글번호: 409 글 작성일: 2022-12-09 17:47:27 글 수정일: 2022-12-09 17:47:27 조회수: 1

좋아요 0싫어요 0

게시판 테스트 글 작성

게시글 작성

1. 글 넘버링
2. 말 씁니다.

글 작성 말 씁니다.

	column1	column2	column3
row1	111	222	333
row2	123	456	789
row3	테이블	작성	테스트

공지사항 작성 권한 부여

127.0.0.1:8081의 메시지

공지사항 작성 권한이 없습니다.

확인

```
<script>
  let submitWriteFormDone = false;
  function submitWriteForm(form) {
    if(form.boardId.value == 1){
      if(${rq.loggedMember.authLevel >2}){
        alert('공지사항 작성 권한이 없습니다.');
```

```
      return;
    }
  }
  if (submitWriteFormDone) {
    alert('처리중입니다');
    return;
  }
  form.title.value = form.title.value.trim();
  if (form.title.value == 0) {
    alert('제목을 입력해주세요');
    return;
  }
  const editor = $(form).find('.toast-ui-editor').data(
    'data-toast-editor');
  // find(찾는 엘리먼트).data(불러오는 데이터의 key 값)
  // find 가 제이쿼리이기 때문에 $로 가져와야함.
  const markdown = editor.getMarkdown().trim();
  // 폼 태그 안에 에디터 안에 데이터트스트에디터의 value 값을 마크다운으로 가져온다.
  if (markdown.length == 0) {
    alert('내용을 입력해주세요');
    editor.focus();
    return;
  }
  form.body.value = markdown;
  form.submit();

  submitWriteFormDone = true;
}
</script>
```

05 각 기능 소개

공지사항 글 작성 권한

```
@RequestMapping("/usr/article/doAdd")
@ResponseBody
public String doAdd(String title, String body, Integer boardId, Model model, String listUri) {
    if(boardId == null) {
        return Ut.jsHistoryBack(Ut.f("게시판을 선택해주세요"));
    }
    if(boardId == 1 && rq.getLoginedMember().getAuthLevel() > 2) {
        return Ut.jsHistoryBack(Ut.f("공지사항 작성 권한이 없습니다."));
    }
    if(Ut.empty(title)) {
        return Ut.jsHistoryBack(Ut.f("제목을 입력해주세요"));
    }
    if(Ut.empty(body)) {
        return Ut.jsHistoryBack(Ut.f("내용을 입력해주세요"));
    }
    String setBody = body.replaceAll("\r\n", "<br>");

    ResultData<Integer> writeArticleRd = articleService.writeArticle(rq.getLoginedMemberId(), title, setBody, boardId);

    int id = (int) writeArticleRd.getData1();

    return Ut.jsReplace(Ut.f("%d번 게시글 작성", id), Ut.f("../article/detail?id=%d&listUri=%s", id, listUri));
}
```

05 각 기능 소개

Article / write.jsp

제목, 내용 입력 체크

```
<script>
let submitWriteFormDone = false;
function submitWriteForm(form) {
  if (submitWriteFormDone) {
    alert('처리중입니다');
    return;
  }
  form.title.value = form.title.value.trim();
  if (form.title.value == 0) {
    alert('제목을 입력해주세요');
    return;
  }
  const editor = $(form).find('.toast-ui-editor').data(
    'data-toast-editor');
  // find(찾는 엘리먼트).data(불러오는 데이터의 key 값)
  // find 가 제이쿼리이기 때문에 $로 가져와야함.
  const markdown = editor.getMarkdown().trim();
  // 폼 태그 안에 에디터 안에 데이터트스트에디터의 value 값을 마크다운으로 가져온다.
  if (markdown.length == 0) {
    alert('내용을 입력해주세요');
    editor.focus();
    return;
  }
  form.body.value = markdown;
  form.submit();

  submitWriteFormDone = true;
}
</script>
```

127.0.0.1:8081 내용:

제목을 입력해주세요

확인

127.0.0.1:8081 내용:

내용을 입력해주세요

확인

127.0.0.1:8081 내용:

게시판을 선택해주세요

확인

05 각 기능 소개

ArticleController

요청이 오면 작성 폼으로 이동시킨다.

게시판, 제목, 내용 을 체크

마크다운 언어 줄 바꿈 태그 변경

서비스에 글 등록 요청

마지막 글 등록 id 가져와서 몇번 게시물 생성 되었는지 표시

```
@RequestMapping("usr/article/write")
public String articleWriteForm(HttpServletRequest req, Model model,String listUri) {

    model.addAttribute("listUri", Ut.getUriEncoded(listUri));
    return "usr/article/write";
}
```

```
@RequestMapping("/usr/article/doAdd")
@ResponseBody
public String doAdd(String title, String body, Integer boardId, Model model, String listUri) {
    if(boardId == null) {
        return Ut.jsHistoryBack(Ut.f("게시판을 선택해주세요"));
    }
    if(Ut.empty(title)) {
        return Ut.jsHistoryBack(Ut.f("제목을 입력해주세요"));
    }
    if(Ut.empty(body)) {
        return Ut.jsHistoryBack(Ut.f("내용을 입력해주세요"));
    }
    String setBody = body.replaceAll("\r\n", "<br>");

    ResultData<Integer> writeArticleRd = articleService.writeArticle(rq.getLoginedMemberId(), title, setBody, boardId);

    int id = (int) writeArticleRd.getData1();

    return Ut.jsReplace(Ut.f("%d번 게시물 작성", id), Ut.f("../article/detail?id=%d&listUri=%s", id,listUri));
}
```

05 각 기능 소개

ArticleService

ArticleRepository에 글 등록 요청

마지막 게시글 번호 요청

```
public ResultData<Integer> writeArticle(int memberId, String title, String body, int boardId) {  
    articleRepository.writeArticle(memberId, title, body, boardId);  
    int id = articleRepository.getLastInsertId();  
  
    return ResultData.from("S-1", Ut.f("%d번 게시글이 생성되었습니다", id), "id", id);  
}
```

ArticleRepository

글 작성 insert 쿼리문

마지막 글 번호 조회 쿼리문

```
<insert id="writeArticle">  
    INSERT INTO article  
    SET regDate = NOW(),  
    updateDate =  
    NOW(),  
    memberId = #{memberId},  
    title = #{title},  
    `body` = #{body},  
    boardId = #{boardId}  
</insert>
```

```
<select id="getLastInsertId" resultType="int">  
    SELECT LAST_INSERT_ID()  
</select>
```

05 각 기능 소개

- 글 조회
- TOAST UI Viewer
- 글 정보, 추천 기능

내 정보

나의 교육정보

공지사항

자유게시판

Q&A

교육일정

관리자 페이지

로그아웃

Dark Mode ☐

작성글 목록

테스트 글 작성

작성글 번호: 409 글 작성일: 2022-12-09 17:47:27 글 수정일: 2022-12-09 17:47:27 조회수: 1

좋아요 0

싫어요 0

게시판 테스트 글 작성

게시글 작성

1. 글 넘버링

2. 잘 됩니다.

