포트폴리오

황재원

jaewon9937@gmail.com
010-9937-5580

이름 : 황재원 (Hwang Jae Won)

나이 : 만 26세 (1993.09.13)

전화번호 : 010 - 9937 - 5580

이메일 : jaewon9937@gmail.com

Git : https://github.com/jaewon0913

학력 : 한서대학교 졸업
(2012 - 2019, 항공소프트웨어공학과)

기타 사항
- KH 정보교육원 수료
  (Java 프레임워크 개발자 과정 6개월, 2019)
- Mac pro 13inch 사용하여 개발 공부 ing…
- Eclipse, Intelli J 개발 툴 사용 중

About Me

''신뢰주는 개발자''

어릴 때는 컴퓨터로 게임을 좋아하는 학생이 지금은 개발자로 꿈을 꾸고 있습니다. 그래서 대학교도 소프트웨어학과를 가게 되었고 자연스럽게 개발자에 더욱 다가갈 수 있었습니다.

저는 저의 목표를 “신뢰를 주는” 개발자로 삼아 공부하고 있습니다. 개발을 완료하고 사용자가 사용할 때 예상치 못한 장애가 발생할 경우 고객과 신뢰관계와 연결이 되기 때문에 항상 꼼꼼히 살펴서 최대한 에러가 발생하지 않도록 하겠습니다.

경력은 시간이 지나면 쌓이지만 실력은 항상 공부해야 쌓인다고 생각합니다. 저 또한 계속 공부하여 “신뢰주는” 개발자로서 남도록 하겠습니다.

Skills

Language

<기본> <기본> <기초> <기초>

Web Skill

<기본> <기본> <기본> <기본>

Framework

<기본> <기초> <기초>

DataBase

<기본> <기초>

Operating System

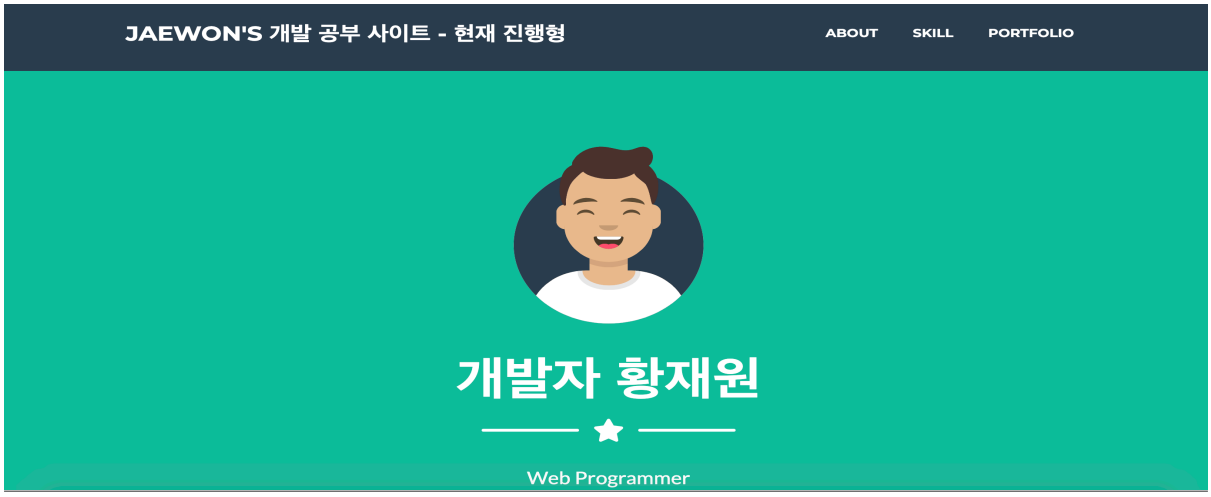
Etc

프로젝트

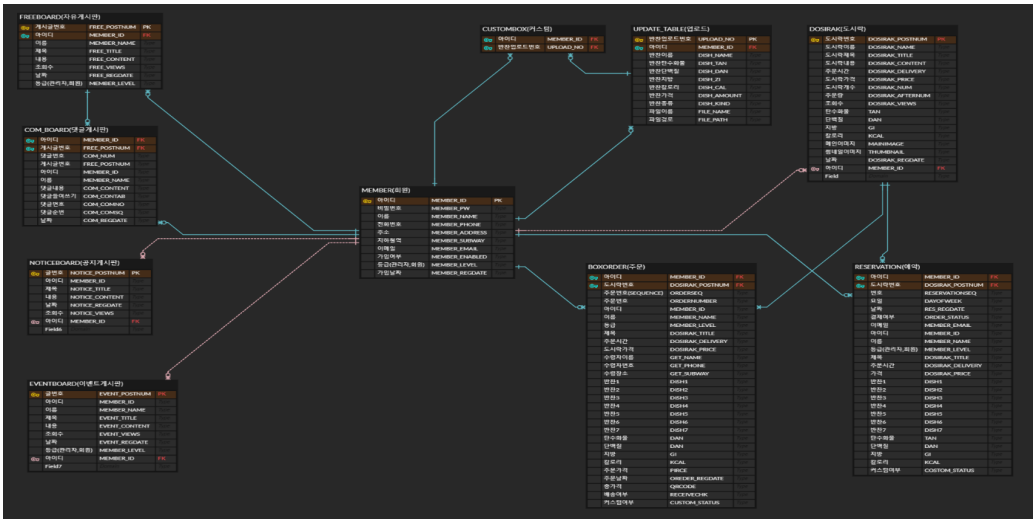
1. SpringBoot + AWS - 현재 진행 중
2. 파이널 프로젝트
3. 세미 프로젝트

SpringBoot + AWS

항목	주요 내용	
프로젝트 명	Blog(포트폴리오 사이트)	
수행 기간 및 인원	2019. 11. 01 ~ (총 1 명)	
개발 목표	SpringBoot를 사용하고 AWS로 웹 서비스 출시 Blog 형식의 사이트 포트폴리오용 개발 사이트 제작	
개발 환경	OS	Mac OS
	IDE	IntelliJ IDEA Ultimate
	구현 언어	Java, HTML5, CSS3, Java Script, MySQL
	사용	SpringBoot(Gradle) , Amazon Web Services(AWS), Git(Github Desktop)
사이트 주소	http://ec2-15-164-249-195.ap-northeast-2.compute.amazonaws.com:8686/	
비고	계속해서 개발 중입니다. 추가적으로 로그인 기능, CRUD, 포트폴리오 추가 등	



