

JSP 기반 Team Project

웹보 중고마켓 (+API)



Date: 2022.03

Name: 최성광

Phone: 010-9992-6962

Email: sungkwang0908@naver.com

Blog: <https://blog.naver.com/sungkwang0908>

GitHub: <https://github.com/CHOISUNGKWANG/Team-Web-Project.git>

목차

1. 프로젝트 소개
2. 개발 환경
3. Version
4. Timeline
5. User Flow
6. ERD
7. Logic Process
8. 프로그램 주요기능
9. 담당 View 파트 핵심 기능 및 로직
10. Map Api & Mail Api 구현 과정
11. 소감 및 추후 개선사항

프로젝트 소개

기획배경

- **타겟 사이트의 웹 페이지 제작**

당근마켓 어플을 벤치마킹 하여
당근마켓의 웹 버전을 생각하며 구상

- **중고제품 거래**

사용자가 직접 사진과 가격을 설정하
여 중고 제품을 판매 및 구매 할 수 있
도록 기획 진행

목적

- **검색페이지 필터 세부화**

타겟 사이트인 당근마켓을 벤치마킹
하며 웹브조 만의 검색 페이지를
적용시켜 구현

- **판매자의 거래 가능지역**

직거래 목적으로 판매자의 거래 가능 반
경을 지도로 한 눈에 볼 수 있게 구현

기대효과

- **코드 분석 능력 향상**

타겟 사이트를 벤치마킹하여, 광고 배
너나 검색 필터같은 기능을 이해하고
활용가능

- **API에 대한 이해**

맵 API와 이메일/SMS API를 페이지에
맞게 활용함으로써
Develop 과정에서 채팅 API를 활용해
판매자와 구매자 간의 실시간 소통이
가능하게 구현할 예정