글 작성 잘 됩니다.

	column1	column2	column3
row1	111	222	333
row2	123	456	789
row3	테이블	작성	테스트

05 각 기능 소개

댓글 작성 및 글 작성자의 화면

뒤로가기

수정

삭제

댓글남기기:

작성

직원	댓글 작성 테스트	<div>수정</div>	<div>삭제</div>
----	-----------	---------------	---------------

댓글 작성 및 글 작성자가 아닌경우

뒤로가기

댓글남기기:

작성

직원	댓글 작성 테스트		
----	-----------	--	--

댓글 작성자의 경우 수정, 삭제 가능

뒤로가기

수정

삭제

댓글남기기:

작성

직원	댓글 작성 테스트 123	<div>수정</div>	<div>취소</div>
----	---------------	---------------	---------------

05 각 기능 소개

글 추천 기능

비동기 작동



테스트 글 작성

직원

글번호: 409 글 작성일: 2022-12-09 17:4

좋아요👍 1

싫어요👎 0



테스트 글 작성

직원

글번호: 409 글 작성일: 2022-12-0

좋아요👍 0

싫어요👎 1

05 각 기능 소개

Article/detail

조회수 증가 스크립트

SetTimeout 으로 글 조회 후 2초 뒤 증가

```
<%--
조회수 증가 비동기 통신 스크립트
get함수를 사용해서 컨트롤러와 매핑된 주소로 연결
id 값을 파라미터로 값을 주고
받은 데이터값(data)를 제이쿼리로 찾은 클래스요소 안에 empty로 비우고 html로 데이터를 넣는다.
실행은 2000ms 뒤에 실행
--%>
<script>
  function ArticleDetail__increaseHitCount() {
    const localStorageKey = 'article__' + params.id + '__alreadyView';
    if (localStorage.getItem(localStorageKey)) {
      return;
    }
    localStorage.setItem(localStorageKey, true);
    $.get('../article/increasedHit', {
      id : params.id,
      ajaxMode : 'Y'
    }, function(data) {
      $('#article-detail__hit').empty().html(data.data1);
    }, 'json');
  }
  $(function() {
    // 실전코드
    //ArticleDetail__increaseHitCount();
    // 연습코드
    setTimeout(ArticleDetail__increaseHitCount, 2000);
  })
</script>
```

05 각 기능 소개

Article/detail

좋아요 버튼 비동기 스크립트

```
<!--
좋아요 버튼 클릭 작동 스크립트
로그인했을때만 작동할 수있도록 경고창을 설정
파라미터값으로 게시물 id와 추천의 경우 point 를 1로 설정
받은 데이터는 총 3가지로 로그인한 사람의 리액션 결과, 좋아요 총 갯수, 싫어요 총 갯수 이다.
최초 상태에서 로그인한 사람의 리액션 결과가 전역변수로 설정되어 좋아요 버튼을 눌렀을때
싫어요, 또는 아무리액션이 없을 경우 좋아요가 될수 있도록 전역변수 변경
좋아요 상태에서 다시 좋아요 버튼을 눌렀을 때는 취소이기 때문에 전역변수를 0으로 설정
리액션 상태에 따른 표시를 위해 설정한 함수를 실행시키는것으로 끝
--%>
<script>
const ArticleDetail__goodReactionPoint = () =>{
  if(!isLoggedIn){
    loginAlert();
    return;
  }
  $.get('../reaction/doReaction',{
    relId : ${article.id},
    point : 1,
    ajaxMode : 'Y'
  }, function(data){
    $('.article-detail__Reaction').empty().html(data.data1);
    $('.article-detail__goodReaction').empty().html(data.data2);
    $('.article-detail__badReaction').empty().html(data.data3);
    if(reaction == -1 || reaction == 0){
      reaction = 1;
    }else if(reaction == 1 ){
      reaction = 0;
    }
    reactionPick();
  }, 'json');
}
</script>
```

05 각 기능 소개

Article/detail

싫어요 버튼 비동기 스크립트

```
<%--
싫어요 버튼 클릭 작동 스크립트
좋아요 버튼 클릭과 동일한 작동 방법
세부 조건만 조금 변경하였다.
--%>
<script>
const ArticleDetail__badReactionPoint = () =>{
  if(!isLoggedIn){
    loginAlert();
    return;
  }
  $.get('../reaction/doReaction',{
    relId : ${article.id},
    point : -1,
    ajaxMode : 'Y'
  }, function(data){
    $('.article-detail__Reaction').empty().html(data.data1);
    $('.article-detail__goodReaction').empty().html(data.data2);
    $('.article-detail__badReaction').empty().html(data.data3);
    if(reaction == 1 || reaction == 0){
      reaction = -1;
    }else if(reaction == -1 ){
      reaction = 0;
    }
    reactionPick();
  }, 'json');
}
</script>
```

05 각 기능 소개

Article/detail

좋아요, 싫어요에 따른 글 색상 변경

로그인하지 않을 경우 작동하지 않도록 경고 메시지 출력

```
<!-- 리액션 결과에 따른 버튼 모양 변화 -->
<script>
  const reactionPick = () =>{
    if(reaction == 1){
      $(".goodReaction").css("color","red");
      $(".badReaction").css("color","black");
    }else if(reaction == -1){
      $(".goodReaction").css("color","black");
      $(".badReaction").css("color","red");
    }else if(reaction == 0){
      $(".goodReaction").css("color","black");
      $(".badReaction").css("color","black");
    }
  }
  $(function() {
    reactionPick();
  })
</script>

<!-- 로그인 경고 -->
<script>
  const loginAlert = () => {
    alert("로그인해주세요");
  }
  $(function() {
    loginAlert();
  })
</script>
```