파이널 프로젝트

항목	주요 내용	
프로젝트 명	커스텀 도시락 제작 판매 사이트	
수행 기간 및 인원	2019. 08. 02 ~ 2019. 09. 02 (총 6 명)	
개발 목표	바쁜 현대 사회에 직장인들이 식사를 잘 챙겨 먹지 못하고 있다. 이러한 이유로 커스텀 도시락 제작 판매 홈페이지를 제작하여 정해진 시간에 원하는 반찬으로 이루어진 도시락을 지정된 장소에서 받아 규칙적인 식사를 돕는다.	
개발 환경	OS	Window 7 , 10 / Mac OS
	구현 언어	Java, HTML5, CSS3, Java Script, Oracle
	사용 기술	Ajax , 주소 API , 캡차 , QR 코드 , 그래프 API(D3.js), Web Notification API Kakao (Login, Pay, Map) API , Spring (Batch / Security / WebSocket) jQuery(Drag & Drop) , Mail SMPT
	사용 도구	Eclipse , Oracle DataBase 11g Express Edition, Apache Tomcat 9.0, Bootstrap 4.0 , Summer Note
담당 역할	Spring WebSocket / 장바구니(Cookie) / 파일 업로드 / 검색 도시락 커스텀(jQuery[drag & drop]) / 매크로 봇 / 프로젝트 형상 관리(Git)	
구현 기능	로그인	Spring Security와 Kakao Login API를 통한 로그인 기능 제공
	결제	Kakao Pay API를 통한 결제 시스템 제공
	지도	Kakao Map API를 통한 도시락 제공 장소 제공
	검색	검색을 통한 각 게시판 정보 보기 제공
	채팅	Spring WebSocket을 통한 상담 기능 제공
	커스텀	jQuery(Drag & Drop)을 통한 커스텀 제작 기능 제공
	QR코드	주문 완료 시 QR코드 생성 기능 제공
	장바구니	Cookie 를 통한 장바구니 기능 제공
DB 설계		
Git 주소	https://github.com/jaewon0913/Final_Project	

채팅 기능

기능 설명

- Spring WebSocket 을 이용한 채팅 구현
- 관리자(상담사)는 각 클라이언트(고객)에게 모든 메시지를 받을 수 있다.
- 관리자는 각 고객에게 웹소켓이 접속 될 때 제공되는 세션의 아이디를 통해 메시지 전달 (서버에서 세션으로 구분하여 알맞은 세션 에게만 메시지가 전달)

코드

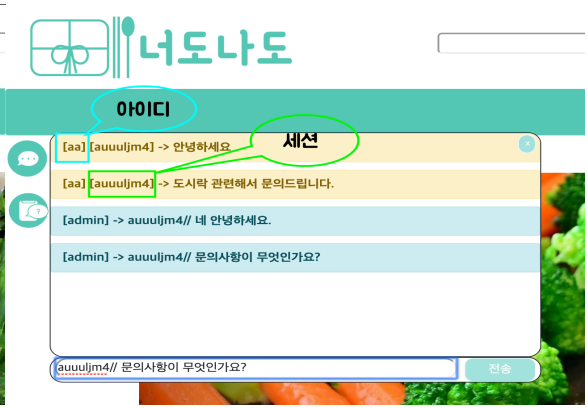
```java
@Override
protected void handleTextMessage(WebSocketSession session, TextMessage message) throws Exception{

    logger.info("{}로 부터 {} 메시지 받음", session.getId(), message.getPayload());

    for(WebSocketSession sess : sessionList) {
        if(session.getPrincipal() == null ) {
            //  로그인이 안된 상대가 보냈을 경우
            sess.sendMessage(new TextMessage(session.getId()+"|" + session.getId() +"|" + message.getPayload()));
        } else {
            //  로그인이 된 상대가 보냈을 경우
            if(session.getPrincipal().getName().equals("admin")) {
                //  만약 보낸 사람이 admin인 경우
                String[] splitMsg = message.getPayload().split("//");
                if(splitMsg[0].equals(sess.getId())) {
                    sess.sendMessage(new TextMessage(session.getPrincipal().getName() + "|admin|" + splitMsg[1]));
                } else if(sess.getPrincipal().getName().equals("admin")){
                    System.out.println("admin이 보낸게 admin에게로");
                    sess.sendMessage(new TextMessage(session.getPrincipal().getName() + "|admin|" + message.getPayload()));
                }
            } else {
                //  만약 보낸 사람이 admin이 아닌 경우
                sess.sendMessage(new TextMessage(session.getPrincipal().getName() + "|" + session.getId() + "|" + message.getPayload()));
            }
        }
    }
}
```

세션

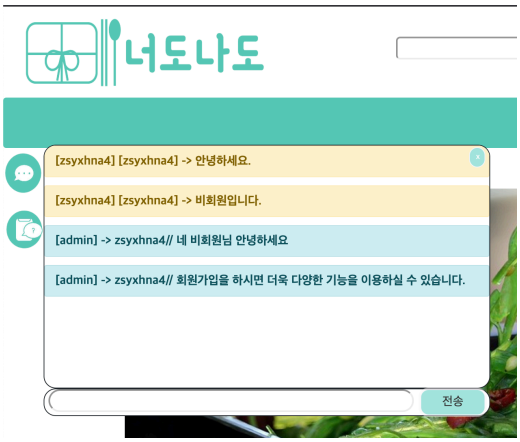
아이디

화면

회원 경우



비회원 경우



장바구니

- **기능 설명**
  - Ajax를 통한 view에서 '장바구니 추가' 버튼을 누를경우 해당 데이터들을 Controller에 전달
  - Cookie 를 통한 데이터를 임시로 저장하여 사용
  - Cookie 가 존재하지 않을 경우 예외처리를 하여 '장바구니가 비어있습니다' 표시
  - 장바구니 기능을 통해서도 결제 가능