개발 환경



Language



Web Standard



Database



Server



JSP



Communication

버전 관리

Ver.1

- **타겟 사이트 : 당근마켓**

타겟 사이트에서 구현되어 있는 대로 UserFlow 구상

- **기본 로직 구현**

MVC를 각자 담당하여 설계에서 확정된 템플릿을 활용하여 기본적인 로직 구현

Ver.2

- **DB 수정**

3개의 테이블에서 겹치는 칼럼 삭제, 찜 버튼 활용을 위한 칼럼 추가

- **비즈니스 메서드 추가**

기존의 겹치는 칼럼 삭제로 인해, 더 상세한 비즈니스 메서드 추가

Ver.3

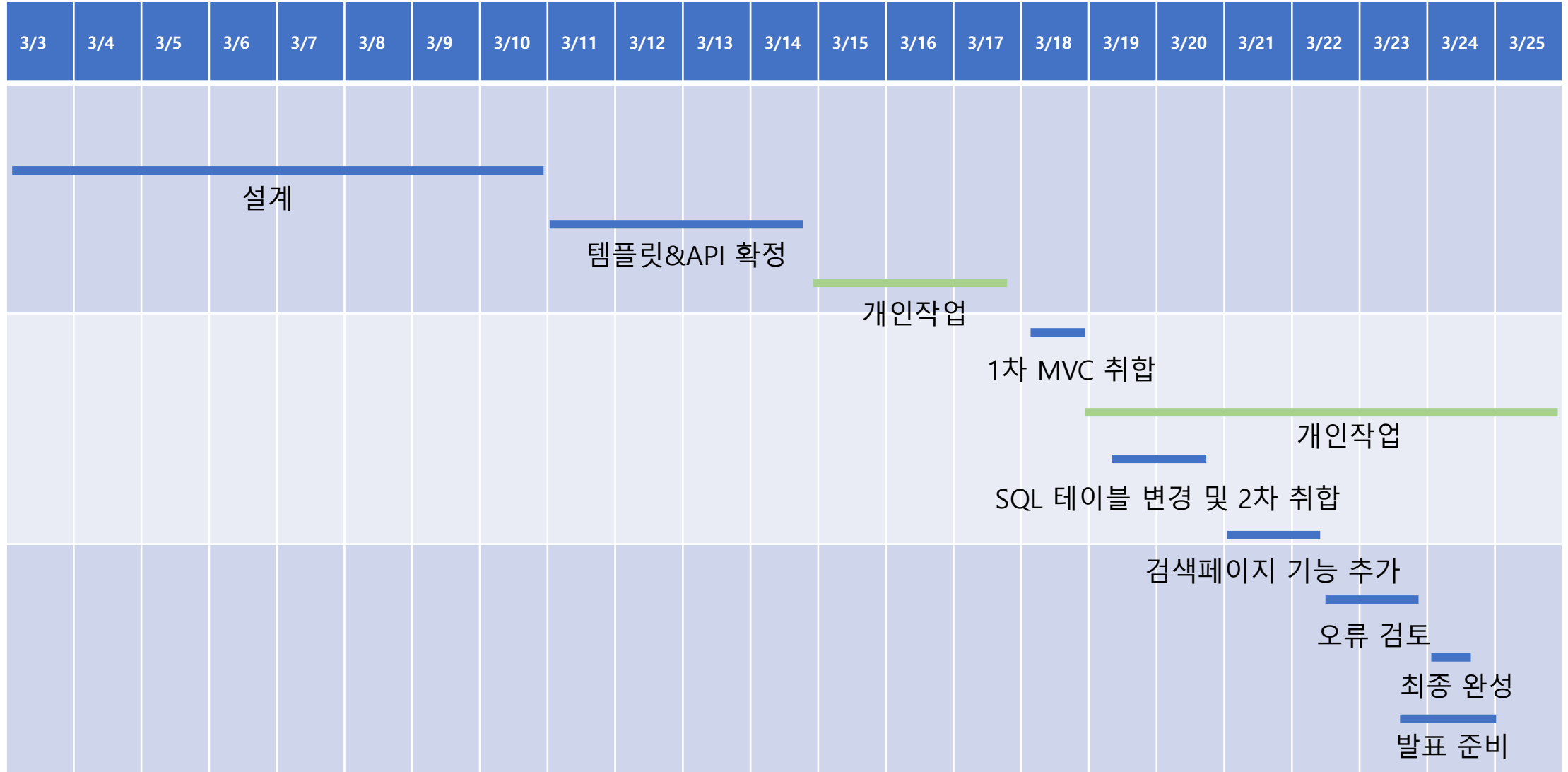
- **검색페이지 필터 세부화**

제목/판매자ID/금액 검색만 가능했으나, 최신순/조회순으로 정렬 할 수 있도록 하고자 함

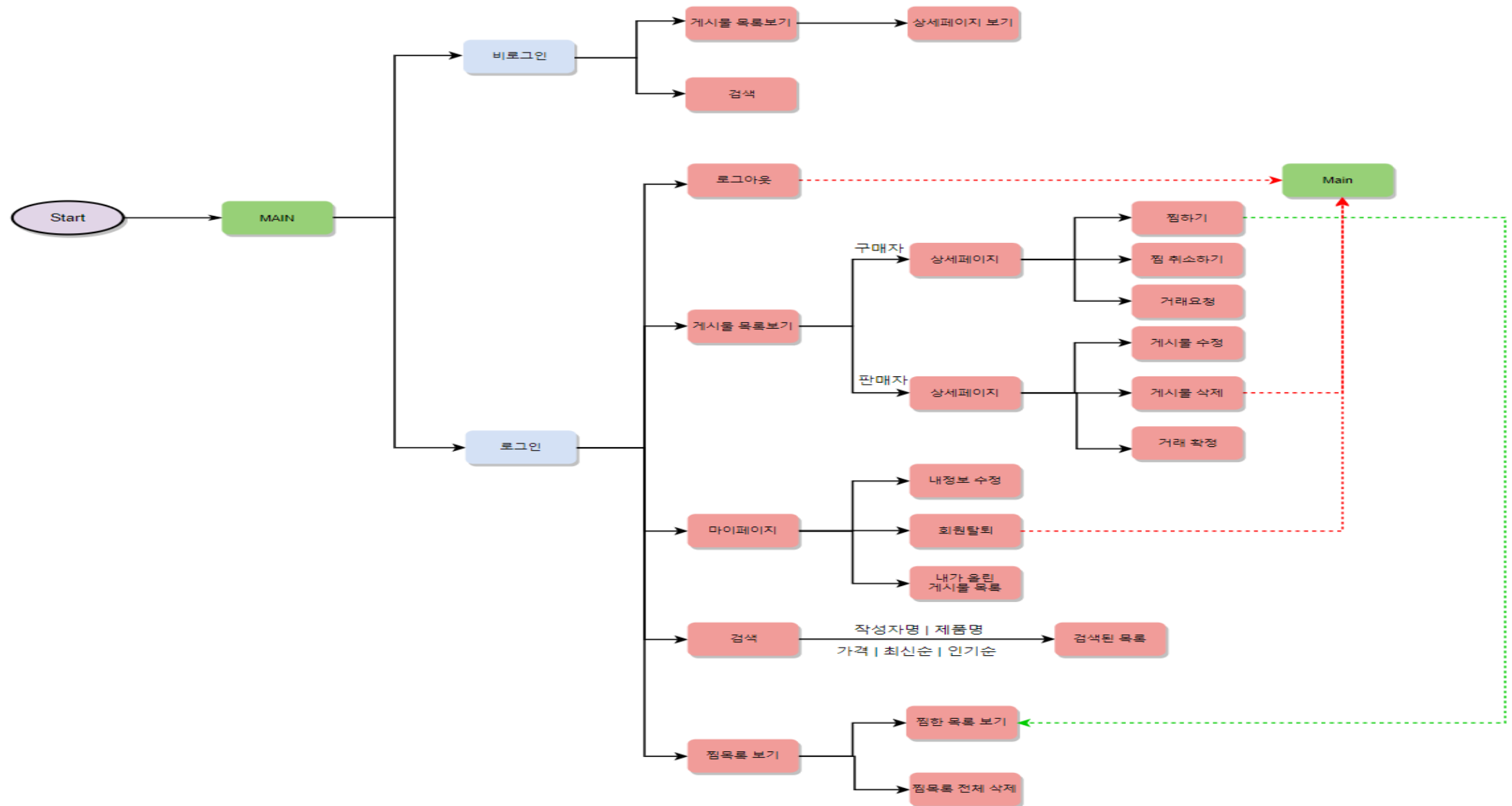
- **비즈니스 메서드 추가**

VO를 추가하여 기존 활용하던 제목/판매자/금액 검색 메서드에서 조회순, 최신순을 추가한 메서드로 변경함

개발 일정



User Flow



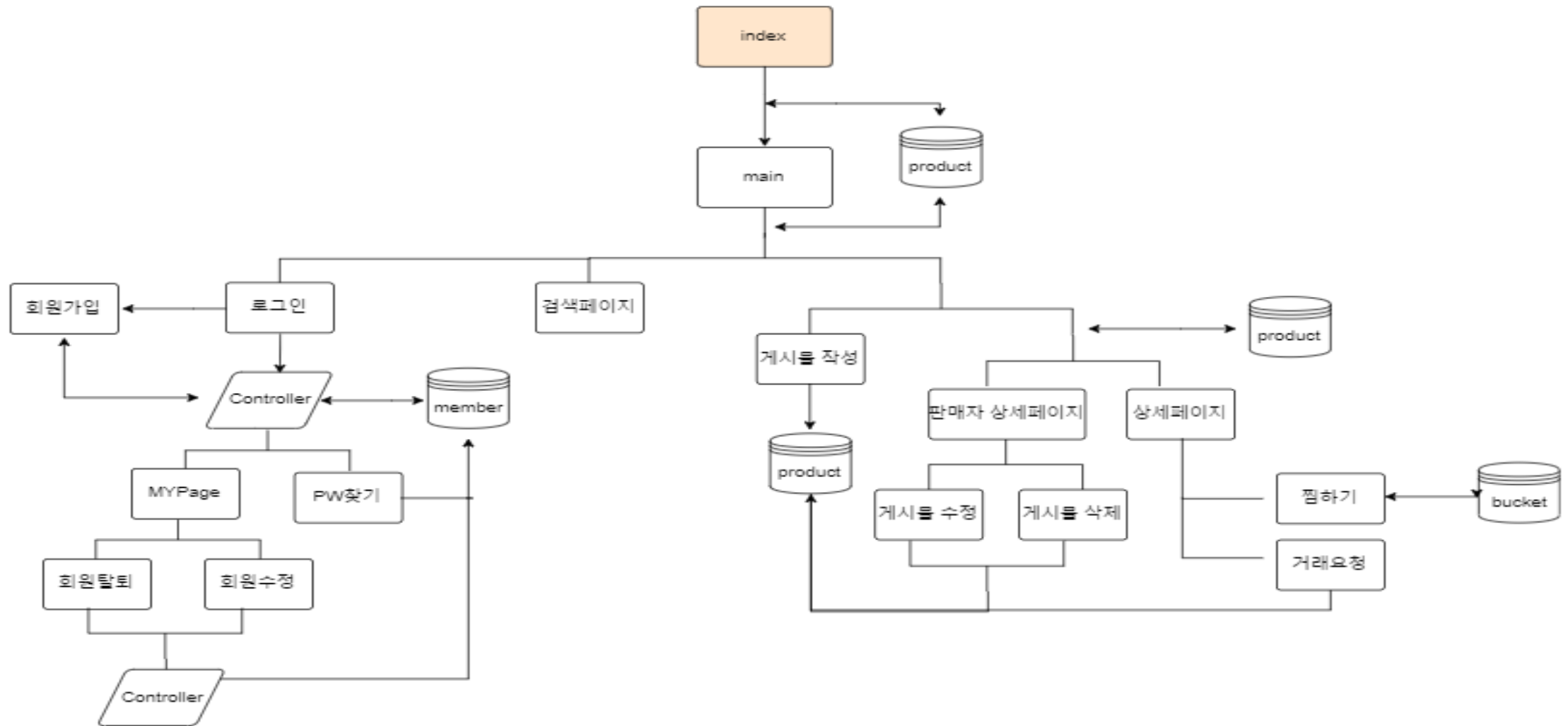
ERD(ver.2)

member		
	name	type
아이디	mid	varchar(50)
비밀번호	mpw	varchar(50)
이름	mname	varchar(20)
닉네임	mnick	varchar(20)
핸드폰 번호	mphone	varchar(30)
주소	addr	varchar(70)
상세주소	addrapt	varchar(100)
이메일	memail	varchar(100)

product		
	name	type
제품번호	ppk	int
제품제목	ptitle	varchar(100)
게시물사진	ppic	varchar(100)
상세설명	pcontent	varchar(300)
가격	pprice	int
판매자 ID	pmid	varchar(20)
판매날짜	pdate	date
거래버튼 상태	pstate	Int
조회수	pcnt	Int
찜횟수	pbkcnt	int

bucket		
	name	type
제품번호	productpk	int
이용자 아이디	memberid	varchar(30)

Logic Process

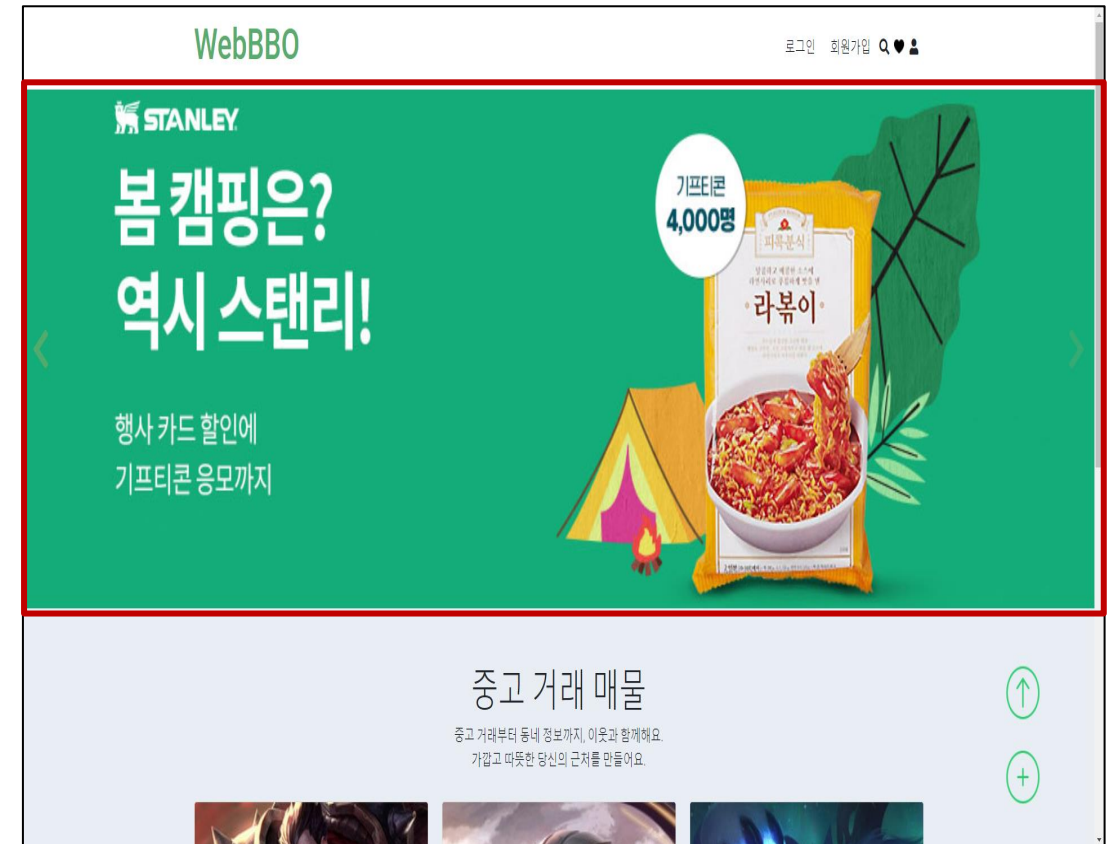


View

슬라이더 배너 & json

- json 파일에 샘플 사진과 샘플 URL 정보를 넣고
ajax로 샘플 데이터를 불러와서 구현

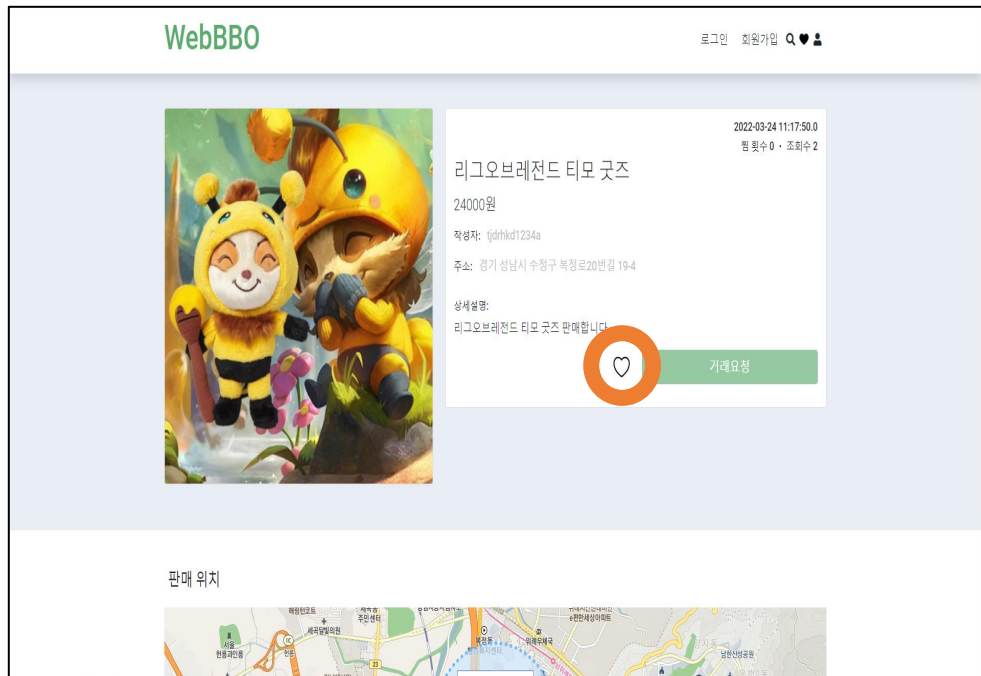
```
[  
  {"src": "/assets/img/add1.png", "href": "https://store.emart.com/news/event/notice_event_view.do?master_cd=16727&trcknCode=mainbanner5"},  
  {"src": "/assets/img/add2.png", "href": "https://store.emart.com/news/event/notice_event_view.do?master_cd=16663&trcknCode=mainbanner8"},  
  {"src": "/assets/img/add3.png", "href": "https://store.emart.com/news/event/notice_event_view.do?master_cd=16746&trcknCode=mainbanner1"}  
]
```