05 각 기능 소개

articleController

Service 에 글 조회 요청

```
/**
 * 게시글 상세보기
 * 게시글을 가져온다
 * body 부분 줄 바꿈 부분 변환
 * 좋아요, 싫어요 조회
 * @param model
 * @param id
 * @param listUri
 * @return String 상세보기 페이지 이동
 */
@RequestMapping("/usr/article/detail")
public String showDetail( Model model, int id, String listUri) {
    Article article = articleService.getForPrintArticle(id);

    String setBody = article.getBody().replaceAll( "<br>", "\r\n");

    ResultData<Integer> getReactionResultRd = reactionService.getReactionResult(id,rq.getLoginedMemberId());

    List<Comment> comments = commentService.getForPrintComments(id);

    Integer reactionRd = 0;
    if(getReactionResultRd.getData1()!=null) {
        reactionRd = getReactionResultRd.getData1();
    }

    model.addAttribute("listUri", Ut.getUriEncoded(listUri));
    model.addAttribute("comments", comments);
    model.addAttribute("article", article);
    model.addAttribute("setBody", setBody);
    model.addAttribute("reactionRd",reactionRd);
    return "usr/article/detail";
}
```

05 각 기능 소개

articleService

```
/**
 * 글 번호로 게시글 조회
 * @param id
 * @return Article
 */
public Article getForPrintArticle(int id) {
    Article article = articleRepository.getForPrintArticle(id);
    return article;
}
```

ReactionService

```
public ResultData<Integer> getReactionResult(int relId, int memberId) {
    return ResultData.from("S-1", "리액션조회성공", "reactionPointRd", reactionRepository.getReactionResult(relId, memberId));
}
```

commentService

```
public List<Comment> getForPrintComments(int id) {
    List<Comment> comments = commentRepository.getComments(id);
    return comments;
}
```

05 각 기능 소개

articleRepository

```
<select id="getForPrintArticle" resultType="Article">
    SELECT A.*,
    M.name AS extra__writerName
    FROM article AS A
    LEFT JOIN
    `member` AS M
    ON A.memberId = M.id
    WHERE A.id = #{id}
</select>
```

commentRepository

```
@Select("""
    SELECT C.*, M.name AS
    extra__writerName
    FROM `comment` AS C
    INNER JOIN
    `member` AS M
    ON C.memberId
    = M.id
    WHERE C.relId = #{relId}
    AND C.delStatus = 0
    ORDER BY C.id ASC;
""")
public List<Comment> getComments(int relId);
```

reactionRepository

```
@Select("""
    SELECT `point` FROM
    reactionPoint WHERE
    relId = #{relId}
    AND memberId = #{memberId};
""")
public Integer getReactionResult(int relId, int memberId);
```

05 각 기능 소개

reactionController

회원이 아무 리엑션을 하지 않았을 경우

또는 리엑션을 했지만 취소한 경우

```
@RequestMapping("/usr/reaction/doReaction")
@ResponseBody
public ResultData<Integer> reactionPoint(int relId, int point) {

    ResultData<Integer> getReactionResult = reactionService.getReactionResult(relId,rq.getLoginedMemberId());

    if(getReactionResult.getData1() == null){
        ResultData<Integer> doReactionRd = reactionService.doReaction(relId,rq.getLoginedMemberId(),point);
        reactionService.updateReaction(relId);
        ResultData<Integer> rd = ResultData.newData(doReactionRd, "Reaction", point);
        Article article = reactionService.getReactionPoint(relId);
        rd.setData2("goodReaction", article.getGoodReactionPoint());
        rd.setData3("badReaction", article.getBadReactionPoint());
        return rd;
    }
    if(getReactionResult.getData1() == 0) {
        ResultData<Integer> doReactionRd = reactionService.changeReaction(relId, rq.getLoginedMemberId(),point);
        reactionService.updateReaction(relId);
        ResultData<Integer> rd = ResultData.newData(doReactionRd, "Reaction", point);
        Article article = reactionService.getReactionPoint(relId);
        rd.setData2("goodReaction", article.getGoodReactionPoint());
        rd.setData3("badReaction", article.getBadReactionPoint());
        return rd;
    }
}
```


05 각 기능 소개

reactionController

좋아요를 눌렀을 경우

싫어요를 눌렀을 경우

```
if(getReactionResult.getData1() == 1) {
    ResultData<Integer> doReactionRd = new ResultData<Integer>();

    if(point == 1) {
        doReactionRd = reactionService.changeReaction(relId, rq.getLoginedMemberId(),0);
        reactionService.updateReaction(relId);
    }
    if(point == -1) {
        doReactionRd = reactionService.changeReaction(relId, rq.getLoginedMemberId(),point);
        reactionService.updateReaction(relId);
    }
    ResultData<Integer> rd = ResultData.newData(doReactionRd, "Reaction", point);
    Article article = reactionService.getReactionPoint(relId);
    rd.setData2("goodReaction", article.getGoodReactionPoint());
    rd.setData3("badReaction", article.getBadReactionPoint());
    return rd;
}
if(getReactionResult.getData1() == -1) {
    ResultData<Integer> doReactionRd = new ResultData<Integer>();
    if(point == 1) {
        doReactionRd = reactionService.changeReaction(relId, rq.getLoginedMemberId(),point);
        reactionService.updateReaction(relId);
    }
    if(point == -1) {
        doReactionRd = reactionService.changeReaction(relId, rq.getLoginedMemberId(),0);
        reactionService.updateReaction(relId);
    }
    ResultData<Integer> rd = ResultData.newData(doReactionRd, "Reaction", point);
    Article article = reactionService.getReactionPoint(relId);
    rd.setData2("goodReaction", article.getGoodReactionPoint());
    rd.setData3("badReaction", article.getBadReactionPoint());
    return rd;
}
```

05 각 기능 소개

ReactionService

처음 추천 기능을 사용했을 경우

```
public ResultData<Integer> doReaction(int relId, int memberId, int point){
    int reactionPointRd = reactionRepository.doReaction(relId,memberId,point);
    if(point == -1) {
        return ResultData.from("S-1", "싫어요", "reactionPointRd", reactionPointRd);
    }
    return ResultData.from("S-2", "좋아요", "reactionPointRd", reactionPointRd);
}
```

이미 기능을 사용했을 때 변경

```
public ResultData<Integer> changeReaction(int relId, int memberId, int point) {
    if(point == -1) {
        return ResultData.from("S-1", "싫어요", "reactionPointRd",reactionRepository.changeReaction(relId, memberId, point));
    }
    return ResultData.from("S-2", "좋아요", "reactionPointRd",reactionRepository.changeReaction(relId, memberId, point));
}
```