- **코드**

```java
@RequestMapping("/createCookie.do")
@ResponseBody
public Map<String, String> cookie(HttpServletResponse response, CookieDto dto) {
    Cookie setCookie_name = new Cookie("cookieName", "커스텀도시락");

    Cookie setCookie_tan = new Cookie("cookieTan", dto.getTan());
    Cookie setCookie_dan = new Cookie("cookieDan", dto.getDan());
    Cookie setCookie_zi = new Cookie("cookieZi", dto.getZi());
    Cookie setCookie_cal = new Cookie("cookieCal", dto.getCal());
    Cookie setCookie_price = new Cookie("cookiePrice", dto.getPrice());
    Cookie setCookie_count = new Cookie("cookieCount", dto.getCount());

    setCookie_name.setMaxAge(60*3);
    setCookie_tan.setMaxAge(60*3);
    setCookie_dan.setMaxAge(60*3);
    setCookie_zi.setMaxAge(60*3);
    setCookie_cal.setMaxAge(60*3);
    setCookie_price.setMaxAge(60*3);
    setCookie_count.setMaxAge(60*3);

    response.addCookie(setCookie_name);
    response.addCookie(setCookie_tan);
    response.addCookie(setCookie_dan);
    response.addCookie(setCookie_zi);
    response.addCookie(setCookie_cal);
    response.addCookie(setCookie_price);
    response.addCookie(setCookie_count);
```

쿠키 시간 설정

쿠키 생성

쿠키 추가

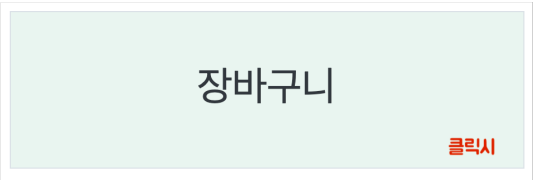
```java
pping("/getCookie.do")
ing getCookie(Model model, HttpSession session, HttpServletRequest request, HttpServletResponse response,
okieDto dto) {
값 가져오기

okie[] cookies = request.getCookies();
ring count = null;
ring[] cName = new String[cookies.length];
ring[] cValue = new String[cookies.length];

(cookies != null) {
  for (int i = 0; i < cookies.length; i++) {
      Cookie c = cookies[i];
      // 저장된 쿠키 이름을 가져온다
      cName[i] = c.getName();
      // 쿠키값을 가져온다
      cValue[i] = c.getValue();
      if (c.getName().equals("cookieCount")) {
          count = c.getValue();
      } else if ("memberCartBasket",
  } catch (Exception e) {
          response.setCharacterEncoding("UTF-8");
          response.setContentType("text/html; charset=UTF-8");

          PrintWriter out;

          try {
              out = response.getWriter();
              out.println("<script>"
                      + "alert('장바구니가 비어있습니다.');"
                      + "location.href = 'mypage.do';</script>");

              out.flush();
          } catch (IOException e1) {
              e1.printStackTrace();
          }
          e.printStackTrace();
          return "error/ErrorPage";
      }
  }
```

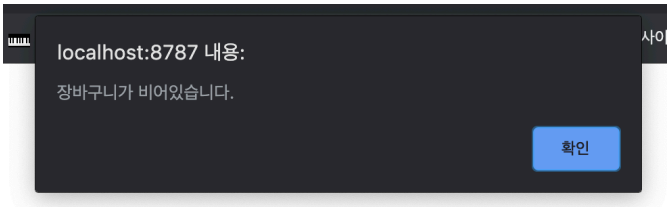
쿠키의 데이터를 가져와 DTO에 다시 저장하여 modelAttribute로 사용

쿠키가 비어있을 경우 예외를 발생시켜 다시 마이페이지로 이동

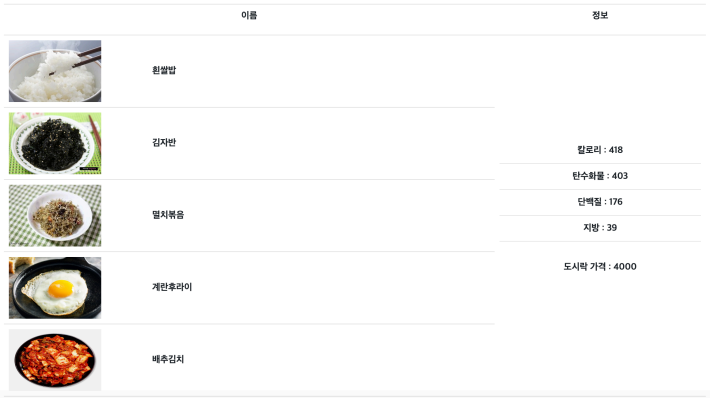
- **화면**

쿠키가 비어있을 경우





쿠키가 존재 할 경우



파일 업로드

- **기능 설명**
  - 도시락에 들어가는 반찬의 사진을 등록하기 위한 파일 업로드 기능
  - 파일을 추가하여 프로젝트 폴더에 파일을 추가하는 것이 아닌 서버(tomcat)에 파일을 추가
    (경로 : .metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps/Final_Project/resources/etc/upload/파일)

- **코드**

```java
try {
    //  request.getSession().getServletContext()에서 /storage 폴더 까지 경로 저장
    inputStream = file.getInputStream();

    String path = WebUtils.getRealPath(request.getSession().getServletContext(), "/resources/etc/upload");

    System.out.println("업로드 될 실제 경로 : " + path);
    System.out.println("servletcontext : " + servletContext.getRealPath("resources/etc/upload"));
    fileobj.setFile_path(path + "/" + filename);

    //  위의 path의 경로를 가진 파일 저장소 생성
    //File storage = new File(testPath);
    File storage = new File(path);

    //  만약 저장소가 존재 하지 않으면 저장소를 만든다.
    if(!storage.exists()) {
        storage.mkdirs();
    }
```

절대 경로로 설정

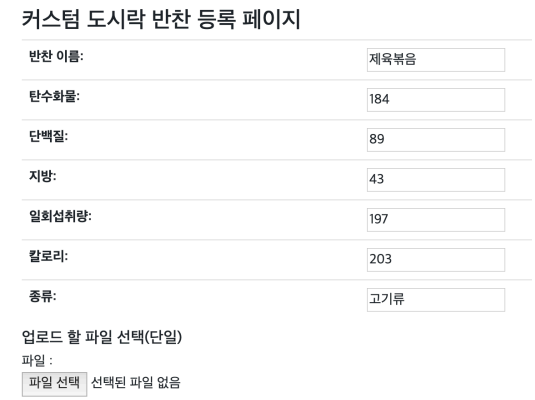
서버의 위치 .metadata/.plugins/org.~.server.core/tmp0/wtpwebapps/

