

새로운 것을 두려워하지 않고 부딪쳐보는 도전정신, 이여울입니다

2020 PORTFOLIO

CONTACT

sleepy2head@naver.com

010 9120 4349



트렌디한 개발자를 꿈꾸는 이여울입니다.

새로운 것을 두려워하지 않고
부딪쳐보는 도전정신,
맡은 일에 최선과 최고를 목표로 살아가고 있는 이여울입니다.

이여울 / yeowool Lee

1992.09.24 / 서울특별시 동대문구

Tel. 010-9120-4349

Email. sleepy2head@naver.com

GitHub. <https://github.com/Leeyeowool>

GRADUATION

2011 평택여자고등학교 졸업

2014 서울모드패션전문학교 패션디자인과 졸업

2020 평생교육진흥원 패션디자인 학사취득예정

CAREER

2014 ~ 2019

HYPER UTOPIA 패션브랜드 디자인, 운영

2020

KH정보교육원 웹개발자 과정 수료

SKILL

Java / JavaScript / HTML

jQuery / CSS / ORACLE SQL

Servlet / JSP / Spring Framework

Mybatis / Ajax / Linux

LICENSE

운전면허 2종 보통

PROJECT.

1M [일맷다]

실시간 인력 공유 서비스

일하고 싶은 만큼만, 일하고 싶은 사람과 일맷다

1M앞 내 방 부터 내가 원하는 위치에서
단순노동부터 맞춤형 서비스까지
라이프스타일에 맞추는 자유로운 스케줄링

1m



실시간 인력 공유 서비스 [일맷다]

기존의 구인구직 사이트도 변해가는 시대에 맞춰 우리는 새로운 플랫폼의 필요성을 느끼게 되었습니다. 기존의 사이트들의 불편한 점은 어떤 것이 있을까요?

보다 정식적이고 구인과 구직 한쪽에만 초점이 맞춰져 있어서 나의 인력이 필요하거나 내가 인력이 필요할 때 쉽게 공고를 올리지 못한다는 점이 있습니다.

또 소소한 일거리를 원하는 직장인들은 아르바이트를 찾기 어렵습니다. 다양한 라이프스타일에 맞춘 자유로운 관리와 기존 고용서비스의 단점을 보완한 **일맷다**는

구인구직 사이트의 기능과 컨시어지 서비스 그리고 SNS 기능을 갖춘

공유 인력 사이트입니다. 내가 일할 수 있는 시간에 원하는 위치에서 인력을

제공하고 제공 받을 수 있도록 다양한 서비스를 이용 할 수 있습니다.



PROJECT.

1M [일맷다]

실시간 인력 공유 서비스

개발
도구





개발 도구

SonarLint를 사용하여 정적 분석을 통해 중복코드와 오류, 코드 스멜(Code Smell)을 검사하였습니다.

SonarLint Report

Filter matched 100 of 1683 items

Resource	Date	Description
AdminMemberDao.java		The type of the "map" object should be an interface such as "Map" rather than
AdminMemberDao.java		The type of the "map" object should be an interface such as "Map" rather than
AdminMemberDao.java		The type of the "map" object should be an interface such as "Map" rather than
AdminMemberDao.java		Provide the parametrized type for this generic.
AdminMemberDao.java		Provide the parametrized type for this generic.
AdminMemberDao.java		Provide the parametrized type for this generic.
AdminMemberDao.java		Add a nested comment explaining why this method is empty, throw an Unsup
AdminMemberServiceImpl		Immediately return this expression instead of assigning it to the temporary vari
AdminMemberServiceImpl		Replace this use of System.out or System.err by a logger.
AdminMemberServiceImpl		Add a nested comment explaining why this method is empty, throw an Unsup
AdverController.java		Refactor your code to get this URI from a customizable parameter.
AdverController.java		Remove this unused "category" local variable.
AdverController.java		Remove this unused "category" local variable.
AdverController.java		Remove this unused "company" local variable.
AdverController.java		Remove this unused import 'chatMatching.spring.MatchingService'.
AdverController.java		Remove this unused import 'javax.servlet.http.HttpServletRequest'.
AdverController.java		Use a logger to log this exception.
AdverController.java		Remove this useless assignment to local variable "category".
AdverController.java		Remove this useless assignment to local variable "category".
AdverController.java		Remove this useless assignment to local variable "company".
AdverController.java		Remove this useless assignment to local variable "json1".
AdverController.java		Replace this use of System.out or System.err by a logger.
AdverController.java		Replace this use of System.out or System.err by a logger.
AdverController.java		Replace this use of System.out or System.err by a logger.
AdverController.java		Replace this use of System.out or System.err by a logger.
AdverController.java		This block of commented-out lines of code should be removed.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Add a "method" parameter to this "@RequestMapping" annotation.
AdverController.java		Define a constant instead of duplicating this literal "userCode" 4 times.
AdverDao.java		Remove this unused import 'chatMatching.vo.ReviewVo'.
AdverDao.java		Remove this unused import 'myBlogErEe.vo.CareerVo'.
AdverDao.java		Remove this unused import 'myBlogErEe.vo.IntroductionVo'.

SonarLint Rule Description

Unnecessary imports should be removed (java:S1128)

Code smell Minor

The imports part of a file should be handled by the Integrated Development Environment (IDE), not manually by the developer.

Unused and useless imports should not occur if that is the case.

Leaving them in reduces the code's readability, since their presence can be confusing.