05 각 기능 소개

ReactionRepository

```
@Insert("""
    INSERT INTO reactionPoint
    SET regDate = NOW(),
    updateDate = NOW(),
    memberId = #{memberId},
    relTypeCode = 'article',
    relId = #{relId},
    `point` = #{point}
    """)
public int doReaction(int relId, int memberId, int point);
```

```
@Delete("""
    UPDATE reactionPoint SET
    `point` = #{point},
    updateDate = NOW()
    WHERE relId= #{relId}
    AND memberId = #{memberId};
    """)
public Integer changeReaction(int relId, int memberId, int point);
```

05 각 기능 소개

commentController

```
@RequestMapping("/usr/comment/doAdd")
@ResponseBody
public String doAdd(int id, String comment, String relTypeCode, String listUri) {

    if(Ut.empty(comment)) {
        return Ut.jsHistoryBack(Ut.f("댓글을 입력해주세요"));
    }

    ResultData commentRd = commentService.doWrite(id,rq.getLoginedMemberId(),comment, relTypeCode);

    return Ut.jsReplace(Ut.f(""), Ut.f("../article/detail?id=%d&listUri=%s", id,listUri));
}
```

```
@RequestMapping("/usr/comment/doDelete")
@ResponseBody
public String doDelete(int id, int relId , String listUri ) {

    ResultData commentRd = commentService.doDelete(id);
    ResultData<Integer> rd = ResultData.newData(commentRd, "Reaction", id);

    return Ut.jsReplace(Ut.f(""), Ut.f("../article/detail?id=%d&listUri=%s", relId,Ut.getUriEncoded(listUri)));
}
```

```
@RequestMapping("/usr/comment/doModify")
@ResponseBody
public String doModify(int id, String comment, int relId , String listUri) {

    ResultData commentRd = commentService.doModify(id, comment);
    ResultData<Integer> rd = ResultData.newData(commentRd, "Reaction", id);

    return Ut.jsReplace(Ut.f(""), Ut.f("../article/detail?id=%d&listUri=%s", relId,listUri));
}
```

05 각 기능 소개

commentService

```
public ResultData doWrite(int id, int memberId ,String comment, String relTypeCode) {  
    commentRepository.doWrite(id,memberId,comment, relTypeCode);  
    return ResultData.from("S-1", Ut.f("댓글 작성 성공"));  
}  
  
public ResultData doDelete(int id) {  
    int deleteRd = commentRepository.doDelete(id);  
    return ResultData.from("S-1", "삭제성공", "deleteRd", deleteRd);  
}  
  
public ResultData doModify(int id, String comment) {  
    int modifyRd = commentRepository.doModify(id,comment);  
    return ResultData.from("S-1", "수정성공", "modifyRd", modifyRd);  
}
```

05 각 기능 소개

commentRepository

```
@Insert("""
    INSERT INTO `comment`
    SET regDate = NOW(),
    updateDate = NOW(),
    memberId = #{memberId},
    relId = #{relId},
    relTypeCode = #{relTypeCode},
    `comment` = #{comment}
    """)
public int doWrite(int relId, int memberId, String comment, String relTypeCode);

@Delete("""
    UPDATE comment SET
    delStatus = 1
    WHERE id = #{id}
    """)
public int doDelete(int id);

@update("""
    <script>
    UPDATE `comment`
    <set>
        <if test="comment != null and comment != ''">`comment` = #{comment},</if>
    updateDate = NOW()
    </set>
    WHERE id = #{id}
    </script>
    """)

public int doModify(int id, String comment);
```

글 수정 페이지

공지사항4

H B I A* S | — 66 | ☰ ☷ ☑ ☲ ☳ | 📑 🖼️ 🔗 | </> CB | Scroll ☐

□L□O□L□O□L□O~

□L□O□L□O□L□O□L□O

□L□O□L□O□L□O

□L□O□L□O□L□O□L□O

Markdown

WYSIWYG

수정 완료

05 각 기능 소개

articleController

```
/**
 * 글 수정 폼
 * 수정 전 게시글 정보 조회
 * 수정 권한 체크
 * @param model
 * @param id
 * @param listUri
 * @return String 게시글 수정 폼 이동
 */
@RequestMapping("usr/article/modify")
public String articleModifyForm( Model model,int id, String listUri) {
    Article article = articleService.getForPrintArticle( id);
    String setBody = article.getBody().replaceAll( "<br>","\r\n");
    if (article.getMemberId() != rq.getLoginedMemberId()) {
        return rq.jsHistoryBackOnView("권한이 없습니다.");
    }
    model.addAttribute("listUri", Ut.getUriEncoded(listUri));
    model.addAttribute("article", article);
    model.addAttribute("setBody", setBody);
    return "usr/article/modify" ;
}
```

```
/**
 * 글 수정
 * 글 존재 여부 확인
 * 권한 체크
 * 제목, 내용 체크
 * 서비스에 수정 요청
 * @param id
 * @param title
 * @param body
 * @param listUri
 * @return String 글 수정실행
 */
@RequestMapping("/usr/article/doModify")
@ResponseBody
public String doModify(int id, String title, String body, String listUri) {
    Article article = articleService.getForPrintArticle(id);
    if (article == null) {
        return Ut.jsHistoryBack(Ut.f("%d번 게시글은 존재하지 않습니다", id));
    }
    if (article.getMemberId() != rq.getLoginedMemberId()) {
        return Ut.jsHistoryBack(Ut.f("%d번 게시글에 대한 권한이 없습니다.", id));
    }
    if(Ut.empty(title)) {
        return Ut.jsHistoryBack(Ut.f("제목을 입력해주세요"));
    }
    if(Ut.empty(body)) {
        return Ut.jsHistoryBack(Ut.f("내용을 입력해주세요"));
    }
    articleService.modifyArticle(id, title, body);
    return Ut.jsReplace(Ut.f("%d번 게시글을 수정했습니다", id), Ut.f("../article/detail?id=%d&listUri=%s", id,listUri));
}
```


05 각 기능 소개

articleService

```
/**
 * 글 수정
 * @param id
 * @param title
 * @param body
 * @return ResultData<Article> 수정된 게시글
 */
public ResultData<Article> modifyArticle(int id, String title, String body) {
    articleRepository.modifyArticle(id, title, body);

    Article article = getForPrintArticle(id);

    return ResultData.from("S-1", Ut.f("%d번 게시물을 수정했습니다", id), "article", article);
}
```