View

Ajax

- 찜하기 버튼
- 찜횟수



```
<script type="text/javascript">
    $("#heartbtn").on("click", function() {

        if(${id} != null) {

            var productpk = "${pdata.ppk}";
            var memberid = "${id}";

            $.ajax({
                type : "post",
                url : "heartUrl.jsp",
                data : {
                    "productpk" : productpk,
                    "memberid" : memberid
                },
                dataType : "json",
                success : function(result) {

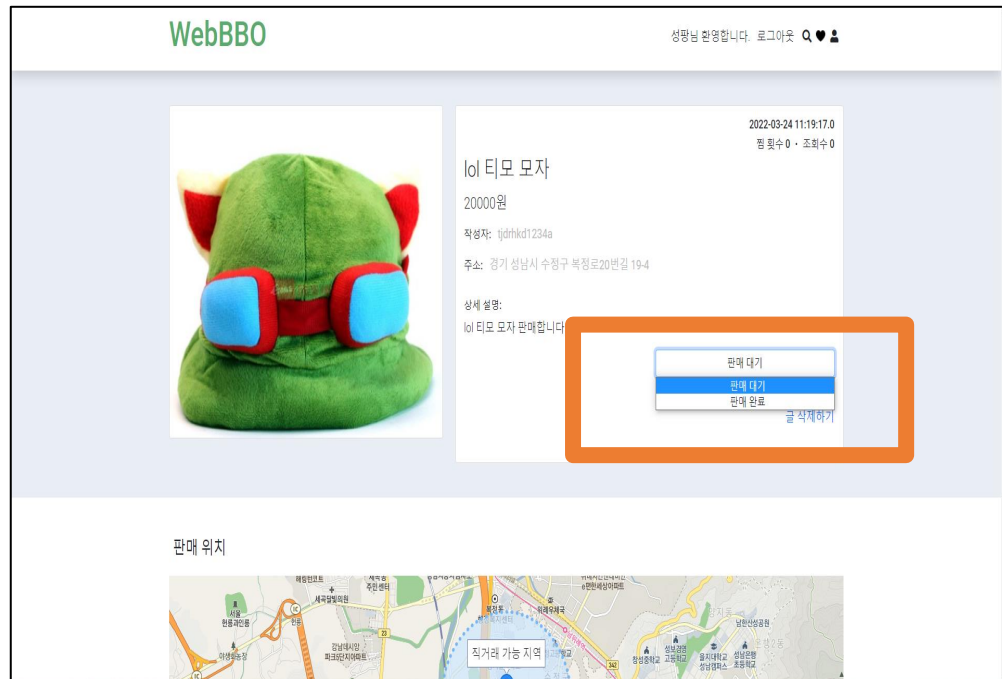
                    console.log(result);

                    if(${bkdata} == null){
                        if (result == 1) {
                            $("#heartbtn").attr('src', 'assets/img/heart (1).png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt+1} · 조회수 ${pdata.pcnt}");
                        } else {
                            $("#heartbtn").attr('src', 'assets/img/heart.png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt} · 조회수 ${pdata.pcnt}");
                        }
                    }
                    else{
                        if (result == 1) {
                            $("#heartbtn").attr('src', 'assets/img/heart (1).png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt} · 조회수 ${pdata.pcnt}");
                        } else {
                            $("#heartbtn").attr('src', 'assets/img/heart.png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt-1} · 조회수 ${pdata.pcnt}");
                        }
                    }
                },
                error : function() {
                    console.log('에러 발생');
                    console.log(err.status + " | " + err.errText);
                }
            });
        }
        else{
            alert('로그인 후 이용 부탁드립니다!');
        }
    });
</script>
```

View

Ajax

- 판매대기 판매완료 버튼



```
<script type="text/javascript">
$("#state").on("change", function() {

    var ppk = "${pdata.ppk}";
    var pstate = $('#state option:selected').val();

    console.log(pstate);

    $.ajax({
        type: "post",
        url: "stateUrl.jsp",
        data : {"ppk": ppk, "pstate":pstate},
        dataType: "json",
        success: function(result){

            console.log("result: "+result);

            if(result == 1){
                alert("판매 완료!");
            }else {
                alert("판매 대기 전환");
            }
        },
        error: function(){
            console.log('에러 발생');
            console.log(err.status+" | "+err.errText);
        }
    });
});
</script>
```

View

Ajax

- 휴대폰 인증과 이메일 인증

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<jsp:useBean id="key" class="Logic.Randomkey" />
<jsp:useBean id="sns" class="Logic.Certified_SNS" />
<%
    System.out.println("SMS Url 접속 완료");

    String result = key.randomkey();
    sns.certified_sns(request.getParameter("phNum"), result);

    System.out.println("인증코드: "+result+" 전송 완료");

    out.write(result);

    System.out.println("success");
%>
```

```
<script type="text/javascript">
    var keyCode = null;
    $("#send-sms").on("click", function() {

        var phNum = $("#handphone").val();

        $.ajax({
            type : "post",
            url : "sendSMSUrl.jsp",
            data : {
                "phNum" : phNum
            },
            dataType : "text",
            success : function(result) {

                keyCode = result.substr(result.length-5,5);
                console.log("문자 전송 완료" + keyCode);
                alert("인증코드를 보냈습니다. 문자를 확인해주세요!");

            },
            error : function() {
                console.log('에러 발생');
                console.log(err.status + " | " + err.errText);
            }
        });
    });

    $("#phonecheck").on("focusout", function() {

        var input = $(this).val();

        if($("#handphone").attr("readonly") == "readonly"){
            $("#pcError").text("이메일 인증을 완료하였습니다.");
        }
        else{

            if ($("#phonecheck").val() == "") {
                $("#pcError").text("인증 필수");
            }
            else{
                if (input != keyCode) {
                    $("#pcError").text("인증번호 틀림지 재인증 요망");
                    $("#phonecheck").val("");
                }
                else {
                    $("#pcError").text("인증 성공");
                    $("#pcError").css("color", "green");
                    $("#phonecheck").attr("readonly", "readonly");
                    $("#handphone").attr("readonly", "readonly");
                }
            }
        }
    });
}</script>
```

Detailed-Address

상세주소를 입력하세요.

Phone number

(-) 없이 번호를 입력하세요.

문자 인증

SMS 인증코드를 입력하세요.

E-mail

이메일을 입력하세요.

이메일 인증

이메일 인증코드를 입력하세요.

필수동의

☐ 개인정보처리방침

View

유효성 검사

- 필수정보입력
- 아이디 닉네임 중복검사
- 비밀번호 유효성 검사

Name

실명을 입력하세요.

필수 정보입니다.

Nickname

닉네임을 입력하세요.

사용 불가능한 닉네임입니다.

ID

아이디를 입력하세요.

사용 불가능한 아이디입니다.

Password

비밀번호를 입력하세요.

8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.

Password verification

비밀번호를 재확인하세요.

```
//값을 입력하지 않았을때 alert창
if (name.value == "") {
    alert("이름을 입력하세요.");
    name.focus();
    return false; //아래코드부터 진행되지 않도록
}

var pattern_num = /[0-9]/;
var pattern_spc = /[~!@#$%^&*()_+|<>?:{}]/;

if (name.value.indexOf(" ") >= 0 || pattern_spc.test(name.value)
    || pattern_num.test(name.value)) {
    alert("한글과 영문 대 소문자를 사용하세요. (특수기호, 공백 사용 불가)");
    name.focus();
    return false; //아래코드부터 진행되지 않도록
}

if (nickname.value == "") {
    alert("닉네임을 입력하세요.");
    nickname.focus();
    return false;
}

if (uid.value == "") {
    alert("아이디를 입력하세요.");
    uid.focus();
    return false;
}

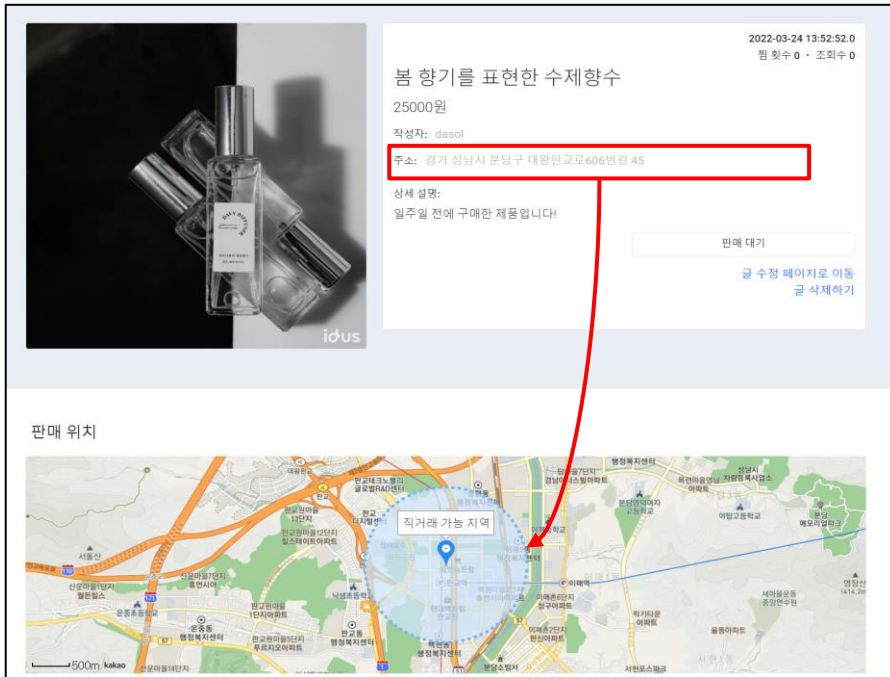
if (upw.value == "") {
    alert("비밀번호를 입력하세요.");
    upw.focus();
    return false;
}

var upwReg = /^(?=.*[a-zA-Z])(?=.*[!@#$%^*+=-])(?=.*[0-9]).{8,16}$/;

if (!upwReg.test(upw.value)) {
    alert("8~16자 영문 대 소문자, 숫자, 특수문자를 사용하세요.");
    upw.focus();
    return false;
}
```

View

MAP API