Noncompliant Code Example

```
package my.company;

import java.lang.String;           // Noncompliant; java.lang classes are always implicitly in
import my.company.SomeClass;      // Noncompliant; same-package files are always implicitly i
import java.io.File;              // Noncompliant; File is not used

import my.company2.SomeType;
import my.company2.SomeType;      // Noncompliant; 'SomeType' is already imported

class ExampleClass {

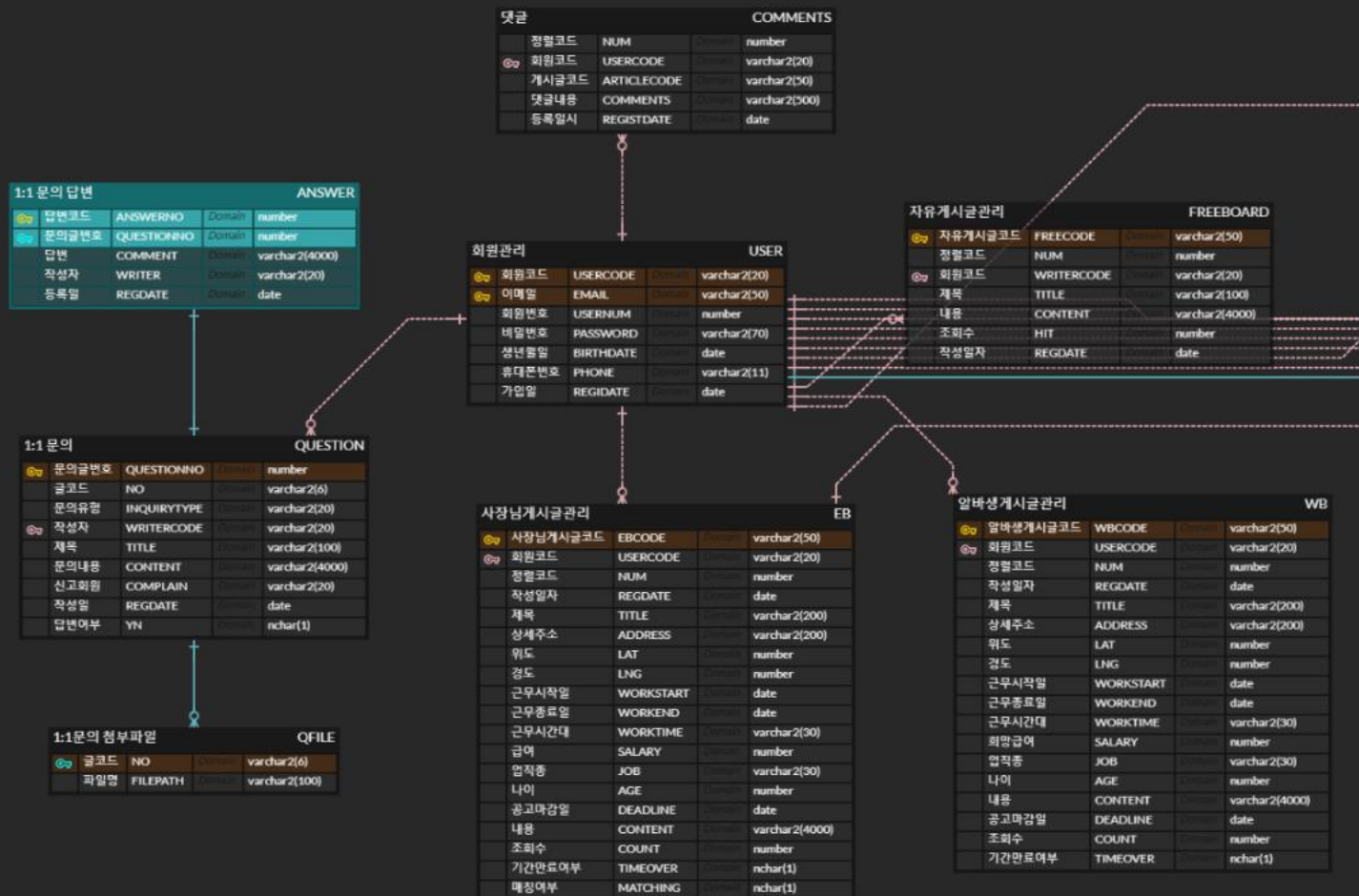
    public String someString;
    public SomeType something;

}
```

Exceptions

Imports for types mentioned in comments, such as Javadocs, are ignored.

일맷다 담당 데이터 베이스 관리



일맷다에서 어떤 기능을 구현했나요?

스프링 MVC2 패턴을 활용하여 REST 방식으로 구현했습니다.
지도검색기능과 에디터api를 이용한 게시판,
고객센터를 구현하였고 홈페이지 전반적인 CSS를 담당했습니다.

프로젝트 기여도

팀원 6명 기여도 20

참여	<div></div>	100
구현	<div></div>	100

프로젝트 기간 20.02.24 - 20.04.03

지도검색

지도api 활용 / 검색기능
ajax를 이용한 동적 리스트 생성



자유게시판

REST-ful 방식
CRUD / 페이징 / 댓글 / 검색

고객센터

공지사항 / 이용약관 / 1:1문의



CSS

메인 css 담당



메인

CSS와 실시간 인기글 구현

데이터베이스에서 10분마다 실시간으로 조회수 상위 5개씩
새로고쳐지면서 동적으로 리스트가 됩니다.

```
// 실시간 인기글 , 광고
function startAjax(){
    setInterval(function(){
        updateAtc()
        updateAd()
    }, 600000);
};
```