05 각 기능 소개

articleRepository

```
<update id="modifyArticle">
  UPDATE article
  <set>
    <if test="title != null and title != ''">title = #{title},</if>
    <if test="body != null and body != ''">`body` = #{body},</if>
    updateDate = NOW()
  </set>
  WHERE id = #{id}
</update>
```

05 각 기능 소개

교육일정 페이지

Fullcalendar API

EduManager

- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정
- 관리자 페이지

로그아웃

Dark Mode ☐



직원님 접속 중

< > today

2022년 12월

month week day

일	월	화	수	목	금	토
27일	28일	29일	30일	1일	2일	3일
22-6기						
4일	5일	6일	7일	8일	9일	10일
11일	12일	13일	14일	15일	16일	17일
18일	19일	20일	21일	22일	23일	24일
25일	26일	27일	28일	29일	30일	31일
1일	2일	3일	4일	5일	6일	7일

05 각 기능 소개

Calendar.jsp

Ajax 로 교육과정을 불러와서 달력에 표현

```
<script>
document.addEventListener('DOMContentLoaded', function() {
    var calendarEl = document.getElementById('calendar');
    var calendar = new FullCalendar.Calendar(calendarEl, {
        // Tool Bar 목록 document : https://fullcalendar.io/docs/toolbar
        headerToolbar: {
            left: 'prev,next,today',
            center: 'title',
            right: 'dayGridMonth,dayGridWeek,dayGridDay'
        },
        selectable: true,
        selectMirror: true,

        navLinks: true, // can click day/week names to navigate views
        editable: true,
        locale: 'ko',
        // Create new event
        dayMaxEvents: true, // allow "more" link when too many events
        // 이벤트 객체 필드 document : https://fullcalendar.io/docs/event-object
        events:[ ]

    });

    function getEducationCourse() {
        $.get('../educationCourse/schedule', {
            ajaxMode : 'Y'
        }, function(data) {
            for(var i of data.data1){
                var courseUrl = 'http://127.0.0.1:8081/usr/educationCourse/detail?id='+i.id;
                calendar.addEvent({title:i.title,start:i.start,end:i.end,url:courseUrl});
            }
        });
    };
    getEducationCourse();
    calendar.render();
});
```

05 각 기능 소개

관리자 권한 이상은 새 교육과정을 추가할 수 있다.

EduManager

내 정보

나의 교육정보

공지사항

자유게시판

Q&A

교육일정

관리자 페이지

로그아웃

Dark Mode

2022년 12월

month week day

일	월	화	수	목	금	토
27일	28일	29일	30일	1일	2일	3일
22-671						
4일	5일	6일	7일	8일	9일	10일
11일	12일	13일	14일	15일	16일	17일
18일	19일	20일	21일	22일	23일	24일
asdasd						
25일	26일	27일	28일	29일	30일	31일
1일	2일	3일	4일	5일	6일	7일

새 교육과정

05 각 기능 소개

새 교육과정 추가 모달창

일	월	화	수	목	
27일	28일	29일	30일	1일	
22-6기					
4일				8일	
11일				15일	
18일				22일	
asdasd					
25일				29일	
1일	2일	3일	4일	5일	

과정명

교육시작일

년-월-일

교육종료일

년-월-일

교육장소

담당자

관리자

추가

닫기

05 각 기능 소개

추가된 과정은 달력에 반영

EduManager

- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정
- 관리자 페이지

[>] 로그아웃

Dark Mode ☐

<

>

today

2022년 12월

새로운 캡처를 실행합니다. (Ctrl+N)

month

week


day


일	월	화	수	목	금	토
27일	28일	29일	30일	1일	2일	3일
22-6기						
4일	5일	6일	7일	8일	9일	10일
	테스트					
11일	12일	13일	14일	15일	16일	17일
18일	19일	20일	21일	22일	23일	24일
asdasd						
25일	26일	27일	28일	29일	30일	31일
1일	2일	3일	4일	5일	6일	7일


새 교육과정


교육과정 상세보기


관리자 권한 화면


 **EduManager**


 내 정보


 나의 교육정보


 공지사항


 자유게시판

 Q&A

 교육일정

 관리자 페이지

 로그아웃

 Dark Mode ☐



과정명

테스트

과정 등록일

2022-12-13

교육시작일

2022-12-06

교육종료일

2022-12-10

교육장소

서울

담당자

직원

현재상태

진행중

교육과정 수정

삭제하기

교육과정 상세보기

로그아웃화면

EduManager



- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정

→ 로그인

Dark Mode ☐

로그아웃 중

과정명

테스트

과정 등록일

2022-12-13

교육시작일

2022-12-06

교육종료일

2022-12-10

교육장소

서울

담당자


직원

현재상태

진행중

교육과정 상세보기

일반회원 화면


 EduManager



-  내 정보
-  나의 교육정보
-  공지사항
-  자유게시판
-  Q&A
-  교육일정

[→] 로그아웃

 Dark Mode ☐

 사용자5 님 접속 중

과정명

테스트

과정 등록일

2022-12-13

교육시작일

2022-12-06

교육종료일

2022-12-10

교육장소

서울

담당자

직원

현재상태

진행중

수강하기

05 각 기능 소개

교육과정 상세보기

수정하기

등록일 외에 수정 가능

127.0.0.1:8081의 메시지

8번 교육과정 수정

확인

EduManager

- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정
- 관리자 페이지

로그아웃

Dark Mode

과정명

테스트

과정 등록일

2022-12-13

교육/시각일

2022-12-06

교육종료일

2022-12-10

교육장소

서울

담당자

직원

현재상태

진행중

수정 완료

취소

삭제하기

직원 남 접속 중

교육과정 삭제

127.0.0.1:8081의 메시지

교육과정을 삭제 했습니다.