```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

var map = new kakao.maps.Map(mapContainer, mapOption); // 지도를 생성합니다

// 지도에 표시할 원을 생성합니다
var circle = new kakao.maps.Circle({
    center: new kakao.maps.LatLng(33.450701, 126.570667), // 원의 중심좌표입니다
    radius: 50, // 미터 단위의 원의 반지름입니다
    strokeWeight: 5, // 선의 두께입니다
    strokeColor: '#7588FA', // 선의 색깔입니다
    strokeOpacity: 1, // 선의 불투명도입니다 1에서 0 사이의 값이며 0에 가까울수록 투명합니다
    strokeStyle: 'dashed', // 선의 스타일입니다
    fillColor: '#CFE7FF', // 채우기 색깔입니다
    fillOpacity: 0.7 // 채우기 불투명도입니다
});

// 지도에 원을 표시합니다
circle.setMap(map);
```

```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

// 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);

// 주소-좌표 변환 객체를 생성합니다
var geocoder = new kakao.maps.services.Geocoder();

// 주소로 좌표를 검색합니다
geocoder.addressSearch('제주특별자치도 제주시 첨단로 242', function(result, status) {

    // 정상적으로 검색이 완료됐으면
    if (status === kakao.maps.services.Status.OK) {

        var coords = new kakao.maps.LatLng(result[0].y, result[0].x);

        // 결과값으로 받은 위치를 마커로 표시합니다
        var marker = new kakao.maps.Marker({
            map: map,
            position: coords
        });

        // 인포윈도우로 장소에 대한 설명을 표시합니다
        var infowindow = new kakao.maps.InfoWindow({
            content: '<div style="width:150px;text-align:center;padding:6px 0;">우리회사</div>'
        });
        infowindow.open(map, marker);

        // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
        map.setCenter(coords);
    }

});
```

Model

검색 필터링 (1)

- 게시물 제목 / 작성자 검색
- 최신순 / 조회순 정렬
 - : SearchCondition이 new값을 받으면 최신순 정렬
 - SearchCondition이 null값이나 다른 값을 받으면 조회순 정렬
- 4개의 SQL문을 분기처리를 사용하여
2개의 비즈니스 메서드로 처리

```
public ArrayList<ProVO> title_select(ProVO vo, int a){
    ArrayList<ProVO> datas = new ArrayList<ProVO>();
    ProVO data=null;
    conn=JDBCUtil.connect();
    try {

        if(vo.getSearchCondition()!=null && vo.getSearchCondition().equals("new")) {
            // null값일때 조회순 수행하지 않도록
            pstmt=conn.prepareStatement(title_select_ppk);
            pstmt.setString(1, vo.getPtitle());
            pstmt.setInt(2, vo.getPprice());
            pstmt.setInt(3, a);
        }
        else {//아무것도 설정하지 않았을때 default - 등록일
            pstmt=conn.prepareStatement(title_select_pcnt);
            pstmt.setString(1, vo.getPtitle());
            pstmt.setInt(2, vo.getPprice());
            pstmt.setInt(3, a);
        }

        ResultSet rs=pstmt.executeQuery();
        while(rs.next()) {
            data=new ProVO();
            data.setPpk(rs.getInt("ppk"));
            data.setPtitle(rs.getString("ptitle"));
            data.setPpic(rs.getString("ppic"));
            data.setPcontent(rs.getString("pcontent"));
            data.setPprice(rs.getInt("pprice"));
            data.setPmid(rs.getString("pmid"));
            data.setPdate(rs.getString("pdate"));
            data.setPcnt(rs.getInt("pcnt"));
            data.setPbkcnt(rs.getInt("pbkcnt"));
            datas.add(data);
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    JDBCUtil.disconnect(pstmt, conn);
    return datas;
}
```

```
//검색
static final String title_select_ppk="select * from product where ptitle like '%||?||%' and pprice BETWEEN ? AND ? order by ppk desc"; //title로 검색 and 금액으로 검색
static final String title_select_pcnt="select * from product where ptitle like '%||?||%' and pprice BETWEEN ? AND ? order by pcnt desc"; //title로 검색 and 금액으로 검색
static final String pmid_select_ppk="select * from product where pmid like '%||?||%' and pprice BETWEEN ? AND ? order by ppk desc"; //pmid로 검색 and 금액
static final String pmid_select_pcnt="select * from product where pmid like '%||?||%' and pprice BETWEEN ? AND ? order by pcnt desc"; //pmid로 검색 and 금액
```


Model

검색 필터링 (2)

- 게시물 제목 / 작성자 검색
- 최신순 / 조회순 정렬
 - : SearchCondition이 new값을 받으면 최신순 정렬
 - SearchCondition이 null값이나 다른 값을 받으면 조회순 정렬
- 위를 진행하기 위해서 VO에 DB에는 저장되지 않지만
조회순/최신순 값을 불러올 SearchCondition이라는 멤버변수 정의

```
public class ProVO {  
    private int ppk;  
    private String ptitle;  
    private String ppic;  
    private String pcontent;  
    private int pprice;  
    private String pmid;  
    private String pdate;  
    private int pcnt;  
    private int pbkcnt;  
    private int pstate;  
    private String searchCondition; // 검색 조건 > 선택에서만 사용
```

```
public ArrayList<ProVO> title_select(ProVO vo, int a){  
    ArrayList<ProVO> datas = new ArrayList<ProVO>();  
    ProVO data=null;  
    conn=JDBCUtil.connect();  
    try {  
  
        if(vo.getSearchCondition()!=null && vo.getSearchCondition().equals("new")) {  
            // null값일때 조회순 수행하지 않도록  
            pstmt=conn.prepareStatement(title_select_ppk);  
            pstmt.setString(1, vo.getPtitle());  
            pstmt.setInt(2, vo.getPprice());  
            pstmt.setInt(3, a);  
        }  
        else { //아무것도 설정하지 않았을때 default - 등록일  
            pstmt=conn.prepareStatement(title_select_pcnt);  
            pstmt.setString(1, vo.getPtitle());  
            pstmt.setInt(2, vo.getPprice());  
            pstmt.setInt(3, a);  
        }  
  
        ResultSet rs=pstmt.executeQuery();  
        while(rs.next()) {  
            data=new ProVO();  
            data.setPpk(rs.getInt("ppk"));  
            data.setPtitle(rs.getString("ptitle"));  
            data.setPpic(rs.getString("ppic"));  
            data.setPcontent(rs.getString("pcontent"));  
            data.setPprice(rs.getInt("pprice"));  
            data.setPmid(rs.getString("pmid"));  
            data.setPdate(rs.getString("pdate"));  
            data.setPcnt(rs.getInt("pcnt"));  
            data.setPbkcnt(rs.getInt("pbkcnt"));  
            datas.add(data);  
        }  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    JDBCUtil.disconnect(pstmt, conn);  
    return datas;  
}
```

Model

Join

- 이용자의 찜 전체목록을 불러오기 위해
찜목록에 담겨있는 제품 pk와, 제품 pk를 조인을 이용하여
제품의 정보를 가져옴

```
public ArrayList<ProVO> selectAll(BucketVO vo) {
    ArrayList<ProVO> datas = new ArrayList<ProVO>();
    conn=JDBCUtil.connect();
    try {
        pstmt=conn.prepareStatement(selectAll);
        pstmt.setString(1, vo.getMemberid());
        ResultSet rs=pstmt.executeQuery();
        while(rs.next()) {
            ProVO data=new ProVO();
            data.setPpk(rs.getInt("ppk"));
            data.setPtitle(rs.getString("ptitle"));
            data.setPpic(rs.getString("ppic"));
            data.setPcontent(rs.getString("pcontent"));
            data.setPprice(rs.getInt("pprice"));
            data.setPmid(rs.getString("pmid"));
            data.setPdate(rs.getString("pdate"));
            data.setPcnt(rs.getInt("pcnt"));
            data.setPbkcnt(rs.getInt("pbkcnt"));
            datas.add(data);
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    JDBCUtil.disconnect(pstmt, conn);
    return datas;
}
```

```
static final String selectAll="select product.* from product join bucket on product.ppk = bucket.productpk where memberid = ?";
```

Controller

메일 API

- 회원가입시 인증번호 발송
- 임시 비밀번호 발송
 - : Apache Commons Lang 3라이브러리 사용

```
import org.apache.commons.lang3.RandomStringUtils;

public class Randomkey {
    public String randomkey(){
        String key = RandomStringUtils.randomAlphanumeric(5);

        return key;
    }
}
```

```
String host = "smtp.gmail.com"; // 사용할 사이트
final String user = ""; // ID
final String password = ""; // PW
String to = mail; // 보낼 이메일 주소

// Get the session object
Properties props = new Properties();
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.smtp.host", host);
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.port", "587");
props.put("mail.smtp.ssl.protocols", "TLSv1.2");

Session session = Session.getDefaultInstance(props, new javax.mail.Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(user, password);
    }
});

try {
    MimeMessage message = new MimeMessage(session);
    message.setFrom(new InternetAddress(user));
    message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

    // Subject
    message.setSubject("웹보 중고마켓");

    // Text
    message.setText("안녕하세요, 웹보 중고마켓입니다.\r\n"
        + "인증번호는 "+key+"입니다.");
}
```

Controller

회원탈퇴

- 이용자의 제품의 찜했던 목록제거
- 이용자가 찜한 제품 찜 횟수 차감
- 이용자가 올린 제품제거

```
ArrayList<ProVO> data=new ArrayList<ProVO>();
//이용자의 제품의 찜했던 목록제거
pvo.setPmid(id);
data=pdao.pmid_select1(pvo);
for(ProVO pvoo:data) {
    bvo.setProductpk(pvoo.getPpk());
    bdao.deleteAll_productpk(bvo);
}
//이용자가 찜한 제품 찜횟수 차감
bvo.setMemberid(id);
datas=bdao.selectAll(bvo);

if(bdao.deleteAll(bvo)) {
    for(ProVO v:datas) {
        pdao.pbkcnt_update2(v);
    }
}else {
    throw new Exception("b");
}