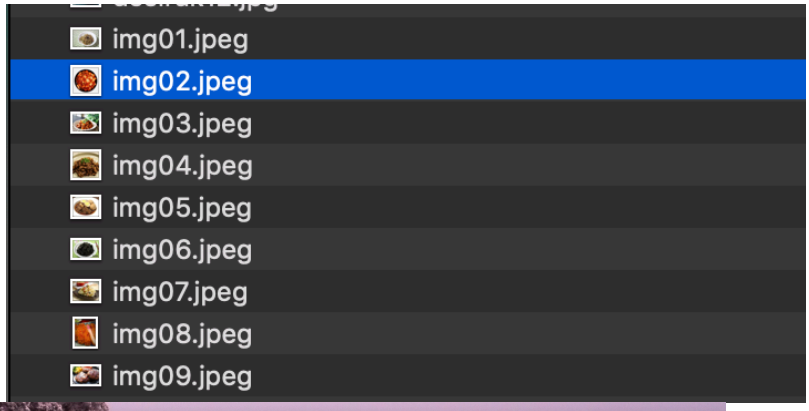
추가로 원하는 위치

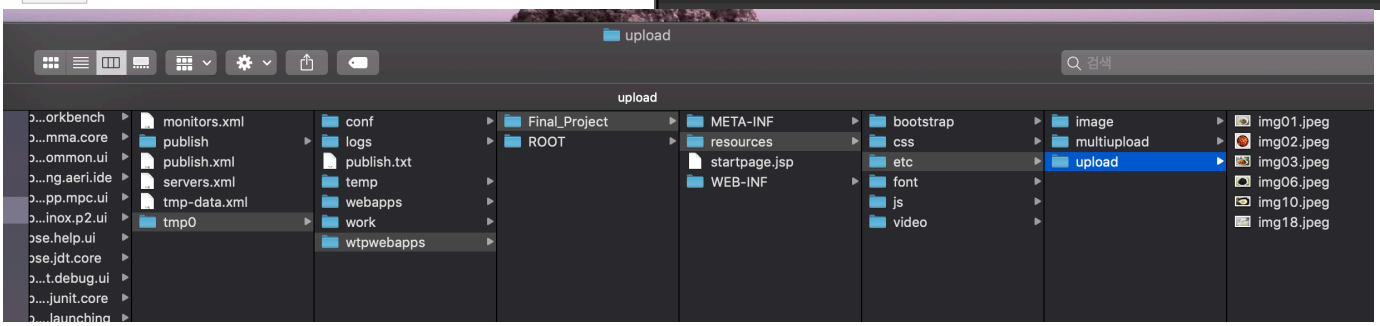
```java
    //  위의 path의 경로에 새로운 파일 생성
    //File newfile = new File(testPath + "/" + filename);
    File newfile = new File(path +"/"+ filename);

    //  경로에 파일이 없으면 파일을 만든다.
    if(!newfile.exists()) {
        newfile.createNewFile();
    }

    //  파일 출력
    outputStream = new FileOutputStream(newfile);

    int read = 0;
    //  파일의 크기를 byte 형식으로 저장
    byte[] b = new byte[(int)file.getSize()];

    //  byte가 다음이 없으면 -1 / 있으면 0 / 다음이 있으면 1 (즉, 파일이 끝나면)
    while((read=inputStream.read(b)) != -1) {
        //  그 크기만큼 읽어서 출력
        outputStream.write(b, 0, read);
    }
```

- **화면**

✖ '파일 선택' 버튼 누른 후 -> 이미지 선택 -> 완료되면 아래의 위치에 이미지 추가 완료







Drag & Drop

- **기능 설명**
  - jQuery에서 제공한 Drag & Drop 기능 사용
  - 반찬 이미지 클릭 시(Drag) 이미지 정보가 복사되어 빈 칸에 Drop 했을 때 데이터 복사되어 추가
  - 반찬 이미지 클릭 시(Drag) 자동으로 화면 중간으로 이동됨

- **코드**

```
// 드래그 시작시
function dragstart_handler(event){
    fnMove();

    event.dataTransfer.effectAllowed = "copy";
    event.dataTransfer.setData("text/plain",event.target.id);
}

// 드래그 시작시 화면 중앙에 위치
function fnMove(){
    var offset = jb(".container").offset();
    jb('html,body').animate({scrollTop : offset.top + 100}, 400);
}

// 드래그 도중
function dragover_handler(event){
    event.dataTransfer.dropEffect = "copy";
    event.preventDefault();
}
```

화면 이동

데이터를 복사
복사 하는 것 : 해당 데이터의 ID 값

```
// 드래그 끝 = 드랍시
function drop_handler(event){
    event.dataTransfer.dropEffect = "copy";
    event.preventDefault();

    var data = event.dataTransfer.getData("text/plain");

    var copyimg = document.createElement("img");
    var original = document.getElementById(data);

    copyimg.src = original.src;
    copyimg.id = original.id + "copy";

    console.log(original.name);
    console.log(event.target.childNodes[1]);

    event.target.childNodes[1].value = original.name;

    console.log(original.src);
    console.log(copyimg.src);

    jb(copyimg).addClass('copydish');

    // 데이터 추가하기
    one_data(original.id, original.name, event.target.childNodes[1]);//  1회 구매창
    multi_data(original.id, original.name, event.target.childNodes[1]);  //  정기 구매창

    event.target.appendChild(copyimg);
    copyimg.setAttribute("style","width : 15rem; height : 15rem;");
    copyimg.setAttribute('draggable', true);

    jb(copyimg).dblclick(function() {
        jb(this).remove();
        one_remove(this.id);
        multi_remove(this.id);
    });
}
```