확인

05 각 기능 소개

educationCourseController

```
@RequestMapping("usr/educationCourse/create")
@ResponseBody
public String create(String title, String startOfEducation, String endOfEducation, String place, int managerMemberId ) {

    if(Ut.empty(startOfEducation)) {
        return Ut.jsHistoryBack(Ut.f("교육 시작일을 지정해주세요"));
    }
    if(Ut.empty(endOfEducation)) {
        return Ut.jsHistoryBack(Ut.f("교육 종료일을 지정해주세요"));
    }
    if(Ut.empty(title)) {
        return Ut.jsHistoryBack(Ut.f("교육과정명을 입력해주세요"));
    }
    if(Ut.empty(place)) {
        return Ut.jsHistoryBack(Ut.f("교육 장소를 입력해주세요"));
    }
    if(Ut.empty(managerMemberId)) {
        return Ut.jsHistoryBack(Ut.f("담당자를 선택해주세요"));
    }

    ResultData rd = educationCourseService.create(title, startOfEducation, endOfEducation, place, managerMemberId);

    return Ut.jsReplace(rd.getMsg(), "../educationCourse/calendar");
}

@RequestMapping("usr/educationCourse/delete")
@ResponseBody
public String delete(int educationCourseId) {

    ResultData rd = educationCourseService.delete(educationCourseId);

    return Ut.jsReplace(rd.getMsg(), "../educationCourse/calendar");
}
```

05 각 기능 소개

educationCourseController

```
@RequestMapping("usr/educationCourse/doModify")
@ResponseBody
public String doModify(
    int educationCourseId,
    String startOfEducation,
    String endOfEducation,
    String title,
    String place,
    int managerMemberId,
    int status, |
    String educationCourseModifyAuthKey) {

    if(educationCourseModifyAuthKey == null) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 생성되지 않았습니다."));
    }
    Attr getAttr = attrService.get("educationCourse", educationCourseId, "extra", "educationCourseModifyAuthKey");

    if(getAttr==null) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 만료되었습니다. 다시 시도해주세요"));
    }
    if(!getAttr.getValue().equals(educationCourseModifyAuthKey)) {
        return Ut.jsHistoryBack(Ut.f("인증코드가 일치하지 않습니다. 다시 시도해주세요"));
    }

    startOfEducation = startOfEducation.replaceAll("-", "");
    endOfEducation = endOfEducation.replaceAll("-", "");

    if(Ut.empty(startOfEducation)) {
        return Ut.jsHistoryBack(Ut.f("교육 시작일을 지정해주세요"));
    }
    if(Ut.empty(endOfEducation)) {
        return Ut.jsHistoryBack(Ut.f("교육 종료일을 지정해주세요"));
    }
    if(Ut.empty(title)) {
        return Ut.jsHistoryBack(Ut.f("교육과정명을 입력해주세요"));
    }
    if(Ut.empty(place)) {
        return Ut.jsHistoryBack(Ut.f("교육 장소를 입력해주세요"));
    }

    ResultData rd = educationCourseService.doModify(educationCourseId, startOfEducation, endOfEducation, title, place, managerMemberId, status);
    return Ut.jsReplace(Ut.f("%d번 교육과정 수정", educationCourseId), Ut.f("../educationCourse/detail?id=%d", educationCourseId));
}
```

05 각 기능 소개

educationCourseService

```
public ResultData doModify(  
    int educationCourseId,  
    String startOfEducation,  
    String endOfEducation,  
    String title,  
    String place,  
    int managerMemberId,  
    int status) {  
  
    educationCourseRepository.doModify(educationCourseId, startOfEducation, endOfEducation, title, place, managerMemberId, status);  
    return ResultData.from("S-1", "교육과정 수정 성공", "id", educationCourseId);  
}
```

```
public ResultData create(String title, String startOfEducation, String endOfEducation, String place,  
    int managerMemberId) {  
  
    EducationCourse educationCourse = educationCourseRepository.getEducationCourseByTitle(title);  
    if (educationCourse != null) {  
        return ResultData.from("F-1", Ut.f("이미 등록된 과정입니다."));  
    }  
  
    educationCourseRepository.create(title, startOfEducation, endOfEducation, place, managerMemberId);  
  
    return ResultData.from("S-1", Ut.f("교육과정 등록 성공"));  
}  
public ResultData delete(int educationCourseId) {  
  
    EducationCourse educationCourse = educationCourseRepository.getEducationCourse(educationCourseId);  
    if (educationCourse == null) {  
        return ResultData.from("F-1", Ut.f("존재하지 않는 교육과정입니다."));  
    }  
    educationCourseRepository.delete(educationCourseId);  
    return ResultData.from("S-1", Ut.f("교육과정을 삭제 했습니다."));  
}
```

05 각 기능 소개

EducationRepository

```
<update id="doModify">
    UPDATE `educationCourse` SET
        updateDate = NOW(),
        startOfEducation = DATE("#{startOfEducation}"),
        endOfEducation = DATE("#{endOfEducation}"),
        title = #{title},
        place = #{place},
        managerMemberId = #{managerMemberId},
        `status` = #{status}
    WHERE id = #{educationCourseId};
</update>
```

```
<insert id="create">
    INSERT INTO educationCourse
    SET regDate = NOW(),
        updateDate = NOW(),
        startOfEducation = DATE("#{startOfEducation}"),
        endOfEducation = DATE("#{endOfEducation}"),
        title = #{title},
        place = #{place},
        managerMemberId = #{managerMemberId}
</insert>

<delete id="delete">
    DELETE FROM educationCourse
    WHERE id = #{educationCourseId};
</delete>
```


05 각 기능 소개

수강 신청 후 나의 교육정보 화면이동

127.0.0.1:8081의 메시지
수강신청되었습니다.

확인

EduManager



사용자5 님 접속 중

내 정보

나의 교육정보

공지사항

자유게시판

Q&A

교육일정

수강중인 교육 리스트

22-5기

교육기간:2022-09-01~2022-09-30

진행중

담당자: 직원

담당자 연락처: 0101235678

수강 취소

테스트

교육기간:2022-12-06~2022-12-10

진행중

담당자: 직원


담당자 연락처: 0101235678

수강 취소


로그아웃

Dark Mode

교육 리스트 수강 취소

 EduManager



 사용자33 님 접속 중

-  내 정보
-  나의 교육정보
-  공지사항
-  자유게시판
-  Q&A
-  교육일정

[→] 로그아웃

 Dark Mode ☐

수강중인 교육 리스트

22-671

교육기간:2022-11-01~2022-11-30

진행중

담당자: 직원

담당자 연락처: 0101235678

수강 취소

05 각 기능 소개

수강취소 confirm

127.0.0.1:8081의 메시지

정말 취소하시겠습니까?

확인

취소

127.0.0.1:8081의 메시지

수강취소되었습니다.