일프로게시판 일나눔게시판 자유게시판 지도검색 고객센터 마이페이지

김문영님
게시물 0 마이페이지 로그아웃

일하고 싶은 만큼만,
일하고 싶은 사람과 일했다

1M달 내 방 부터 원하는 위치에서,
단순노동부터 맞춤형 서비스까지
라이프스타일에 맞추어 자유로운 스케줄링

지금 일맺기



일프로 인기광고

일나눔 인기광고

일프로 광고광고

신청하기

일나눔 광고광고

신청하기

일프로 인기광고

일나눔 인기광고

일프로 광고광고

신청하기

일나눔 광고광고

신청하기

- 1 인터프리티
코딩 맡겨주세요
서대문구 | 급여 : 70000
- 2 인터프리티
일 너무너무 하고싶습니다
강남구 | 급여 : 70000
- 3 인터프리티
일 너무너무 하고싶습니다
강남구 | 급여 : 70000
- 4 인터프리티
중국어 영어 번역합니다
서초구 | 급여 : 60000
- 5 인터프리티
강아지 산책 시켜드립니다
강서구 | 급여 : 50000

- 1 인터프리티
카페 6시간 인력 구함
서대문구 | 급여 : 40000
- 2 인터프리티
장 대신 봐주실 분 구합니다!
동대문구 | 급여 : 5000
- 3 인터프리티
장 대신 봐주실 분 구합니다!
동대문구 | 급여 : 5000
- 4 인터프리티
장 대신 봐주실 분 구합니다!
동대문구 | 급여 : 5000
- 5 인터프리티
장 대신 봐주실 분 구합니다!
동대문구 | 급여 : 5000

- 1 인터프리티
베이비 시터 구하시는 분 찾아요
하남시 | 급여 : 50000
- 2 인터프리티
강아지 산책 시켜드립니다 ...
서초구 | 급여 : 20000
- 3 인터프리티
중국어 영어 번역합니다
서초구 | 급여 : 60000
- 4 인터프리티
강아지 산책 시켜드립니다 ...
서초구 | 급여 : 20000
- 5 인터프리티
중국어 영어 번역합니다
서초구 | 급여 : 60000

- 1 인터프리티
카페 6시간 인력구함
서대문구 | 급여 : 40000
- 2 인터프리티
용두동 후꾸미 사다주세요
중랑구 | 급여 : 40000
- 3 인터프리티
치킨집 서빙 구함
동대문구 | 급여 : 35000
- 4 인터프리티
성수동 고양이 캣타워 운반해..
중랑구 | 급여 : 40000

순위	닉네임	평균평점	조회수	순위	닉네임	평균평점	조회수
1	인터프리티	3.10	★★★★★	4	김지영	3.85	★★★★★
2	김혜진	3.5	★★★★★	5	김지영	5	★★★★★
3	인터프리티	2.75	★★★★★	6	브록	5	★★★★★
4	인터프리티	5	★★★★★	7	김지영	4	★★★★★
5	인터프리티	5	★★★★★	8	인터프리티	4	★★★★★
6	브록	5	★★★★★	9	김지영	4	★★★★★
7	김지영	4	★★★★★	10	김지영	4	★★★★★
8	인터프리티	3.5	★★★★★				
9	김지영	3	★★★★★				
10	김지영	0.5	★				

지도검색

사용자의 ip를 활용하여 사용자의 현재 위치로 지도의 첫 화면으로 띄어줍니다.

일프로/일나눔게시판에 글을 작성하면서 주소를 입력하게 되는데 이때

사용자의 주소를 좌표로 변환하여 지도를 불러내면서 좌표가 마커로 찍히게 됩니

다. 실제로 좌표를 입력하게 되면 사용자 주소가 정확한 위치에 표시되는데 마커를

리스트링을 응용하여 사용자의 개인정보가 그대로 노출되는 것을 차단하고 인근

범위 단위로 근무지를 공개해사이트의 신뢰성과 안정성을 높였습니다.



[VIEW]

```
//사용자 IP 얻어오기
function getIp(){
    $.ajax({
        url: "${pageContext.request.contextPath}/getip",
        dataType: "json",
        async: false,
        success: function(data) {
            clientIp = data.ip;
        },
        error: function(errorThrown) {
            console.log('실패')
        }
    })
};
```