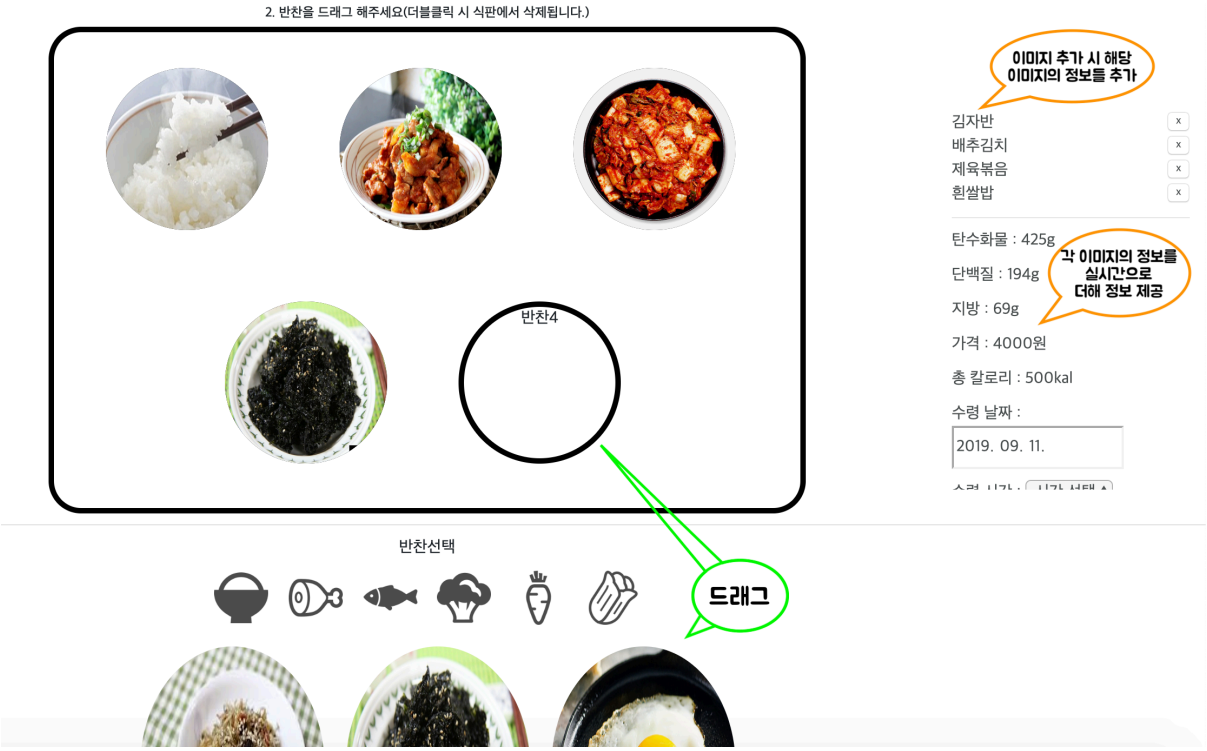
'img' 태그 생성

생성한 'img' 태그에 복사한 데이터의 src 복사

생성한 'img' 태그에 copydish 클래스 추가

복사된 이미지 더블 클릭시 데이터 삭제

- **화면**



매크로 봇

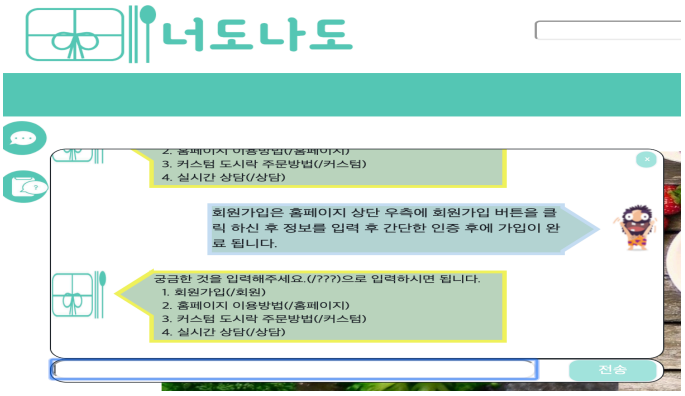
- **기능 설명**
  - 간단한 설명 등을 제공하는 기능
  - 알맞은 키워드 입력시 해당 키워드에 맞는 문장 출력

- **코드**

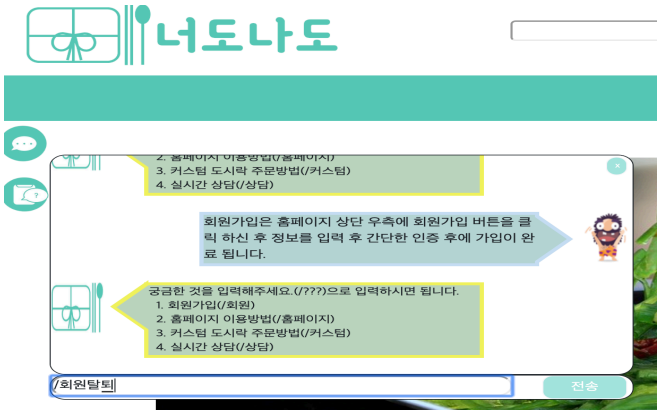
```
function botSendMessage(){
    var message = $("#bot_message").val();
    if(message == "/회원"){
        $("#bot_msg_div").append(
                "<img alt = '손님' src = 'resources/bootstrap/image/bot2.png' style = 'width : 5rem; height: 7rem; float : right;'>" +
                "<div class = 'arrow_box_right' style = 'width : 32rem; right : -14rem; float : left;'>" +
                "회원가입은 홈페이지 상단 우측에 회원가입 버튼을 클릭 하신 후 정보를 입력 후 간단한 인증 후에 가입이 완료 됩니다." +
                "</div>" +
                "<br style = 'clear : both' /><br>" +
                "<img alt='너도나도' src='resources/bootstrap/image/nado2.png' style='width: 5rem; height: 7rem; float : left;'>" +
                "<div class = 'arrow_box' style = 'width : 32rem; left : -14rem; float : right;'>궁금한 것을 입력해주세요.(/???)으로 입력하시면 됩니다.<br/>" +
                "   1. 회원가입(/회원)<br>" +
                "   2. 홈페이지 이용방법(/홈페이지)<br>" +
                "   3. 커스텀 도시락 주문방법(/커스텀)<br>" +
                "   4. 실시간 상담(/상담)<br></div>" +
                " <br style = 'clear : both' /><br>");
    } else if(message == "/홈페이지"){
        $("#bot_msg_div").append(
                "<img alt = '손님' src = 'resources/bootstrap/image/bot2.png' style = 'width : 5rem; height: 7rem; float : right;'>" +
                "<div class = 'arrow_box_right' style = 'width : 32rem; right : -14rem; float : left;'>" +
                "홈페이지 크게 도시락 주문 / 영양 게시판 / 게시판 / 배달장소 확인 으로 이루어져 있습니다.<br>" +
                "1. 도시락 주문은 원하는 도시락을 클릭 또는 커스텀하여 주문을 할 수 있습니다.<br>" +
                "2. 영양 게시판은 각 도시락에 대한 영양 정보나 자신이 주문했던 도시락의 영양 정보를 보여줍니다.<br>" +
                "3. 게시판은 자유게시판, 이벤트게시판, 공지사항으로 자유롭게 글 작성, 수정 등이 가능합니다." +
                "4. 배달장소 확인은 각자 자신이 받을 지역의 역을 보여줍니다." +
                "</div>" +
                "<br style = 'clear : both' /><br>" +
                "<img alt='너도나도' src='resources/bootstrap/image/nado2.png' style='width: 5rem; height: 7rem; float : left;'>" +
                "<div class = 'arrow_box' style = 'width : 32rem; left : -14rem; float : right;'>궁금한 것을 입력해주세요.(/???)으로 입력하시면 됩니다.<br/>" +
                "   1. 회원가입(/회원)<br>" +
                "   2. 홈페이지 이용방법(/홈페이지)<br>" +
```