확인

05 각 기능 소개

educationCourseController

```
@RequestMapping("usr/educationCourse/doDelete")
@ResponseBody
public String doDelete(int educationCourseId) {

    ResultData rd = educationCourseService.doDelete(educationCourseId, rq.getLoginMemberId());

    return Ut.jsReplace(rd.getMsg(), "../educationCourse/eduStatus");
}
```

educationCourseService

```
public ResultData doDelete(int educationCourseId, int memberId) {

    registInfo rd = educationCourseRepository.getRegistInfo(educationCourseId, memberId);

    if(rd == null) {
        return ResultData.from("F-1", Ut.f("수강중인 과정이 아닙니다."));
    }

    educationCourseRepository.doDelete(educationCourseId, memberId);

    return ResultData.from("S-1", "수강취소되었습니다.", "id", educationCourseId);
}
```

educationCourseRepository

```
<delete id="doDelete">
    DELETE FROM registInfo
    WHERE courseId = #{educationCourseId}
    AND memberId = #{memberId};
</delete>
```

관리자 권한의 회원관리 memberList



- 내 정보
- 나의 교육정보
- 공지사항
- 자유게시판
- Q&A
- 교육일정
- 관리자 페이지

회원 관리

회원 수: 12명

이름 ▾

검색어를 입력해주세요

Q 검색

번호	가입일자	생년월일	아이디	이름	전화번호	이메일	<input type="checkbox"/>
12	12-12 22:51	2001-11-03	id11	사용자11	01052312322	a22213@gmail.com	<input type="checkbox"/>
11	12-12 22:51	2001-11-03	id10	사용자10	01012344312	aqd33@gmail.com	<input type="checkbox"/>
10	12-12 22:51	2001-11-03	id9	사용자9	01051231232	a21afgf3@gmail.com	<input type="checkbox"/>
9	12-12 22:51	2001-11-03	id8	사용자8	01056789012	a21agf3@gmail.com	<input type="checkbox"/>
8	12-12 22:51	1999-10-03	id7	사용자7	01012123678	asddd@gmail.com	<input type="checkbox"/>
7	12-12 22:51	2001-11-03	id6	사용자6	01052390122	a2adfg13@gmail.com	<input type="checkbox"/>
6	12-12 22:51	2001-11-03	id5	사용자5	01056344312	a2sadfg3@gmail.com	<input type="checkbox"/>
5	12-12 22:51	2001-11-03	id4	사용자4	01051234012	a21adfgdf3@gmail.com	<input type="checkbox"/>
4	12-12 22:51	2001-11-03	id3	사용자3	01056789012	a21adfg33@gmail.com	<input type="checkbox"/>
3	12-12 22:51	1999-10-03	id2	사용자2	01012345678	asddd@gmail.com	<input type="checkbox"/>

선택삭제

회원 이름 검색, 체크박스 선택 삭제 기능

회원 관리

회원 수: 12명

이름 ▾

검색어를 입력해주세요

🔍 검색

번호	가입일자	생년월일	아이디	이름	전화번호	이메일	<input type="checkbox"/>
12	12-12 22:51	2001-11-03	id11	사용자11	01052312322	a22213@gmail.com	<input checked="" type="checkbox"/>
11	12-12 22:51	2001-11-03	id10	사용자10	01012344312	aqd33@gmail.com	<input checked="" type="checkbox"/>
10	12-12 22:51	2001-11-03	id9	사용자9	01051231232	a21afgf3@gmail.com	<input checked="" type="checkbox"/>
9	12-12 22:51	2001-11-03	id8	사용자8	01056789012	a21agf3@gmail.com	<input checked="" type="checkbox"/>
8	12-12 22:51	1999-10-03	id7	사용자7	01012123678	asddd@gmail.com	<input checked="" type="checkbox"/>
7	12-12 22:51	2001-11-03	id6	사용자6	01052390122	a2adfg13@gmail.com	<input checked="" type="checkbox"/>
6	12-12 22:51	2001-11-03	id5	사용자5	01056344312	a2sadfg3@gmail.com	<input checked="" type="checkbox"/>
5	12-12 22:51	2001-11-03	id4	사용자4	01051234012	a21adfgdf3@gmail.com	<input checked="" type="checkbox"/>
4	12-12 22:51	2001-11-03	id3	사용자3	01056789012	a21adfg33@gmail.com	<input checked="" type="checkbox"/>
3	12-12 22:51	1999-10-03	id2	사용자2	01012345678	asddd@gmail.com	<input checked="" type="checkbox"/>

🗑 선택삭제

1

2

05 각 기능 소개

회원 삭제 기능

127.0.0.1:8081의 메시지

정말 삭제하시겠습니까?

확인

취소

127.0.0.1:8081의 메시지

해당 회원이 삭제되었습니다.

확인

05 각 기능 소개

Check box 전체 체크

```
<script>
$('.checkbox-all-member-id').change(function() {
  const $all = $(this);
  const allChecked = $all.prop('checked');
  $('.checkbox-member-id').prop('checked', allChecked);
});
$('.checkbox-member-id').change(function() {
  const checkboxMemberIdCount = $('.checkbox-member-id').length;
  const checkboxMemberIdCheckedCount = $('.checkbox-member-id:checked').length;
  const allChecked = checkboxMemberIdCount == checkboxMemberIdCheckedCount;
  $('.checkbox-all-member-id').prop('checked', allChecked);
});
</script>
```

회원 탈퇴 script

```
<script>
$('.btn-delete-selected-members').click(function() {
  const values = $('.checkbox-member-id:checked').map((index, el) => el.value).toArray();
  if ( values.length == 0 ) {
    alert('삭제할 회원을 선택 해주세요.');
    return;
  }
  if ( confirm('정말 삭제하시겠습니까?') == false ) {
    return;
  }
  document['do-delete-members-form'].ids.value = values.join(',');
  document['do-delete-members-form'].submit();
});
</script>
```


05 각 기능 소개

memberController

```
@RequestMapping("usr/member/memberList")
public String memberList(
    @RequestParam(defaultValue = "1") Integer page,
    @RequestParam(defaultValue = "") String searchWord,
    @RequestParam(defaultValue = "") String searchFrom,
    Model model) {

    int itemsInAPage = 10;
    int limitFrom = (page - 1) * itemsInAPage;

    List<Member> memberList = memberService.getMemberList(limitFrom, itemsInAPage, searchWord, searchFrom);

    int getTotalMember = memberService.getTotalMember(searchWord, searchFrom);
    int pageCount = (int) Math.ceil((double) getTotalMember / itemsInAPage);

    model.addAttribute("memberList", memberList);
    model.addAttribute("searchFrom", searchFrom);
    model.addAttribute("searchWord", searchWord);
    model.addAttribute("getTotalMember", getTotalMember);
    model.addAttribute("page", page);
    model.addAttribute("pageCount", pageCount);
    return "/usr/member/memberList";
}
```