[CONTROLLER]

```
//IP 필요청해서 얻기
@RequestMapping(value="/getip", produces = "text/json; charset=utf-8")
@ResponseBody
public String getIp() throws UnsupportedEncodingException, IOException, ParseException {
    String ipJson = new JsonIpClient().sendGet();
    return ipJson;
}

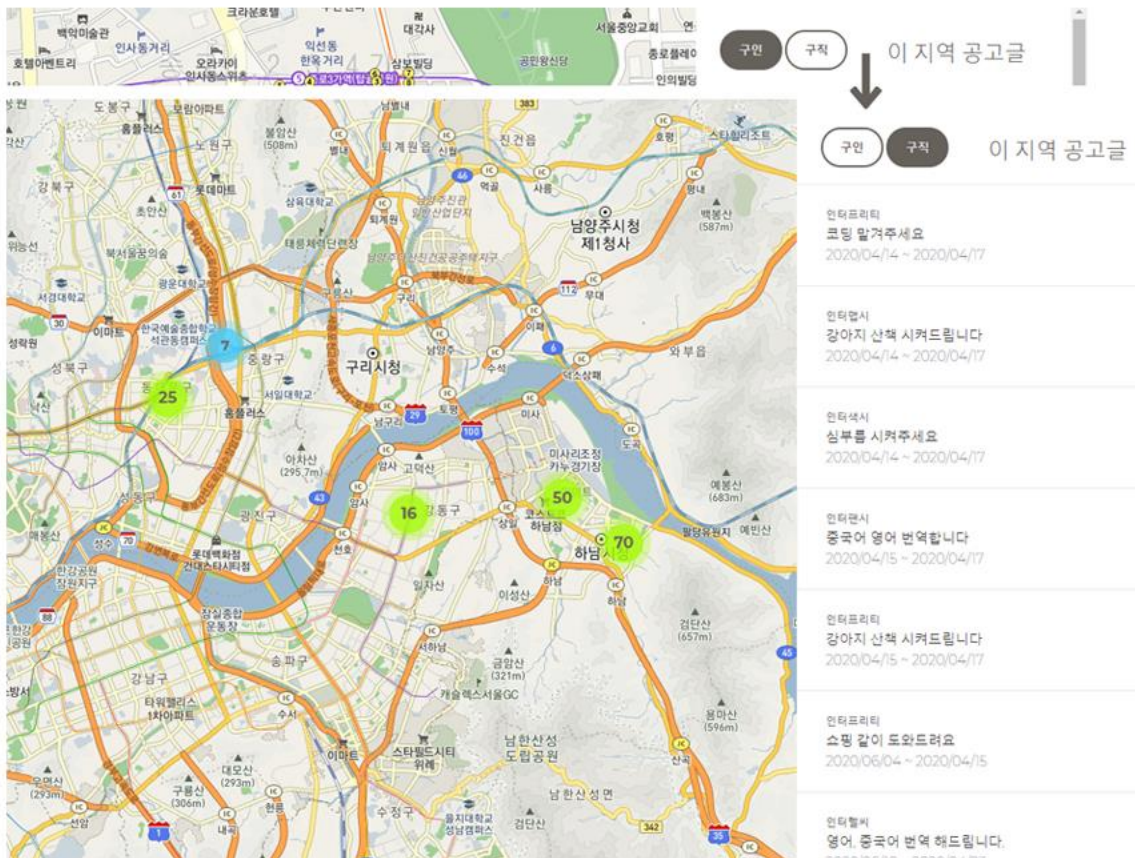
class JsonIpClient {

    private static final String USER_AGENT = "Mozilla/5.0";
    private static final String GET_URL = "http://jsonip.com/";

    public String sendGet() throws ClientProtocolException, IOException {
        //http client 생성
        CloseableHttpClient httpClient = HttpClients.createDefault();
        //get 메서드와 URL 설정
        HttpGet httpGet = new HttpGet(GET_URL);
        //agent 정보 설정
        httpGet.addHeader("User-Agent", USER_AGENT);
        httpGet.addHeader("Content-type", "application/json");
        //get 요청
        CloseableHttpResponse httpResponse = httpClient.execute(httpGet);
        String json = EntityUtils.toString(httpResponse.getEntity(), "UTF-8");
        httpGet.close();
        return json;
    }
}
```


지도검색

지도를 움직이거나, 확대, 검색을 하게 되면 **지도가 변경**되는데
이때 화면 범위의 실제 위도,경도를 실시간으로 받아내고
구인/구직 버튼 클릭에 맞춰 **ajax로 현재 좌표에 해당하는 글만 뽑아**
동적으로 리스트가 변경되도록 구현했습니다.



[VIEW]

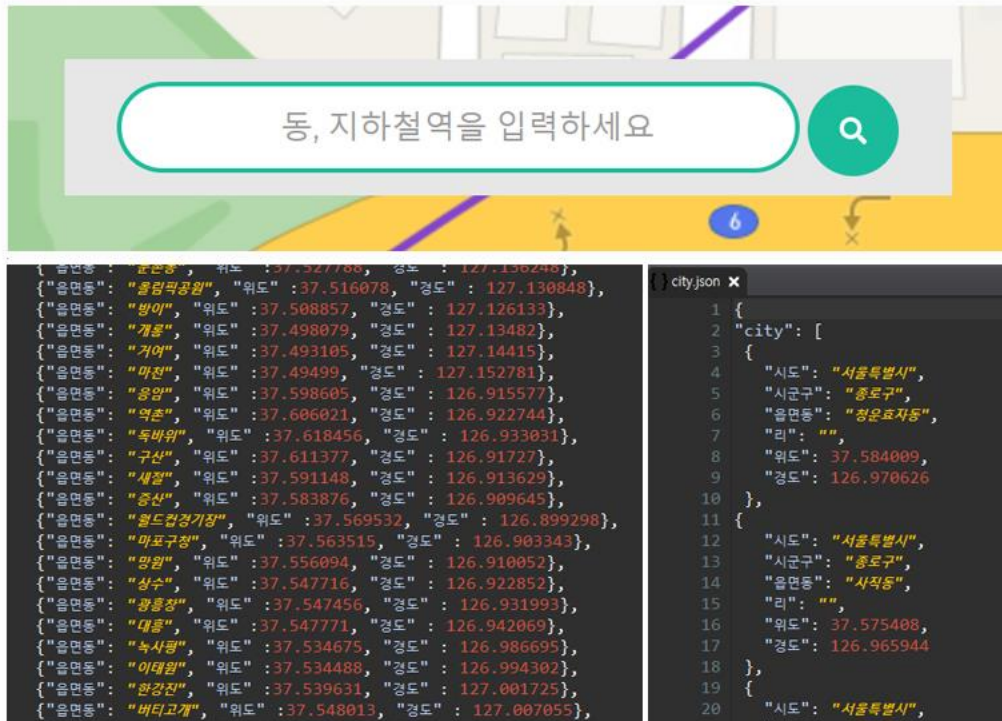
```
//리스트에 현재지도영역에 포함되는 글만 가져오는 함수
function mapList(lldata) {
    $("#placesList").scrollTop(0); // 스크롤 항상 위로 지정
    $.ajax({
        url: "${pageContext.request.contextPath}/map2",
        type: "POST",
        data: lldata,
        contentType: "application/json",
        dataType: "json",
        success: function(data) {
            var list = document.getElementById("List");
            var itemStr = '';
            //리스트모양만들기(html코드생성)
            for(i = 0; i < data.list.length; i++){
                itemStr += '<div class="info">' +
                    '<b><small>' + data.list[i].usercode + '</b></small><br>' ;
                if(data.list[i].ebcode != ' ' || null){
                    itemStr += '<span><a class="atag" href="<c:url value="employerBoard/read/">
                        + data.list[i].ebcode + "</>" target="_blank" >'
                        + data.list[i].title + '</a></span>'; //계시글 연결
                }else{
                    itemStr += '<span><a class="atag" href="<c:url value="workerBoard/read/">
                        + data.list[i].wbcode + "</>" target="_blank">'
                        + data.list[i].title + '</a></span>'; //계시글 연결
                }
                itemStr += '<br><span class="date">' + data.list[i].startdate + ' ~ '
                    + data.list[i].enddate + '</span>' + '</div>';
            }
            $("#List").html(itemStr);
        },
        error: function(errorThrown) {
        }
    });
}
```