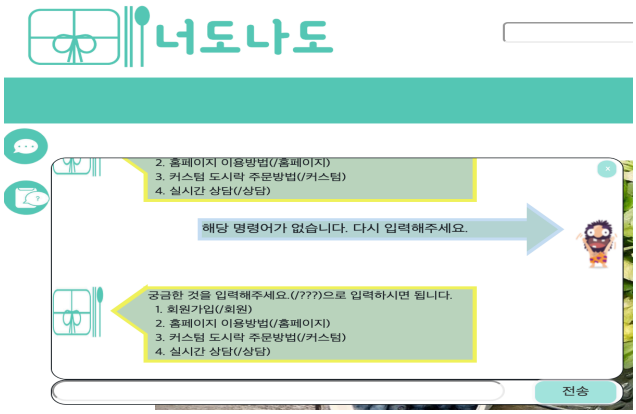
- **화면**

키워드 있을 때

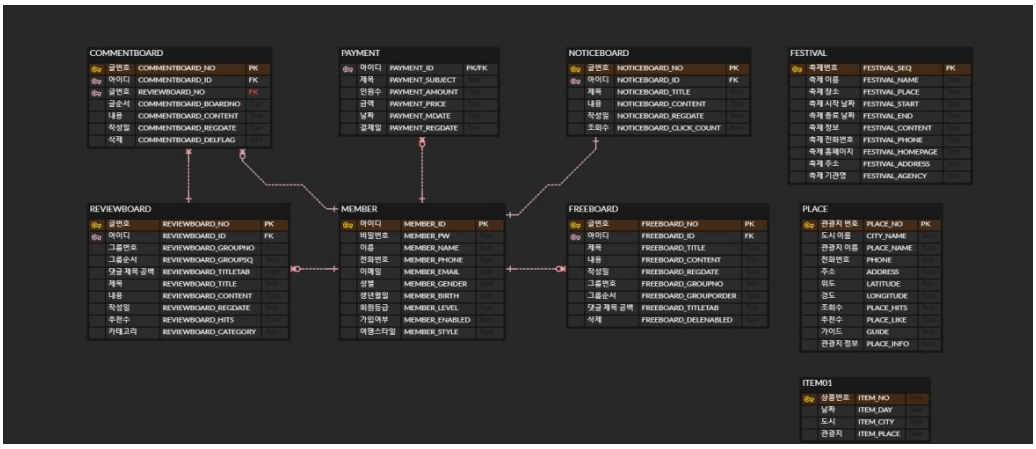


키워드가 없을 때





세미 프로젝트

항목	주요 내용	
프로젝트 명	외국인을 위한 국내 여행 정보 사이트	
수행 기간 및 인원	2019. 05. 27 ~ 2019. 06. 21 (총 5 명)	
개발 목표	외국인들이 우리나라를 여행을 하게 되었을 때 우리나라 전통미를 강조하며 다른 사이트와 다른 편리함과 다양한 정보를 제공 우리나라의 축제 및 관광지 등의 정보를 쉽게 수집 할 수 있도록 할 것이다.	
개발 환경	OS	Window 7 , 10 / Mac OS
	구현 언어	Java, HTML5, CSS3, Java Script, Oracle
	사용 기술	Ajax, Servlet, JSP, Google Map API, FaceBook Login API, Kakao (Pay / Map) API , 공공데이터포털 데이터 API, 기상청(날씨누리) API
	사용 도구	Eclipse , Oracle DataBase 11g Express Edition, Apache Tomcat 9.0 ,Summer Note
담당 역할	전체 프로젝트 총괄 관리(팀장), 웹소켓 채팅, 날씨 제공, 게시판(CRUD, 페이징), 기타 기능, 프로젝트 형상관리(Git)	
구현 기능	로그인	Facebook Login API를 사용한 로그인 기능 과 기본 로그인 기능 제공
	결제	Kakao Pay API를 통한 결제 시스템 제공
	지도	Google Map API, Kakao Map API를 통한 장소 정보 제공 대중 교통 이용 시 걸리는 시간 제공(길찾기)
	게시판	CRUD , 페이징, 검색 기능
	채팅	WebSocket을 통한 상담 기능 제공
	날씨	기상청 데이터 크롤링을 통한 날씨 정보 제공
	투어	Kakao Map API를 통한 경로 소개
	축제, 관광지	각 월 별 축제 정보 제공 각 도시 별 관광지 정보 제공
DB 설계		
Git 주소	https://github.com/jaewon0913/Semi_Project_git	

채팅 기능

- **기능 설명**
  - WebSocket 을 이용한 채팅 구현
  - 관리자(상담사)는 각 클라이언트(고객)에게 모든 메시지를 받을 수 있다.
  - 관리자는 각 고객에게 웹소켓이 접속 될 때 제공되는 세션의 아이디를 통해 메시지 전달 (서버에서 세션으로 구분하여 알맞은 세션 에게만 메시지가 전달)

- **코드**

```java
@ServerEndpoint("/broadcasting")
public class websocket {
    private static Set<Session> clients = Collections.synchronizedSet(new HashSet<Session>());

    @OnMessage
    public void onMessage(String message, Session session) throws IOException {
        System.out.println(message);
        synchronized(clients) {
            for(Session client : clients) {
                if(!client.equals(session)) {
                    client.getBasicRemote().sendText(message);
                }
            }
        }
    }

    @OnOpen
    public void onOpen(Session session) {
        System.out.println(session);
        clients.add(session);
    }

    @OnClose
    public void onClose(Session session) {
        clients.remove(session);
    }
}
```

주소 끝이 'broadcating' 인 곳을 찾는다

메시지 전송 (접속 한 세션을 다 찾아 전송)