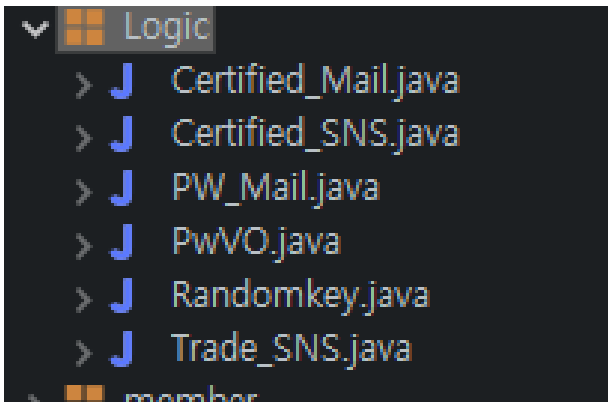
//이용자가 올린 제품제거
pdao.delete_pmid(pvo);

//회원삭제
mvo.setMid(id);
if( mdao.delete(mvo)) {
    session.invalidate();
    forward=new ActionForward();
    forward.setPath("main.do");
    forward.setRedirect(true);
}else {
    System.out.println("회원탈퇴에서 에러발생!");
    throw new Exception("e");
}
return forward;
}
```

Controller

모듈화

- 여러 군데에서 사용되는 코드를 메서드화 시켜 코드간소화
- 유지보수



```
import org.apache.commons.lang3.RandomStringUtils;

public class Randomkey {
    public String randomkey(){
        String key = RandomStringUtils.randomAlphanumeric(5);

        return key;
    }
}

public class Certified_SNS {
    public void certified_sns(String phone,String key) {

        String api_key = "";
        String api_secret = "|";
        Message coolsms = new Message(api_key, api_secret);

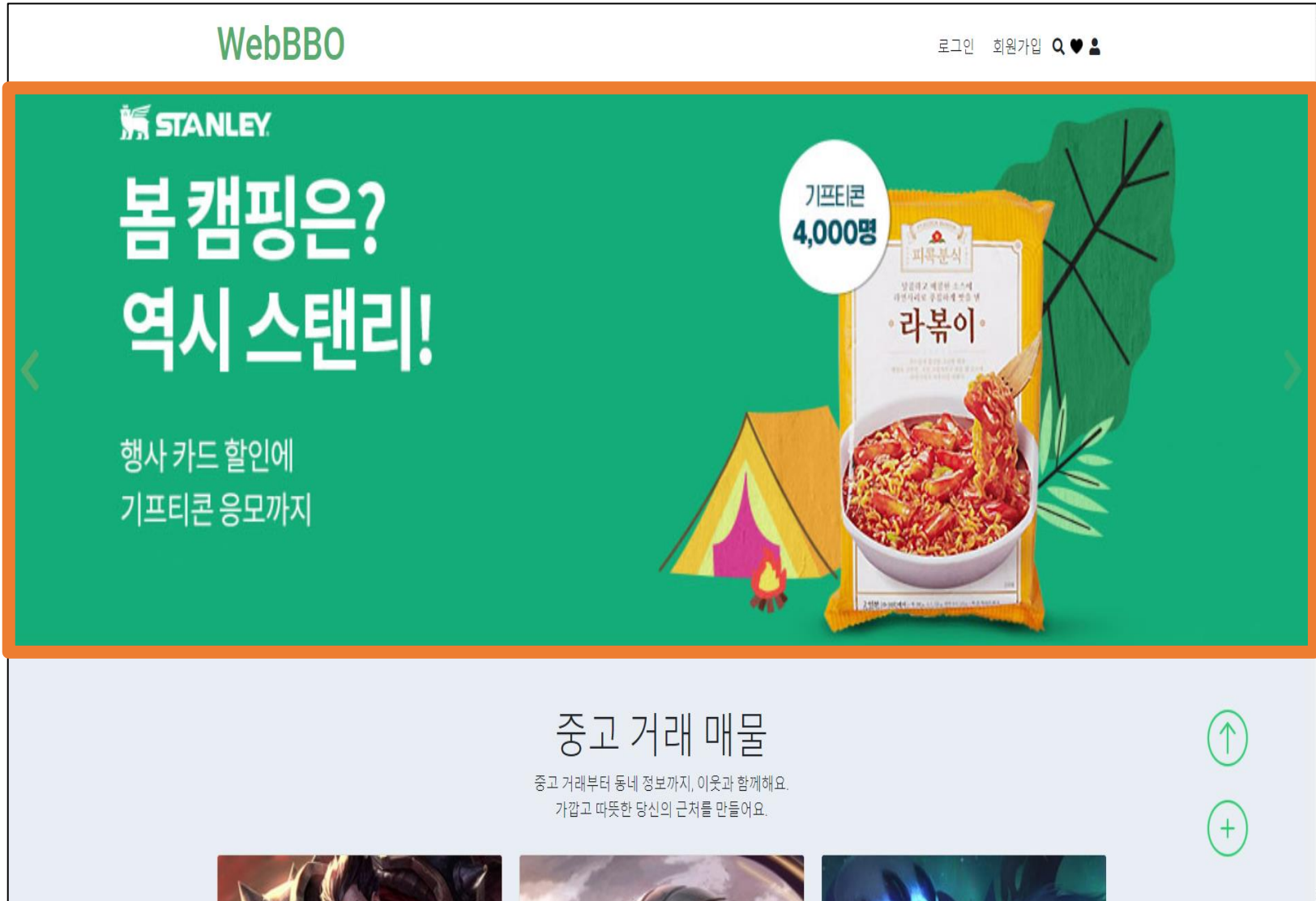
        // 4 params(to, from, type, text) are mandatory. must be filled
        HashMap<String, String> params = new HashMap<String, String>();
        params.put("to", phone);
        params.put("from", "01025556482");
        params.put("type", "SMS");
        params.put("text", "안녕하세요, 월배 중고마켓입니다.\r\n"
            + "인증번호는 "+key+"입니다.");
        params.put("app_version", "test app 1.2"); // application name and version

        try {
            coolsms.send(params);
            String();
            {
                message();
                code();
            }
        } catch (CoolSMSException e) {
            System.out.println(e.getMessage());
            System.out.println(e.getCode());
            return false;
        }
    }
}
```

담당 View 파트 핵심 기능 및 로직

1. 샘플 데이터로 슬라이드 배너 구현 (json 파일)
2. 좋아요 기능 (ajax)
3. 판매 상태 전환 (ajax)
4. 아이디 + 닉네임 중복검사 (ajax)
5. 문자 + 이메일 인증 (ajax)
6. 구현에 어려웠던 점
7. 에러사항 및 해결과정

샘플 데이터 (1/2)



메인 페이지 슬라이더 배너 구현

json 파일에 샘플 사진과
샘플 URL을 넣어 놓고 ajax로
샘플 데이터를 불러와서 구현

샘플 데이터 (2/2)

```
[{"src": "./assets/img/add1.png", "href": "https://store.emart.com/news/event/notice_event_view.do?master_cd=16727&trcknCode=mainbanner5"}, {"src": "./assets/img/add2.png", "href": "https://store.emart.com/news/event/notice_event_view.do?master_cd=16663&trcknCode=mainbanner8"}, {"src": "./assets/img/add3.png", "href": "https://store.emart.com/news/event/notice_event_view.do?master_cd=16746&trcknCode=mainbanner1"}]
```

```
<script type="text/javascript">
$.ajax({
  type : "post",
  url : "data.json",
  dataType : "json",
  success : function(data) {

    console.log('로드성공');
    console.log(data);

    var ele = "";

    $(".carousel-inner").append(ele);

    for(var i=0;i<data.length;i++){

      if(i==0){
        ele += "<div class='carousel-item active'>";
      } else{
        ele += "<div class='carousel-item'>";
      }
      ele += "<div class='order-lg-last'>";
      ele += "<a href='"+data[i].href+"'><img class='img-fluid' src='"+data[i].src+"' alt='제품사진'></a>";
      ele += "</div>";
      ele += "</div>";
    }

    $(".carousel-inner").append(ele);