[CONTROLLER]

```
//이 요청은 리스트를 만들기 위한 준비
@RequestMapping(value="/map2", produces = "text/json; charset=utf-8")
@ResponseBody
public String getItem(@RequestBody String lldata) throws ParseException {
    String data = llMapService.getItem(lldata);
    return data;
}
```

지도검색

지도 API의 검색기능이 아닌 실제 전국 주소와 지하철역 좌표 데이터 정보를 수집하여 제이슨 파일로 만들었습니다. 검색창에 입력한 값을 제이슨 파일에서 찾아 검색 후 위치가 변경되도록 검색 기능을 구현하였습니다.



//검색어 제이슨파일 검색 준비

```
function getXHR() {
    var req;
    try {
        req = new XMLHttpRequest();
    } catch (e) {
        try {
            req = new ActiveXObject("Msxm12.XMLHTTP");
        } catch (e) {
            req = new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return req;
};
```

//검색어 제이슨파일에서 검색

```
function asyncSend() {
    var req = getXHR();
    var keyword = $('#q').val(); //검색어
```

//제이슨파일에 지하철역이름이 '역' 들어있지않아서 검색 시 모두 찾을 수 있게 준비

```
if(keyword.indexOf('역') != -1){
    keyword = keyword.substr(0, keyword.length -1);
}
```

```
req.onreadystatechange = function() {
    if (req.readyState == 4) {
        if (req.status == 200) {
            jsonData = JSON.parse(req.responseText);
            for (i = 0; i < jsonData.city.length; i++) {
                if (jsonData.city[i].읍면동 == keyword) { //제이슨에서 검색어 찾기
                    lat = jsonData.city[i].위도;
                    lng = jsonData.city[i].경도;
                    var moveLatLon = new kakao.maps.LatLng(lat, lng);
                    panTo(moveLatLon);
                    break;
                }
            }
        }
    }
    req.open('GET', 'searchJson/city.json');
    req.send(null);
};
```


지도검색

리스트에서 원하는 글을 클릭하면 해당 글로 이동



- 고글

인터프리티
물건 대신 팔아주실분
2020/04/10 ~ 2020/04/12

인터프리티
영어 짧은거 번역해주실 수 있나요
2020/04/10 ~ 2020/04/12

인터프리티
카페 6시간 인력 구함
2020/04/10 ~ 2020/04/12

인터팬시
강아지 산책 부탁해요
2020/04/10 ~ 2020/04/12

인터섹시
옷 골라주실 분
2020/04/10 ~ 2020/04/12

인터맵시
물건 대신 팔아주실분
2020/04/10 ~ 2020/04/12



수정 삭제

영어 짧은거 번역해주실 수 있나요

작성자 : 김코딩
게시글코드 : E46

주소	서울특별시 동대문구 서울시립대로29나길 15-5 (휘경동)
근무시작/종료일	2020/04/11 ~ 2020/04/13
근무시간대	오후
급여	10000원
업적종	사무직
나이	30대
공고마감일	2020/06/11

영어 짧은 글 번역해 주실 분 구합니다!
채팅 걸어주세요!



상세검색

form에 다양한 검색 필터를 준비하였고 시/도를 선택하면 선택한 시/도에 맞는 구/군이 나타나도록 구현하였습니다.

각 카테고리는 다중 선택도 가능하고 선택을 하지 않을 수 도 있습니다.

그렇기 때문에 **MyBatis**의 if문과 foreach문을 사용하여 **사용자가 선택한 값이 쿼리로 만들어지게** 구현하였습니다.