웹소켓 실행시

웹소켓 종료시

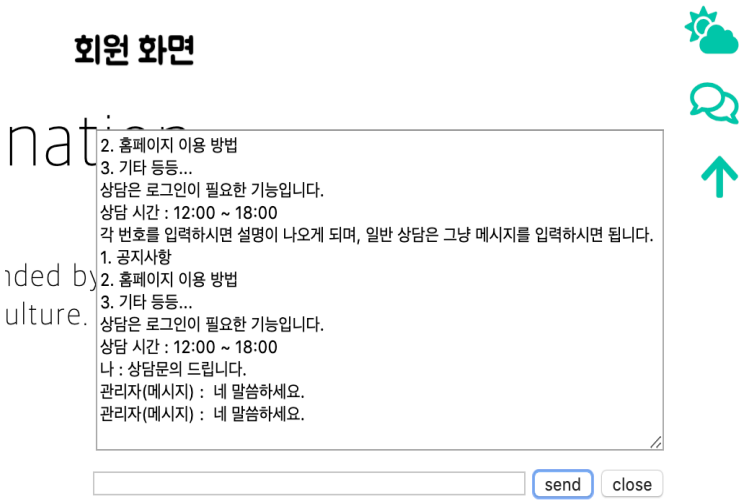
```java
var textarea = document.getElementById("messageWindow");
var webSocket = new WebSocket('ws://localhost:8787/Semi_Project_final/broadcasting');
var inputMessage = document.getElementById('inputMessage');
webSocket.onerror = function(event) {
    onError(event)
};
webSocket.onopen = function(event) {
    onOpen(event)
};
webSocket.onmessage = function(event) {
    onMessage(event)
};
function onMessage(event) {
    var message = event.data.split("|");
    var sender = message[0];
    var content = message[1];
    if(content == ""){
    //메시지가 비어 있으면 보내지 않는다.
    } else {
        if(sender == "admin"){
            if(content.match("/")){
                if(content.match("/" + $("#chat_id").val())){
                    var temp = content.replace("/" + $("#chat_id").val(), "(메시지) : ").split(": ");
                    if(temp[1].trim == ""){
                    } else {
                        textarea.value += "관리자" + content.replace("/" + $("#chat_id").val(), "(메시지)
                    }
                } else {
                }
            }
        } else {
            if(content.match("/")){
                if(content.match("/" + $("#chat_id").val())) {
                    var temp = content.replace("/" + $("#chat_id").val(), "(메시지) : ").split(": ");
                    if(temp[1].trim == ""){
                    } else {
                        textarea.value += sender + content.replace("/" + $("#chat_id").val(), "(메시지)
                    }
                } else {
                }
```

이 주소로 웹소켓을 열겠다

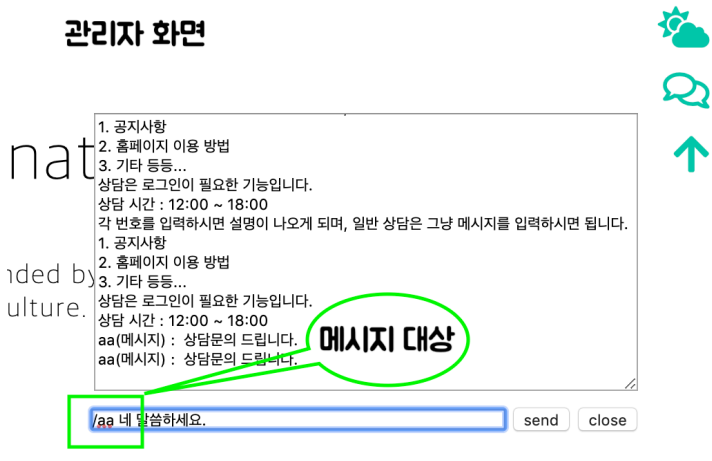
메시지 전송 (@OnMessage 로 전송)

- **화면**

회원 화면



관리자 화면



날씨 제공

- **기능 설명**
  - 기상청(날씨누리) 데이터를 크롤링하여 데이터를 jSon으로 파싱하여 날씨 정보 제공
  - 해당 지역 날씨를 클릭 시 해당 지역 미세먼지 정보(표) 제공

- **코드**

```java
$(function() {
    $(".weatherbutton").click(function() {
        $(".k_map").css("display", "block");
        $("#chatbox").css("display", "none");

        var url = "etc.do?command=weather";

        $.ajax({
            type : "GET",
            url : url,
            dataType : "text",
            success : function(data){
                var tmp = $.trim(data);
                var obj = JSON.parse(tmp);

                $.each(obj, function(key,val){
                    if(key == "DATA"){
                        var list = val;
                        for(var i = 0 ; i<list.length; i++){
                            $("#weather"+i).empty();
                        }
                        for(var i = 0 ; i<list.length; i++){
                            var str = list[i];
                            $("#weather"+i).append(str.wf + str.tmx + "/" + str.tmn);
                        }
                    }
                });
            },
public String[] getweatherdata() {
    String url = "http://www.weather.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108";
    Document doc = null;

    try {
        doc=Jsoup.connect(url).get();
    } catch (IOException e) {
        e.printStackTrace();
    }

    Elements elements_test = doc.getElementsByTag("location");

    weatherdata = new String[elements_test.size()];

    for(int i = 0 ; i <elements_test.size() ; i ++) {
        weatherdata[i] = elements_test.get(i).getElementsByTag("city").text() + "/"
        + elements_test.get(i).getElementsByTag("data").get(0).getElementsByTag("wf").text() + "/"
        + elements_test.get(i).getElementsByTag("data").get(0).getElementsByTag("tmn").text() + "/"
        + elements_test.get(i).getElementsByTag("data").get(0).getElementsByTag("tmx").text();
    }

    String[] choice_data = new String[12];
    choice_data[0] = weatherdata[0];
    choice_data[1] = weatherdata[1];
    choice_data[2] = weatherdata[6];
    choice_data[3] = weatherdata[8];
    choice_data[4] = weatherdata[12];
```

컨트롤러에서 온 jSon 데이터 파싱

기상청에서 크롤링

원하는 태그의 데이터만 골라서 저장

```java
if(command.equals("weather")) {
    String[] weather_data = new weather().getweatherdata();

    JSONObject object = new JSONObject(); // {}
    JSONArray array = new JSONArray(); // []
    JSONObject total = new JSONObject(); // {}

    String[] split_data = null;

    for (int i = 0; i < weather_data.length; i++) {
        split_data = weather_data[i].split("/");
        object.put("city", split_data[0]);
        object.put("wf", split_data[1]);
        object.put("tmn", split_data[2]);
        object.put("tmx", split_data[3]);

        array.add(object);
    }
    total.put("DATA", array);

    out.print(total);
```