  },
  error : function() {
    console.log('에러발생!');
    console.log(err.status + " | " + err.responseText);
  }
});
</script>
```

위 사진이 샘플 데이터를
넣어둔 json파일

아래 사진이 메인 페이지에 구현한
스크립트 코드

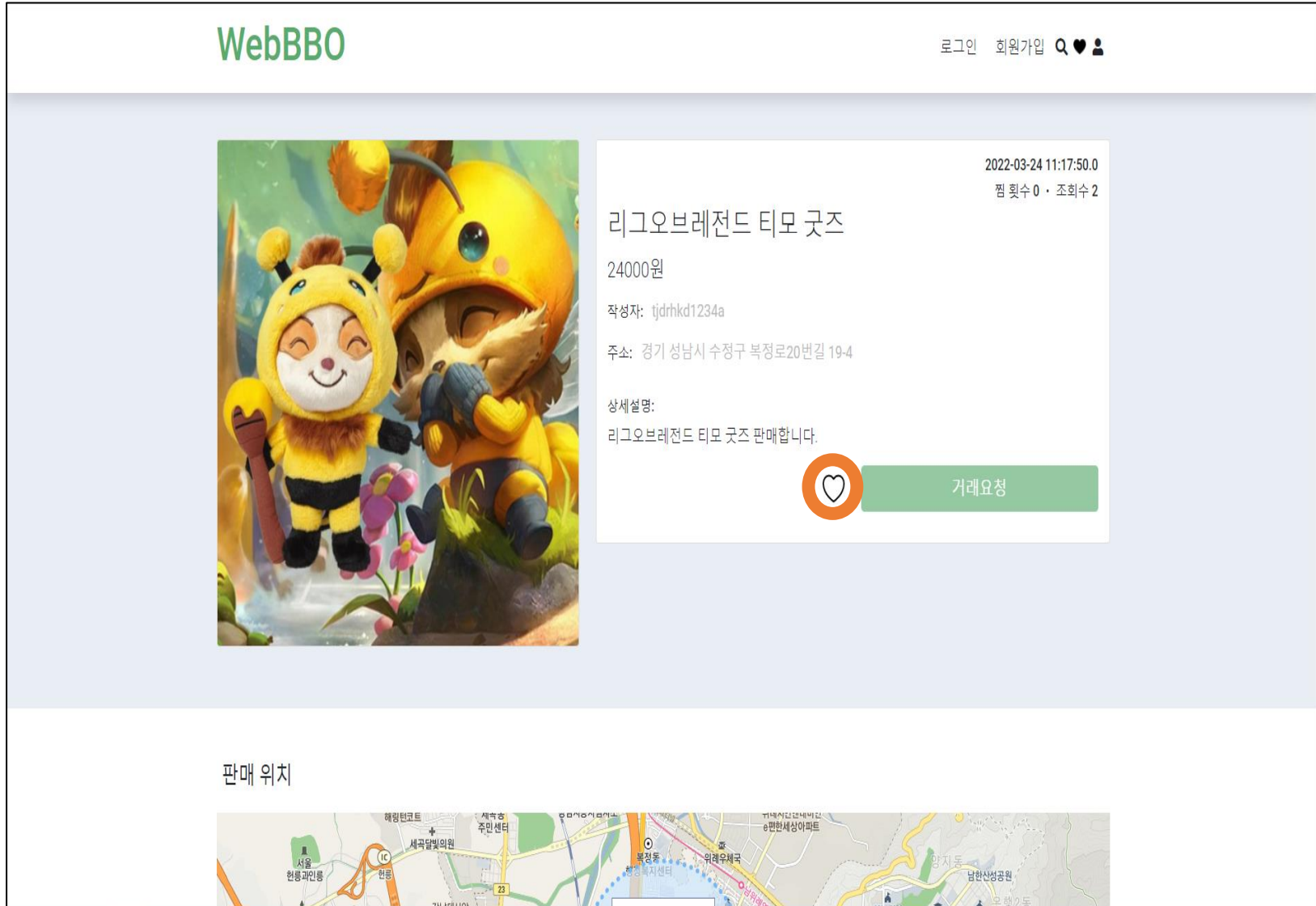
- JSON 구조 설명

1. JSON 데이터는 이름과 값의 쌍으로 이루어짐
2. JSON 데이터는 쉼표(,)로 나열
3. 객체(object)는 중괄호({})로 둘러싸아 표현
4. 배열(array)은 대괄호([])로 둘러싸아 표현

ajax에서 기재된 url로 보낼
데이터는 없고 url에서
json타입으로 데이터를 받아옴

받아온 json 데이터는 배열처럼
사용되어 for문으로 데이터를
추출하고 html 태그를 만들어
데이터를 넣고 반복되는 형태를
원하는 클래스에 삽입

좋아요 기능 (1/4)



찜 하기 버튼을 누르면 DB에
사용자가 찜을 했다는 것과
제품의 찜 횟수를 증가시키고
저장해야 됨

DB를 거치려면 컨트롤러를 거쳐야
되는데 페이지 이동이 발생됨

찜 버튼 클릭할 때마다
페이지 이동이 발생하는 문제를
해결하기 위해 View에서 ajax로 처리

좋아요 기능 (2/4)

```
<script type="text/javascript">
  $("#heartbtn").on("click", function() {

    if(${id} != null) {

      var productpk = "${pdata.ppk}";
      var memberid = "${id}";

      $.ajax({
        type : "post",
        url : "heartUrl.jsp",
        data : {
          "productpk" : productpk,
          "memberid" : memberid
        },
        dataType : "json",
        success : function(result) {

          console.log(result);

          if(${bkdata} == null){
            if (result == 1) {
              $("#heartbtn").attr('src', 'assets/img/heart (1).png');
              $("#cnt").text("찜 횟수 ${pdata.pbkcnt+1} · 조회수 ${pdata.pcnt}");
            } else {
              $("#heartbtn").attr('src', 'assets/img/heart.png');
              $("#cnt").text("찜 횟수 ${pdata.pbkcnt} · 조회수 ${pdata.pcnt}");
            }
          }
          else{
            if (result == 1) {
              $("#heartbtn").attr('src', 'assets/img/heart (1).png');
              $("#cnt").text("찜 횟수 ${pdata.pbkcnt} · 조회수 ${pdata.pcnt}");
            } else {
              $("#heartbtn").attr('src', 'assets/img/heart.png');
              $("#cnt").text("찜 횟수 ${pdata.pbkcnt-1} · 조회수 ${pdata.pcnt}");
            }
          }
        },
        error : function() {
          console.log('에러 발생');
          console.log(err.status + " | " + err.errText);
        }
      });
    }
    else{
      alert('로그인 후 이용 부탁드립니다!');
    }
  });
</script>
```

상세 페이지의 스크립트 코드

찜 하기 버튼은 사용자가 로그인 했을 때만
이용할 수 있도록 id 데이터가 있을 때
if문을 통과할 수 있도록 구현

제품의 pk와 id 데이터를 받아 ajax url에 기재된
heartUrl.jsp로 데이터들을 보냄

좋아요 기능 (3/4)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="bucket.*"%>
<jsp:useBean id="bdao" class="bucket.BucketDAO" />
<jsp:useBean id="bvo" class="bucket.BucketVO" />
<jsp:setProperty property="*" name="bvo"/>
<jsp:useBean id="pdao" class="product.ProDAO" />
<jsp:useBean id="pvo" class="product.ProVO" />
<jsp:setProperty property="ppk" name="pvo" param="productpk"/>
<%
    String result = null;

    System.out.println(bvo);
    System.out.println(pvo);

    BucketVO data = bdao.selectOne(bvo);
    if(data == null){
        if(bdao.insert(bvo)&&pdao.pbkcnt_update(pvo)){
            result = "1";
        }
    }
    else{
        if(bdao.deleteOne(bvo)&&pdao.pbkcnt_update2(pvo)){
            result = "0";
        }
    }

    out.write(result);
%>
```

VO와 DAO를 사용하기 위해
import, useBean을 사용

ajax에서 보내준 데이터를
VO에 자동으로 set 해주기 위해
setProperty 사용

사용자의 찜 여부를 확인하기 위해
selectOne 메서드를 사용

데이터가 없으면 찜을 안 한 상태

데이터가 있으면 찜을 한 상태

데이터가 없다면 DB에
insert 메서드를 사용해
찜한 상태로 만들어주고
update 메서드를 통해
찜 횟수를 추가하고
result 값을 1로 세팅

반대로 데이터가 있으면 DB에
delete 메서드를 사용해
찜 한 상태를 없애고
update 메서드를 통해
찜 횟수를 차감하고
result 값을 0으로 세팅

out.write()을 사용해 result 값을
ajax로 반환

좋아요 기능 (4/4)

```
<script type="text/javascript">
    $("#heartbtn").on("click", function() {

        if(${id} != null) {

            var productpk = "${pdata.ppk}";
            var memberid = "${id}";

            $.ajax({
                type : "post",
                url : "heartUrl.jsp",
                data : {
                    "productpk" : productpk,
                    "memberid" : memberid
                },
                dataType : "json",
                success : function(result) {

                    console.log(result);

                    if(${bkdata} == null){
                        if (result == 1) {
                            $("#heartbtn").attr('src', 'assets/img/heart (1).png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt+1} · 조회수 ${pdata.pcnt}");
                        } else {
                            $("#heartbtn").attr('src', 'assets/img/heart.png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt} · 조회수 ${pdata.pcnt}");
                        }
                    }
                    else{
                        if (result == 1) {
                            $("#heartbtn").attr('src', 'assets/img/heart (1).png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt} · 조회수 ${pdata.pcnt}");
                        } else {
                            $("#heartbtn").attr('src', 'assets/img/heart.png');
                            $("#cnt").text("찜 횟수 ${pdata.pbkcnt-1} · 조회수 ${pdata.pcnt}");
                        }
                    }
                },
                error : function() {
                    console.log('에러 발생');
                    console.log(err.status + " | " + err.errText);
                }
            });
        }
        else{
            alert('로그인 후 이용 부탁드립니다!');
        }
    });
</script>
```

success function 내에서

- 원래 사용자가 제품을
찜 한 상태가 아닐 때

반환된 result 값이 1일 때
(찜을 하는 경우) 찜 버튼 이미지를
변경하고 찜 횟수 1증가

result 값이 1이 아닐 때
(찜을 푸는 경우) 찜 버튼 이미지를
변경하고 원래 찜 횟수로 변경

- 원래 사용자가 제품을
찜 한 상태일 때


반환된 result 값이 1일 때
(찜을 하는 경우) 찜 버튼 이미지를
변경하고 원래 찜 횟수로 변경

result 값이 1이 아닐 때
(찜을 푸는 경우) 찜 버튼 이미지를
변경하고 찜 횟수 1 차감

판매 상태 전환 (1/4)

WebBBO

성광님 환영합니다. 로그인 🔍 ❤️ 👤



2022-03-24 11:19:17.0
찜 횟수 0 · 조회수 0

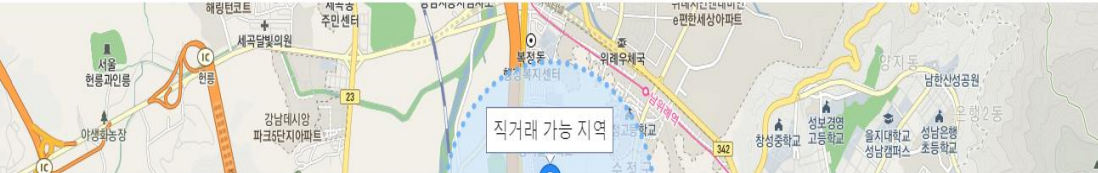
이 티모 모자
20000원
작성자: tjdrhkd1234a
주소: 경기 성남시 수정구 복정로20번길 19-4
상세 설명:
이 티모 모자 판매합니다.