상세검색

시/도

●서울특별시

●인천광역시

●대전광역시

●광주광역시

●대구광역시

●울산광역시

●부산광역시

●경기도

●강원도

●충청북도

●충청남도

●전라북도

●전라남도

●경상북도

●경상남도

●제주특별자치도

구/군

■강남구

■강동구

■강북구

■강서구

■관악구

■광진구

■구로구

■금천구

■노원구

■도봉구

■동대문구

■동작구

■마포구

■서초구

■서대문구

■성동구

■성북구

■송파구

■양천구

■영등포구

■용산구

■은평구

■종로구

■중구

■종로구

■중랑구

근무 시작일

2020-04-27

근무 종료일

2020-04-27

근무 시간대

☐오전

☐오후

☐상시

업적종

☐서비스

☐사무업

☐디자인

☐운전/배달

☐교육

☐IT/컴퓨터

☐단순

나이대

☐10대

☐20대

☐30대

☐40대

☐50대

[QUERY]

```
<sql id="searchCriteria">
<if test="addr.size != 0">
    and REGEXP_LIKE(ADDRESS,
    <foreach item="addrItem" collection="addr" open='' close='' index = "index" separator="||">
        #{addrItem}
        <if test="index != addr.size-1">
            ||'|'
        </if>
    </foreach>
    )</if>

    <if test="workStartDate != null and workEndDate != null">
        and WORKSTART <![CDATA[   >= #{workStartDate} and WORKEND <= #{workEndDate} ]]>
    </if>

    <if test="worktime.size != 0">
        and REGEXP_LIKE(WORKTIME,
    <foreach item="worktimeItem" collection="worktime" open='' close='' index = "index" separator="||">
        #{worktimeItem}
        <if test="index != worktime.size-1">
            ||'|'
        </if>
    </foreach>
    )</if>

    <if test="job.size != 0">
        and REGEXP_LIKE(JOB,
    <foreach item="jobItem" collection="job" open='' close='' index = "index" separator="||">
        #{jobItem}
        <if test="index != job.size-1">
            ||'|'
        </if>
    </foreach>
    )</if>

    <if test="age.size != 0">
        and REGEXP_LIKE(AGE,
    <foreach item="ageItem" collection="age" open='' close='' index = "index" separator="||">
        #{ageItem}
        <if test="index != age.size-1">
            ||'|'
        </if>
    </foreach>
    ) </if>

</sql>
```

```
//인자로 보낼 객 만들기
HashMap<String,Object> detail = new HashMap<String,Object>();

detail.put("addr", addr);
detail.put("workStartDate", newStartDate);
detail.put("workEndDate", newEndDate);
detail.put("worktime", worktimeArr);
detail.put("job", jobArr);
detail.put("age", ageArr);
detail.put("rowStart", rowStart);
detail.put("rowEnd", rowEnd);

detail.forEach((key, value)
-> System.out.println("key: " + key + ", value: " + value));

List<EmployerBoardVO> data = employerBoardService.detailSearch(detail);
model.addAttribute("employerBoardList", data);

[ CONTROLLER ]
```

자유게시판 CRUD

게시판에 필요한 기본적인 생성, 읽기, 수정, 삭제는 **REST 방식**으로 구현하고

수정버튼과 삭제버튼은 JSTL을 사용해 **세션에 저장된 ID와 작성자ID를 비교하여 동일할 경우에만 보여지게** 코드를 작성하였습니다.

C

```
// 새 글 작성을 위한 요청을 처리
@RequestMapping(value = "/freeWrite", method = RequestMethod.GET)
public String write(HttpServletRequest request, Model model) {

    // 새 글 등록을 위한 요청을 처리
    @RequestMapping(value = "/freeWrite", method = RequestMethod.POST)
    public String write(@ModelAttribute("newFree") FreeBoardVo vo) {
        System.out.println("vo도착");
        freeBoardService.write(vo);
        return "redirect:/freeList";
    }
}
```

R

```
// 특정 게시글 보기
@RequestMapping(value = "/freeList/read/{freecode}")
public String read(HttpServletRequest request, Model model, @PathVariable String freecode) {
    HttpSession session = request.getSession();
    String id = (String) session.getAttribute("myUserCode");
    model.addAttribute("myUserCode", id);
    model.addAttribute("freeBoardVo", freeBoardService.freeRead(freecode));
    return "freeboard/freeRead";
}
```

U

```
// 글 수정
@RequestMapping(value = "/edit/{freecode}", method = RequestMethod.GET)
public String edit(@PathVariable String freecode, Model model) {

    @RequestMapping(value = "/edit/{freecode}", method = RequestMethod.POST)
    public String edit(@ModelAttribute("edit") FreeBoardVo vo) {
        System.out.println("수정할 vo도착");
        freeBoardService.edit(vo);
        return "redirect:/freeList";
    }
}
```

D

```
//글 삭제 요청 처리
@RequestMapping(value="/freeBoard/delete/{freecode}", method=RequestMethod.GET)
public String ebDelete(@PathVariable String freecode, Model model) {
    model.addAttribute("freecode", freecode);
    return "freeboard/freeDelete";
}

//글 삭제
@RequestMapping(value = "/freeBoard/delete", method = RequestMethod.POST)
public String delete2(String freecode, Model model) {
    freeBoardService.delete(freecode);
    return "redirect:/freeList";
}
```

자유게시판

여러가지 옵션을 준비해 사용자가 입력한 선택한 옵션과 검색어에 맞춰

한가지 쿼리를 조건에 맞춰 재사용할 수 있도록 마이바티스로 **동적쿼리**를 준비하여

검색기능을 구현하였습니다.

일프로 · 일나눔 · 소통을 위한 통합게시판

no.	제목
F137	사진테스트
F136	반갑습니다
F134	부천 만두리아 꼭 가세요

제목 ▼

제목

내용

작성자

제목+내용

요즘 일자리 정말 없네요..

회기 맛집 추천합니다

오늘 날씨 엄청 좋네요

안녕하세요!!

반갑습니다

안녕하세요!!

요즘 일자리 정말 없네요..

제목

검색어를 입력하세요

1

2

3

4

5

>

[VIEW]

```
<!-- 검색창 -->
<div class="search row">
  <div class="col-xs-2 clo-sm-2">
    <select name="searchType" class="form-control">
      <option value="t">
        <c:out value="${sc.searchType eq 't' ? 'selected' : ''}" />>제목</option>
      <option value="c">
        <c:out value="${sc.searchType eq 'c' ? 'selected' : ''}" />>내용</option>
      <option value="w">
        <c:out value="${sc.searchType eq 'w' ? 'selected' : ''}" />>작성자</option>
      <option value="tc">
        <c:out value="${sc.searchType eq 'tc' ? 'selected' : ''}" />>제목+내용</option>
    </select>
  </div>
</div>
```