05 각 기능 소개

memberService

```
public List<Member> getMemberList(int limitFrom, int itemsInAPage, String searchWord, String searchFrom) {  
    return memberRepository.getMemberList(limitFrom, itemsInAPage, searchWord, searchFrom);  
}
```

memberRepository

```
<select id="getMemberList" resultType="Member">  
SELECT * FROM `member`  
where delStatus = 0  
<if test="searchWord != ''">  
    AND ${searchFrom}  
    LIKE '%${searchWord}%'  
</if>  
    ORDER BY id DESC  
    LIMIT #{limitFrom} , #{itemsInAPage}  
</select>
```

05 각 기능 소개

인터셉터 적용

```
// /resource/common.css
// 인터셉터 적용
public void addInterceptors(InterceptorRegistry registry) {
    |
    InterceptorRegistration ir;

    ir = registry.addInterceptor(beforeActionInterceptor)
        .addPathPatterns("/**")
        .addPathPatterns("/favicon.ico")
        .excludePathPatterns("/resource/**")
        .excludePathPatterns("/error");

    ir = registry.addInterceptor(needLoginInterceptor)
        .addPathPatterns("/usr/member/doLogout")
        .addPathPatterns("/usr/member/userInfo")
        .addPathPatterns("/usr/member/doModify")
        .addPathPatterns("/usr/member/quitMember")
        .addPathPatterns("/usr/member/memberList")
        .addPathPatterns("/usr/article/doAdd")
        .addPathPatterns("/usr/article/write")
        .addPathPatterns("/usr/article/modify")
        .addPathPatterns("/usr/article/doModify")
        .addPathPatterns("/usr/article/doDelete")
        .addPathPatterns("/usr/reaction/doReaction")
        .addPathPatterns("/usr/comment/doAdd")
        .addPathPatterns("/usr/comment/doDelete")
        .addPathPatterns("/usr/comment/doModify")
        .addPathPatterns("/usr/educationCourse/eduStatus")
        .addPathPatterns("/usr/educationCourse/doModify")
        .addPathPatterns("/usr/educationCourse/create");

    ir = registry.addInterceptor(needAdministratorInterceptor)
        .addPathPatterns("/usr/member/memberList")
        .addPathPatterns("/usr/educationCourse/doModify")
        .addPathPatterns("/usr/educationCourse/create");

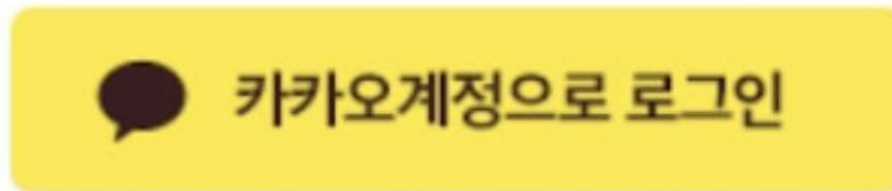
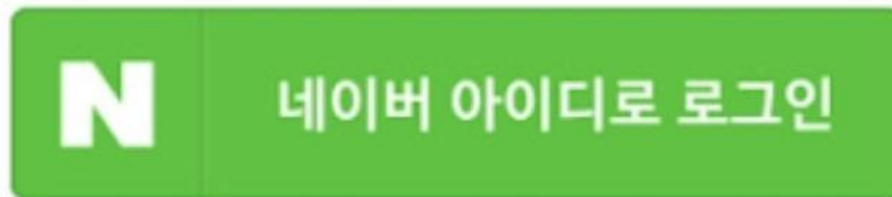
}
```

06

추가 될 기능

05 각 기능 소개

네이버, 카카오 등 로그인 API 사용



PILOT CERTIFICATE																				
1. Name: [REDACTED]				Department: [REDACTED]				3. Date of Birth (Resident Registration No./Passport No.): [REDACTED].***												
DATE	A/C TYPE	NR LDG	TYPE OF PILOTING TIME							CONDITIONS OF FLIGHT										ETC.
			PIC	CO PILOT		IP	STUDENT	FLT ENGINEER	TOTAL	DAY				NIGHT				INSTRUMENT		
				PIC UNDER SUPERVISION	SIC					VFR	CROSS COUNTRY		VFR	CROSS COUNTRY		ACTUAL INST.	SIMULATED INST.			
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
TOTAL			[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

This is to certify the above information pursuant to Article 79 of the 「Enforcement Rules of the Aviation Act」.

Issue Date: M[REDACTED] 4, 2017 Issued by/Address: [REDACTED] Inc.
[REDACTED], Seoul, Korea Issuer: President S[REDACTED] [REDACTED]
Telephone: [REDACTED]


갤러리 게시판


갤러리 게시판


HOME / 갤러리 게시판


제목 ▼


검색





스위스 여행-8
01:52 | 0
시그달 





스위스 여행-7
01:37 | 0
시그달 





스위스 여행-6
00:37 | 1
시그달 





스위스 여행-5
00:36 | 1
시그달 





스위스 여행-4
00:36 | 1
시그달 




스위스 여행-3
00:35 | 1
시그달 



스위스 여행-2
00:35 | 2
시그달 



스위스 여행-1
00:35 | 1
시그달 

글쓰기

05 각 기능 소개

달력 기능 확장

Editor link

Locale
en

Air Calendar Demo

Timezone
(GMT-4) America/Havana

Today < > April 2019 Month

< April, 2019 >

SU	MO	TU	WE	TH	FR	SA
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Today

☐ Enable editing blocked events

Browser locale : en-US
Calendar locale : en

Browser timezone:
America/New_York
Calendar timezone:
America/Havana

Disclaimer: This demo shows an example of what you can build with this plugin. But this is not a default of the calendar. You have the complete freedom to design whatever you like. The context menu popup is not part of the plugin. You can build your own using Group Focus (requires some work) or Popup

W	Sun	Mon	Tue	Wed	Thu	Fri	Sat
14	31	1	2	3	4	5	6
15	7	8	9	10	11	12	13
16	14	15	16	17	18	19	20
17	21	22	23	24	25	26	27
18	28	29	30	1	2	3	4
19	5	6	7	8	9	10	11