판매 대기

판매 대기

판매 완료

판매 위치



판매자가 자신이 등록한
상품 페이지에 들어갔을 때의
페이지 모습

구매자가 판매자에게 거래요청을
하고 거래가 성사되면
판매자는 판매 상태를 판매 완료로
바꿀 수 있음

페이지에서 판매 상태 변경을
페이지 이동 없이 구현하고자
ajax를 사용

판매 상태 전환 (2/4)

```
<script type="text/javascript">
$("#state").on("change", function() {

    var ppk = "${pdata.ppk}";
    var pstate = $('#state option:selected').val();

    console.log(pstate);

    $.ajax({
        type: "post",
        url: "stateUrl.jsp",
        data : {"ppk": ppk, "pstate":pstate},
        dataType: "json",
        success: function(result){

            console.log("result: "+result);

            if(result == 1){
                alert("판매 완료!");
            }else {
                alert("판매 대기 전환");
            }
        },
        error: function(){
            console.log('에러 발생');
            console.log(err.status+" | "+err.errText);
        }
    });
});
</script>
```

판매 상태를 나타내는 select 박스에 change 이벤트가 발생하면 제품의 pk와 현재 선택한 select 박스의 value를 받아오고 ajax url에 기재된 stateUrl.jsp로 pk 데이터와 value 데이터를 전달

판매 상태 전환 (3/4)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="product.*"%>
<jsp:useBean id="pdao" class="product.ProDAO" />
<jsp:useBean id="pvo" class="product.ProVO" />
<jsp:setProperty property="*" name="pvo"/>
<%
    if(pdao.update_state(pvo)){
        System.out.println("stateUrl 로그: 업데이트 완료");
    }

    String result = request.getParameter("pstate");

    out.write(result);
%>
```

stateUrl.jsp에서 필요한 VO와 DAO를 사용하기 위해 import, useBean을 쓰고 전달된 데이터를 VO에 자동으로 set 해주기 위해 setProperty를 사용

받은 제품의 pk와 제품의 판매 상태를 update 메서드를 통해 DB에 저장

result에 pstate(판매 상태) 값을 주고 out.write()로 result 값을 ajax로 반환

판매 상태 전환 (4/4)

```
<script type="text/javascript">
$("#state").on("change", function() {

    var ppk = "${pdata.ppk}";
    var pstate = $('#state option:selected').val();

    console.log(pstate);

    $.ajax({
        type: "post",
        url: "stateUrl.jsp",
        data : {"ppk": ppk, "pstate":pstate},
        dataType: "json",
        success: function(result){

            console.log("result: "+result);

            if(result == 1){
                alert("판매 완료!");
            }else {
                alert("판매 대기 전환");
            }
        },
        error: function(){
            console.log('에러 발생');
            console.log(err.status+" | "+err.errText);
        }
    });
});
</script>
```

success function을 보면

받은 result 값이 1이라면 (판매 완료 상태)
판매 완료라는 알람 창을 띄움

받은 result 값이 1이 아니라면 (판매 대기 상태)
판매 대기 전환이라는 알람 창을 띄움

닉네임 + 아이디 중복검사 (1/4)

WebBBO

로그인 회원가입 🔍 ❤️ 👤

SignIn

WebBBO 중고거래에 오신 것을 환영합니다!
----- 회원가입 -----

Name

실명을 입력하세요.

Nickname

닉네임을 입력하세요.

ID

아이디를 입력하세요.

Password

비밀번호를 입력하세요.

Password verification

비밀번호를 재확인하세요.

주소 검색

Address

주소 검색을 이용해주세요.

Detailed-Address

상세주소를 입력하세요.

Phone number

(-) 없이 번호를 입력하세요.

문자 인증

SMS 인증코드를 입력하세요.

E-mail

이메일을 입력하세요.

이메일 인증

이메일 인증코드를 입력하세요.

필수동의

☐ 개인정보처리방침

☐ 서비스약관

가입하기

Copyright © 2022 Team WebBBo | Designed by WebBBo

회원가입 페이지에서 닉네임과 아이디 중복 검사를 하려면 DB에 입력한 값과 같은 닉네임과 아이디가 있는지 확인을 해야함

컨트롤러를 돌려 DB를 확인하게 되면 페이지를 넘어가야 되는 문제가 생김

닉네임 아이디 중복검사를 ajax로 구현

닉네임 + 아이디 중복검사 (2/4)

```
<script type="text/javascript">
    $("#nickname").on("focusout", function() {

        var mnick = $(this).val();

        $.ajax({
            type : "post",
            url : "nickUrl.jsp",
            data : {
                "mnick" : mnick
            },
            dataType : "json",
            success : function(result) {

                console.log(result);

                if($("#nickname").val() == ""){
                    $("#nickError").text("필수 정보입니다.");
                } else{
                    if (result == "0") {
                        $("#nickError").text("사용 가능한 닉네임입니다.");
                        $("#nickError").css("color", "green");
                    } else {
                        $("#nickError").text("사용 불가능한 닉네임입니다.");
                        $("#nickError").css("color", "red");
                        $("#nickname").val('');
                    }
                }
            },
            error : function() {
                console.log('에러 발생');
                console.log(err.status + " | " + err.responseText);
            }
        });
    });
</script>
```

닉네임 중복검사용 스크립트

focusout 이벤트 발생 시 입력한 닉네임을 불러와서
url에 기재된 nickUrl.jsp로 닉네임 데이터 전달

닉네임 + 아이디 중복검사 (3/4)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="member.*"%>
<jsp:useBean id="mdao" class="member.MemDAO" />
<jsp:useBean id="mvo" class="member.MemVO" />
<jsp:setProperty property="*" name="mvo" />
<%
    String result = null;

    MemVO data=mdao.selectmnick(mvo);
    System.out.println("DAO결과");
    if(data == null){
        result = "0"; // 닉네임 만들기 가능
    }else{
        result = "1"; // 닉네임 이미 있음
    }
    System.out.println(result);
    out.write(result);
%>
```

VO와 DAO를 사용하기 위해
import, useBean 사용

받은 닉네임 데이터를
자동으로 VO에 set 하기 위해
setProperty 사용

DB에 selectOne 메서드를 사용해
존재하는 닉네임인지 검사

메서드를 통해 반환된 데이터가
null이면 닉네임을 만들 수 있는
상태이고 result를 0으로 세팅

메서드를 통해 반환된 데이터가
null이 아니면 닉네임을 만들 수
없는 상태이고 result를 1로 세팅

out.write()로 result 값을
ajax로 반환

닉네임 + 아이디 중복검사 (4/4)

```
<script type="text/javascript">
    $("#nickname").on("focusout", function() {

        var mnick = $(this).val();

        $.ajax({
            type : "post",
            url : "nickUrl.jsp",
            data : {
                "mnick" : mnick
            },
            dataType : "json",
            success : function(result) {

                console.log(result);

                if($("#nickname").val() == ""){
                    $("#nickError").text("필수 정보입니다.");
                } else{
                    if (result == "0") {
                        $("#nickError").text("사용 가능한 닉네임입니다.");
                        $("#nickError").css("color", "green");
                    } else {
                        $("#nickError").text("사용 불가능한 닉네임입니다.");
                        $("#nickError").css("color", "red");
                        $("#nickname").val('');
                    }
                }
            },
            error : function() {
                console.log('에러 발생');
                console.log(err.status + " | " + err.responseText);
            }
        });
    });
</script>
```

success function 내에서

닉네임 입력을 아무것도 하지
않았을 때 필수 정보라는
텍스트를 띄워주고

닉네임 입력을 한 상태라면

반환된 result 값이 0일 때
(사용가능한 닉네임일 때)
사용가능한 닉네임이라는
텍스트가 뜨고

반환된 result 값이 1일 때
(이미 있는 닉네임일 때)
사용 불가능한 닉네임이라는
텍스트가 뜨고 입력한 값을 초기화

아이디 중복검사 위 과정과 동일

문자 + 이메일 인증 (1/4)

WebBBO

로그인 회원가입 🔍 ❤️ 👤

SignIn

WebBBO 중고거래에 오신 것을 환영합니다!
----- 회원가입 -----

Name

실명을 입력하세요.

Nickname

닉네임을 입력하세요.

ID

아이디를 입력하세요.

Password

비밀번호를 입력하세요.

Password verification

비밀번호를 재확인하세요.

주소 검색

Address

주소 검색을 이용해주세요.

Detailed-Address

상세주소를 입력하세요.

Phone number

(-) 없이 번호를 입력하세요.

문자 인증

SMS 인증코드를 입력하세요.

E-mail

이메일을 입력하세요.

이메일 인증

이메일 인증코드를 입력하세요.

필수동의

☐ 개인정보처리방침

☐ 서비스약관

가입하기

Copyright © 2022 Team WebBBo | Designed by WebBBo

회원가입 페이지에서 페이지 이동 없이
난수 코드를 생성하는 로직과 문자와 이메일을
전송하는 로직을 사용하기 위해 ajax로 구현

문자 + 이메일 인증 (2/4)

```
<script type="text/javascript">
var keyCode = null;
$("#send-sms").on("click", function() {

    var phNum = $("#handphone").val();

    $.ajax({
        type : "post",
        url : "sendSMSUrl.jsp",
        data : {
            "phNum" : phNum
        },
        dataType : "text",
        success : function(result) {

            keyCode = result.substr(result.length-5,5);
            console.log("문자 전송 완료" + keyCode);
            alert("인증코드를 보냈습니다. 문자를 확인해주세요!");

        },
        error : function() {
            console.log('에러 발생');
            console.log(err.status + " | " + err.responseText);
        }
    });
});

$("#phonecheck").on("focusout", function() {

    var input = $(this).val();

    if($("#handphone").attr("readonly") == "readonly"){
        $("#pcError").text("이미 인증을 완료하였습니다.");
    }
    else{

        if ($("#phonecheck").val() == "") {
            $("#pcError").text("인증 필수");
        }
        else{
            if (input != keyCode) {
                $("#pcError").text("인증번호 틀릴지 재인증 요망");
                $("#phonecheck").val('');
            }
            else {
                $("#pcError").text("인증 성공");
                $("#pcError").css("color", "green");
                $("#phonecheck").attr("readonly", "readonly");
                $("#handphone").attr("readonly", "readonly");
            }
        }
    }

}

});
</script>
```

인증 문자 전송용 스크립트

문자 인증 버튼을 클릭하게 되면 입력한 번호를 받아오고
url에 기재된 sendSMSUrl.jsp로 번호 데이터를 전달

문자 + 이메일 인증 (3/4)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<jsp:useBean id="key" class="Logic.Randomkey" />
<jsp:useBean id="sns" class="Logic.Certified_SNS" />
<%
    System.out.println("SMS Url 접속 완료");

    String result = key.randomkey();
    sns.certified_sns(request.getParameter("phNum"), result);

    System.out.println("인증코드: "+result+" 전송 완료");

    out.write(result);

    System.out.println("success");
%>
```

난수 코드를 생성하는 메서드와 문자를 전송하는 메서드를 사용하기 위해 useBean을 사용

result에 난수 코드를 넣어주고 받은 번호 데이터와 result를 문자 전송 메서드의 인풋으로 넣어 실행

out.write()를 사용해 난수 코드를 넣은 result를 ajax로 반환

문자 + 이메일 인증 (4/4)