[QUERY]

```
<sql id="search">
  <if test="searchType != null">
    <if test="searchType == 't'.toString()">and TITLE like '%' || #{keyword} || '%</if>
    <if test="searchType == 'c'.toString()">and CONTENT like '%' || #{keyword} || '%</if>
    <if test="searchType == 'w'.toString()">and WRITERCODE like '%' || #{keyword} || '%</if>
    <if test="searchType == 'tc'.toString()">and (TITLE like '%' || #{keyword} || '%') or (CONTENT
      like '%' || #{keyword} || '%')</if>
  </if>
</sql>

<select id="ListSearch" resultType="FreeBoardVo"
  parameterType="Search">
  SELECT FREECODE, WRITERCODE, TITLE, CONTENT, to_char(REGDATE,'YYYY/MM/DD')as REGDATE,HIT
  from (
    select NUM, FREECODE, WRITERCODE,
    TITLE, CONTENT, REGDATE, HIT, row_number()
    over(order by NUM desc) as rNum from FREEBOARD
    <![CDATA[ where NUM > 0 ]]>
    <include refid="search"></include>
  )
  FREEBOARD where rNum between #{rowStart} and #{rowEnd}
  order by NUM desc
</select>
```


자유게시판

페이징에 필요한 페이지 정보를 담는 변수들과 메서드를 따로 관리할 수 있게
VO를 따로 생성하고 검색어, 검색 옵션을 관리할 수 있는 VO를 만들어
전체 목록 페이징과 검색결과에 따라 페이징이 변경 될 수 있도록 구현하였습니다.

글쓰기

작성자	작성일	조회수
인터헬씨	2020/04/09	238
garnet2929	2020/04/09	24
인터색시	2020/04/09	9
인터팬시	2020/04/08	10
인터맵시	2020/04/08	6
인터프리티	2020/04/08	7
브룩	2020/04/07	49
garnet2929	2020/04/07	6
인터색시	2020/04/07	6
인터패시	2020/04/06	6

< 6 7 8 9 10 >

검색

1 2 3 4 5 >

[VIEW]

```
<!-- 페이징 -->
<div id="page">
<div class="col-md-offset-3">
  <ul class="pagination">
    <c:if test="${pageMaker.prev}">
      <li class="page-item">
        <a class="page-link"
          href="freeList${pageMaker.makeSearch(pageMaker.startPage - 1)}">&lt;</a></li>
    </c:if>

    <c:forEach begin="${pageMaker.startPage}"
      end="${pageMaker.endPage}" var="idx">
      <li>
        <c:out value="${pageMaker.pagerd.page == idx ? 'class=active' : ''}" />
        <a href="freeList${pageMaker.makeSearch(idx)}">${idx}</a>
      </li>
    </c:forEach>

    <c:if test="${pageMaker.next && pageMaker.endPage > 0}">
      <li class="page-item">
        <a class="page-link"
          href="freeList${pageMaker.makeSearch(pageMaker.endPage + 1)}">&gt;</a></li>
    </c:if>
  </ul>
```

[CONTROLLER]

```
public String listPage(HttpServletRequest request, @ModelAttribute("sc")
    Search_sc, Model model) {
    PageMaker pageMaker = new PageMaker();
    pageMaker.setPagerd(sc);
    pageMaker.setTotalCount(freeBoardService.countSearch(sc));
    model.addAttribute("pageMaker", pageMaker);
}
```

[QUERY]

```
<select id="countSearch" resultType="int">
  select count(NUM)
  from FREEBOARD <![CDATA[ where NUM > 0 ]]>
  <include refid="search"></include>
</select>
```

```
public class PageMaker {
    private int totalCount; //게시물 총 갯수
    private int startPage; // 현재 페이지의 시작번호
    private int endPage; //현재 페이지의 끝번호
    private boolean prev; //이전 페이지로 이동하는 링크의 존재 여부
    private boolean next; //다음 페이지로 이동하는 링크의 존재 여부
}

public class Search extends PageReady {
    private String searchType = "";
    private String keyword = "";
}
```

자유게시판

사용자가 원하는대로 꾸며 자유롭게 업로드할 수 있도록 **썸머노트 에디터 API**를 사용
썸머노트는 base64로 인코딩 후 저장하는 방식으로 **이미지 파일 관리가 어려움** 그래서
callback을 이용하여 이미지를 특정 경로에 업로드 후 고유한 url를 저장하여 관리하
는 방식으로 구현했습니다.

제목을 입력하세요.

내용을 입력하세요

글 작성

목록으로

[VIEW]

```
function uploadSummernoteImageFile(file, editor) {
    data = new FormData();
    data.append("file", file);
    $.ajax({
        data: data,
        type: "POST",
        url: "${pageContext.request.contextPath}/uploadSummernoteImageFile",
        contentType: false,
        processData: false,
        dataType: "json",
        success: function(data) {
            $(editor).summernote('insertImage', data.url);
        }
    });
}

callbacks: {
    onImageUpload : function(files) {
        uploadSummernoteImageFile(files[0],this);
    }
}
```

[CONTROLLER]

```
// (예이엑스로) 이미지 업로드 했을 때, 외부 리소스 경로에 파일 업로드
@RequestMapping(value = "/uploadSummernoteImageFile", produces = "application/json",
    method = RequestMethod.POST)
@ResponseBody
public String uploadSummernoteImageFile(@RequestParam("file") MultipartFile multipartFile) {
    JSONObject jsonObject = new JSONObject();
    String originalFileName = multipartFile.getOriginalFilename(); // 오리지널 파일명
    String extension = originalFileName.substring(originalFileName.lastIndexOf(".")); // 파일 확장자
    String savedFileName = UUID.randomUUID() + extension; // 저장될 파일명 (UUID는 파일명의 중복 방지를 위해 랜덤 생성)
    File targetFile = new File(fileRoot + savedFileName);
    try {
        InputStream fileStream = multipartFile.getInputStream();
        FileUtils.copyInputStreamToFile(fileStream, targetFile); // 파일 저장
        jsonObject.put("url", "/interpreter/summernoteImage/" + savedFileName);
    } catch (IOException e) {
        e.printStackTrace();
    }
    return jsonObject.toString();
}
```

[XML]

```
<!-- 요청url을 location에 설정된 서버의 로컬경로를 맵핑 -->
```

```
<mvc:resources mapping="/summernoteImage/**"
    location="file:///d:/dev/new/"></mvc:resources>
```

자유게시판

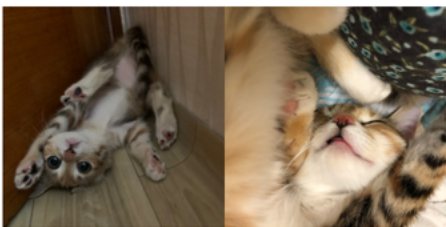
Ajax를 활용하여 댓글을 구현하였고 로그인된 상태에서만 댓글을 등록할 수 있도록 하고 세션에 담긴 ID와 댓글을 등록한 ID를 비교하여 동일 할 경우만 수정과 삭제 버튼이 생성 되도록 구현하였습니다.