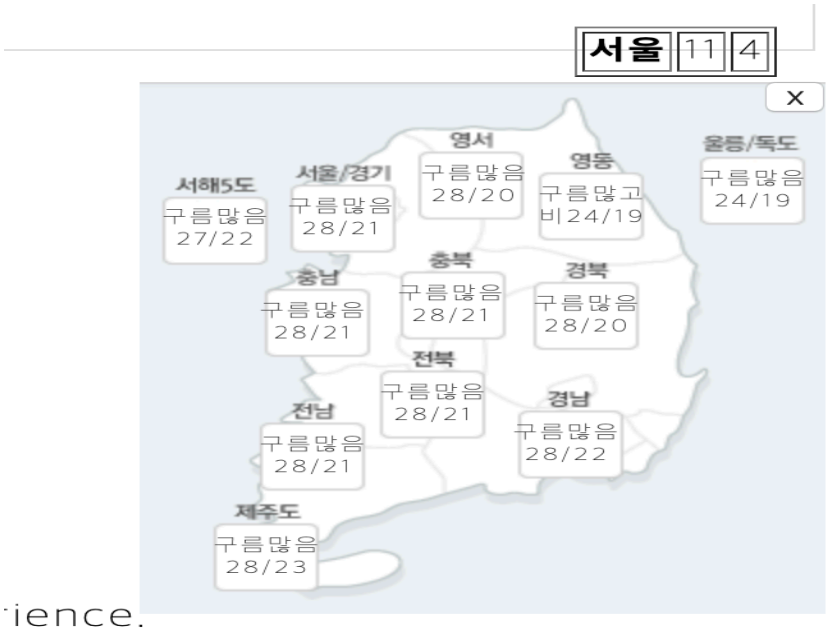
데이터 호출

jSon 선언

호출 한 데이터를 Split 하여 원하는 데이터만 저장

[DATA : {[city: ~~] , [wf : ~~], [tmn:~~] , [tmx:~~]}, {[]} ...] 형식으로 저장

- **화면**



게시판

- 기능 설명
  - 기본 CRUD 제공
  - 글 작성(create) 시 텍스트에디터(summer note)를 사용
  - 페이징 기능 구현
- 코드

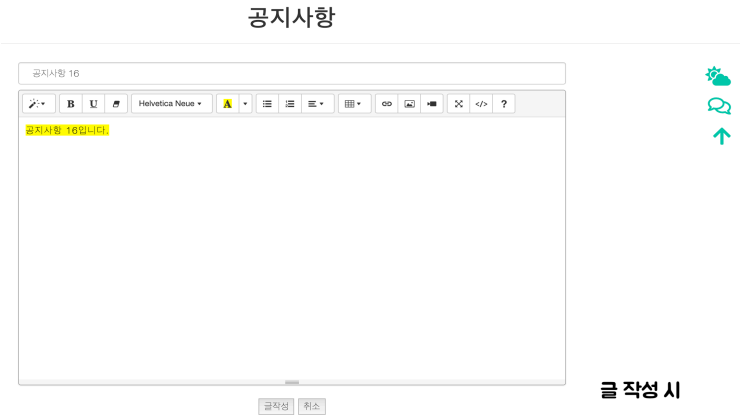
페이징

```java
public class Paging {
    private int page =1; //현재 페이지 (get)
    private int totalCount; //row 전체의 수 (get)
    private int beginPage;  //출력 시작
    private int endPage;    //출력 끝
    private int displayRow =10;  //한 페이지에 몇 개의 로우 (선택 set)
    private int displayPage =10;  //한 페이지에 몇 개의 페이지 (선택 set)
    boolean prev; //prev 버튼이 보일건지 안보일건지
    boolean next; //next 버튼이 보일건지 안보일건지
```

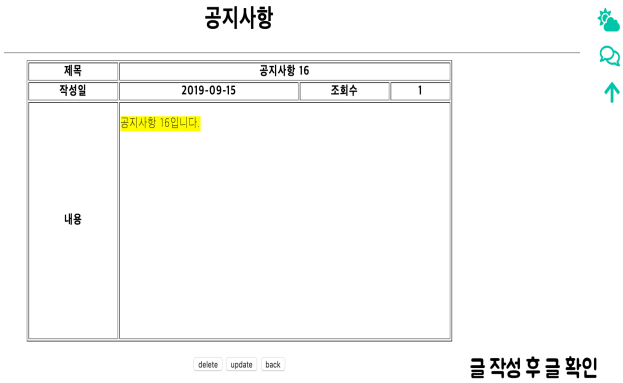
```java
private void paging(){
    // prev, next, beginPage, endPage를 계산해서 만든다.
    // 2+9 = 11, 11/10 = 1, 1*10 = 10
    // 10+9 = 19, 19/10 = 1, 1*10 = 10
    // 111+9 = 120 120/10 = 12, 12*10 = 120

    // (2+9)/10 * 10 (1번 방법)
    //endPage = ((page+(displayPage-1))/displayPage)*displayPage;

    // 1/10 0.1(올림) 1 (2번 방법)
    endPage = ((int)Math.ceil(page/(double)displayPage))*displayPage;
    System.out.println("endPage : " + endPage);

    beginPage = endPage - (displayPage - 1);
    System.out.println("beginPage : " + beginPage);

    // 글 32개
    // 32/10 = 3.2 (올림) 4페이지
    // 2=?  2/10
    int totalPage = (int)Math.ceil(totalCount/(double)displayRow);

    if(totalPage<endPage){
        endPage = totalPage;
        next = false;
    }else{
        next = true;
    }
    prev = (beginPage==1)?false:true;//page가 11이상에만 나온다.
```

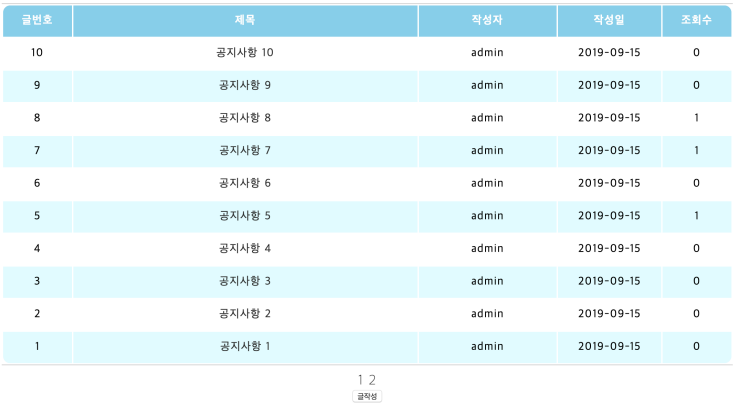
- 화면

Create(글 작성 , summer Note)



글 작성 시



글 작성 후 글 확인

Read (페이징)