안녕하세요! 저희 집 고양이 보고가세요

수정하기 삭제하기



2020/04/22



우리집 고양이에요 너무 귀엽죠!

Comments 2

앞으로 많이 올려주세요!!

등록

브록

앞으로 많이 올려주세요!!

인터프리터

너무 귀여워요 ㅠㅠ

[VIEW]

```
<div class="container"> <!-- 댓글창 -->
  <form id="commentListForm" name="commentListForm" method="post">
    <div id="commentList"></div>
  </form>
</div>

/* 댓글 불러오기(Ajax)*/
function getCommentList() {
  var articleCode = $('#articleCode');

  $.ajax({
    type : 'post',
    url : '${pageContext.request.contextPath}/freeboard/commentList',
    data : articleCode.val(),
    contentType : "application/String",
    success : function(data) {
```

Ajax 성공 시(JS) 뷰페이지의 값을 삽입한다.

```
$("#commentList").html(html);
```

[CONTROLLER]

```
// 댓글 리스트 읽어오기
@RequestMapping(value = "/freeboard/commentList", produces = "text/json;charset=UTF-8")
public String commentList(@RequestBody String articleCode, HttpServletRequest request, Model model)
    throws Exception {
    String myUserCode = getUserId(request, model);
    ArrayList<HashMap> hmlist = new ArrayList<HashMap>();
    List<FreeCmtVO> commentVO = freeBoardService.freeSelectCmt(articleCode);

    // 해당 게시물 댓글
    if (commentVO.size() > 0) {
        for (int i = 0; i < commentVO.size(); i++) {
            HashMap hm = new HashMap();
            hm.put("num", commentVO.get(i).getNum());
            hm.put("userCode", commentVO.get(i).getUserCode());
            hm.put("comment", commentVO.get(i).getComments());
            hm.put("registDate", commentVO.get(i).getRegistDate());
            hmlist.add(hm);
        }
    }
    model.addAttribute("hmlist", hmlist);
    return "freeboard/commentView";
}
```

댓글jsp를 따로 만들어 댓글이 생성된 뷰페이지를 받아온다.

고객센터 1:1문의하기

- 1:1문의는 로그인 한 후에만 작성 가능하도록 구현했습니다.
- 셀렉트박스에서 불만 및 신고 옵션 클릭 시에만 회원 확인 기능이 활성화되고 가입된 회원만 검증하여 신고할 수 있게 구현했습니다.
- MultipartFile을 이용해 다중 파일 업로드가 가능합니다.

1:1 문의하기

제목

상담유형

이용문의

설명

문의 내용을 입력하세요.

신고 회원 닉네임

회원확인

첨부파일

파일 선택

파일 8개

등록

다시쓰기

상담유형

불만 및 신고

이용문의

불만 및 신고

기타

문의 내용을 입력하세요.

192.168.30.72:8080 내용:

존재하지 않는 회원입니다.

확인

[VIEW]

```
//닉네임칸 활성화
$('#inquirytype').change(function(){
    if($("#select[name=inquirytype]").val() == "불만 및 신고"){
        $("input[name=complain]").attr('disabled', false);
        $('#usercodeCheck').attr('disabled', false);
    }else {
        $("input[name=complain]").attr('disabled', true);
        $('#usercodeCheck').attr('disabled', true);
    }
});
```

[CONTROLLER]

```
// 1:1 글 등록을 위한 요청을 처리 다중파일업로드가능
@RequestMapping(value = "/qnaStep2", method = RequestMethod.POST)
public String write(@ModelAttribute("qna")QnaRequestVo vo,
    @RequestParam("uploadFile") List<MultipartFile> file) {
    csCenterService.write(vo);
    int currval = csCenterService.currval();
    String qno = "Q"+ currval;
    List<MultipartFile> files = file;
    for(int i = 0; i < files.size(); i++) {
        MultipartFile f = files.get(i);
        UUID uuid = UUID.randomUUID(); //랜덤이름 만들기
        if(!f.getOriginalFilename().isEmpty()) { //파일을 첨부하지 않아도 무조건 인자로 빈 파일이 넘어오기 때문에 조건 추가
            int index = f.getOriginalFilename().lastIndexOf('.');
            String format = f.getOriginalFilename().substring(index);
            String saveName = uuid + format;
            //저장하기전 경로랑 이름지정
            File saveFile = new File("D:\\dev\\file\\",saveName); // 저장할 폴더 이름, 저장할 파일 이름
            String fileName = saveFile.getName();
            try {
                FileCopyUtils.copy(f.getBytes(),saveFile); //진짜 물리적 저장은 여기서 진행
            } catch (IOException e) {
                e.printStackTrace();
            }
            HashMap<String,String> fileinsert = new HashMap<>();
            fileinsert.put("qno", qno);
            fileinsert.put("fileName", fileName);
            csCenterService.filewrite(fileinsert);
        }
    }
}
```

**감사합니다.
꼭 필요한 인재가
되겠습니다!**

CONTACT

이 여 울

sleepy2head@naver.com

010 9120 4349