```
<script type="text/javascript">
var keyCode = null;
$("#send-sms").on("click", function() {

    var phNum = $("#handphone").val();

    $.ajax({
        type : "post",
        url : "sendSMSUrl.jsp",
        data : {
            "phNum" : phNum
        },
        dataType : "text",
        success : function(result) {

            keyCode = result.substr(result.length-5,5);
            console.log("문자 전송 완료" + keyCode);
            alert("인증코드를 보냈습니다. 문자를 확인해주세요!");

        },
        error : function() {
            console.log('에러 발생');
            console.log(err.status + " | " + err.responseText);
        }
    });
});

$("#phonecheck").on("focusout", function() {

    var input = $(this).val();

    if($("#handphone").attr("readonly") == "readonly"){
        $("#pcError").text("이미 인증을 완료하였습니다.");
    }
    else{

        if ($("#phonecheck").val() == "") {
            $("#pcError").text("인증 필수");
        }
        else{
            if (input != keyCode) {
                $("#pcError").text("인증번호 불일치 재인증 요망");
                $("#phonecheck").val('');
            }
            else {
                $("#pcError").text("인증 성공");
                $("#pcError").css("color", "green");
                $("#phonecheck").attr("readonly", "readonly");
                $("#handphone").attr("readonly", "readonly");
            }
        }
    }

});
</script>
```

받는 데이터 타입을 text로 설정하고 받은 난수 코드의 빈 공백을 제거하기 위해 substr을 사용해 받아온 데이터의 길이-5를 한 지점부터 5개만 잘라서 사용

이후 인증코드를 보냈다는 알람 창을 띄우고 인증 코드를 적는 입력란이 focusout 되면 인증 코드를 입력한 값을 받아와서 입력한 값이 인증 코드와 다르면 입력 값을 초기화 시키고 입력한 값이 인증 코드와 같다면 readonly 속성을 추가해 더 이상의 입력을 방지

이메일 인증 코드 전송용 ajax도 위의 과정과 동일

구현에 어려웠던 점

WebBBO

로그인 회원가입 Q ♥

STANLEY

봄 캠핑은?
역시 스탠리!

행사 카드 할인에
기프트콘 응모까지

기프트콘
4,000명

```
<script type="text/javascript">$.ajax({  type : "post",  url : "data.json",  dataType : "json",  success : function(data) {    console.log('프트성공');    console.log(data);    var ele = "";    $(".carousel-inner").append(ele);    for(var i=0;i<data.length;i++){      if(i==0){        ele += "<div class='carousel-item active'>";      } else{        ele += "<div class='carousel-item'>";      }      ele += "<div class='order-lg-last'>";      ele += "<a href='"+data[i].href+"'><img class='img-fluid' src='"+data[i].src+"' alt='제품사진'></a>";      ele += "</div>";      ele += "</div>";    }    $(".carousel-inner").append(ele);  },  error : function() {    console.log('에러발생!');    console.log(err.status + " | " + err.errText);  }  });</script>
```

반복되는 html 태그를 만들어 줄 때
forEach문을 사용하지 않고
단순 for문을 사용한 이유

페이지를 분석한 결과
메인 페이지의 슬라이드 배너가
동작하려면 첫번째 div 속성에
active 클래스가 존재해야 함

첫번째 div에 active 클래스를
넣지 않고 만든다거나 모든 div에
active클래스를 넣으면
슬라이드 배너가 나타나지 않음

따라서 첫번째 div에만
active 클래스를 주기 위해
단순 for문을 사용

에러사항 및 해결과정 (1/2)

WebBBO 로그인 회원가입

SignIn

WebBBO 광고거래에 오신 것을 환영합니다
— 회원가입 —

Name
실명을 입력하세요.

Nickname
닉네임을 입력하세요.

ID
아이디를 입력하세요.

Password
비밀번호를 입력하세요.

Password verification
비밀번호를 재확인하세요.

주소 검색

Address
주소 검색을 이용해주세요.

Detailed-Address
상세주소를 입력하세요.

Phone number
(-) 없이 번호를 입력하세요.

문자 인증

SMS 인증코드를 입력하세요.

E-mail
이메일을 입력하세요.

이메일 인증

이메일 인증코드를 입력하세요.

로그인
약관동의
약관정보보러가기
서비스약관

가입하기

Copyright © 2022 Team WebBBo | Designed by WebBBo

```
<script type="text/javascript">
var keyCode = null;
$("#send-sms").on("click", function() {

var phNum = $("#handphone").val();

$.ajax({
type : "post",
url : "sendSMSUrl.jsp",
data : {
"phNum" : phNum
},
dataType : "text",
success : function(result) {

keyCode = result.substr(result.length-5,5);
console.log("문자 전송 완료" + keyCode);
alert("인증코드를 보냈습니다. 문자를 확인해주세요!");

},
error : function() {
console.log('에러 발생');
console.log(err.status + " | " + err.responseText);
}
});
});

$("#phonecheck").on("focusout", function() {

var input = $(this).val();

if($("#handphone").attr("readonly") == "readonly"){
$("#pcError").text("이미 인증을 완료하였습니다.");
}
else{

if ($("#phonecheck").val() == "") {
$("#pcError").text("인증 필수");
} else{
if (input != keyCode) {
$("#pcError").text("인증번호 틀림지 재인증 요망");
$("#phonecheck").val("");
} else {
$("#pcError").text("인증 성공");
$("#pcError").css("color", "green");
$("#phonecheck").attr("readonly", "readonly");
$("#handphone").attr("readonly", "readonly");
}
}
}

});
</script>
```

ajax로 받는 데이터 타입을 text로 설정한 이유

에러사항 및 해결과정 (2/2)

```
<script type="text/javascript">
var keyCode = null;
$("#send-sms").on("click", function() {

    var phNum = $("#handphone").val();

    $.ajax({
        type : "post",
        url : "sendSMSUrl.jsp",
        data : {
            "phNum" : phNum
        },
        dataType : "text",
        success : function(result) {

            keyCode = result.substr(result.length-5,5);
            console.log("문자 전송 완료" + keyCode);
            alert("인증코드를 보냈습니다. 문자를 확인해주세요!");

        },
        error : function() {
            console.log('에러 발생');
            console.log(err.status + " | " + err.errText);
        }
    });
});

$("#phonecheck").on("focusout", function() {

    var input = $(this).val();

    if($("#handphone").attr("readonly") == "readonly"){
        $("#pcError").text("이미 인증을 완료하였습니다.");
    }
    else{
        if ($("#phonecheck").val() == "") {
            $("#pcError").text("인증 필수");
        }
        else{
            if (input != keyCode) {
                $("#pcError").text("인증번호 불일치 재인증 요망");
                $("#phonecheck").val('');
            }
            else {
                $("#pcError").text("인증 성공");
                $("#pcError").css("color", "green");
                $("#phonecheck").attr("readonly", "readonly");
                $("#handphone").attr("readonly", "readonly");
            }
        }
    }

});
</script>
```

```
code:200
message:

SXPSK

error:SyntaxError: Unexpected token S in JSON at
position 6
```

```
result:
cYzPB
```

ajax로 받는 인증 코드
데이터 타입을 text로 설정한 이유

json 타입으로 인증 코드 데이터를
받을 시 다음과 같은 200 에러 발생

JSON 데이터는 엄격한 방식으로
구문 분석됨

잘못된 JSON이 거부되고
구문 분석 오류가 발생

인증코드가 json 데이터로 반환되지
못해 인증 코드와 같은
String 타입의 text 타입으로 설정

text 타입으로 설정 후
데이터를 받게 되면
다음과 같은 공백이 섞인
결과값을 반환 받게 됨

공백 없이 인증 코드만
추출하기 위해 substr을 사용

Map Api 구현 과정 (1/3)

```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

// 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);

// 주소-좌표 변환 객체를 생성합니다
var geocoder = new kakao.maps.services.Geocoder();

// 주소로 좌표를 검색합니다
geocoder.addressSearch('제주특별자치도 제주시 칠단로 242', function(result, status) {

    // 정상적으로 검색이 완료됐으면
    if (status === kakao.maps.services.Status.OK) {

        var coords = new kakao.maps.LatLng(result[0].y, result[0].x);

        // 결과값으로 받은 위치를 마커로 표시합니다
        var marker = new kakao.maps.Marker({
            map: map,
            position: coords
        });

        // 인포윈도우로 장소에 대한 설명을 표시합니다
        var infowindow = new kakao.maps.InfoWindow({
            content: '<div style="width:150px;text-align:center;padding:6px 0;">우리회사</div>'
        });
        infowindow.open(map, marker);

        // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
        map.setCenter(coords);
    }

});
```

```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

var map = new kakao.maps.Map(mapContainer, mapOption); // 지도를 생성합니다

// 지도에 표시할 원을 생성합니다
var circle = new kakao.maps.Circle({
    center : new kakao.maps.LatLng(33.450701, 126.570667), // 원의 중심좌표입니다
    radius: 50, // 미터 단위의 원의 반지름입니다
    strokeWeight: 5, // 선의 두께입니다
    strokeColor: '#75B8FA', // 선의 색깔입니다
    strokeOpacity: 1, // 선의 불투명도입니다 1에서 0 사이의 값이며 0에 가까울수록 투명합니다
    strokeStyle: 'dashed', // 선의 스타일입니다
    fillColor: '#CFE7FF', // 채우기 색깔입니다
    fillOpacity: 0.7 // 채우기 불투명도입니다
});

// 지도에 원을 표시합니다
circle.setMap(map);
```

Map Api를 다룰 때
목적이 원하는 장소에서
원하는 반경(범위)을 표시하는 것

카카오 Map Api 샘플에서
[주소로 장소 표시하기]와
[원 표시하기]를 합쳐서 구현

왼쪽 코드 [주소로 장소 표시하기]

오른쪽 코드 [원 표시하기]

Map Api 구현 과정 (2/3)

```
<script>
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
mapOption = {
  center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
  level: 6 // 지도의 확대 레벨
};

// 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);

// 주소-좌표 변환 객체를 생성합니다
var geocoder = new kakao.maps.services.Geocoder();

// 주소로 좌표를 검색합니다
geocoder.addressSearch('경기도 성남시 수정구 복정로 20번길 19-4', function(result, status) {

  var coords;
  // 정상적으로 검색이 완료됐으면
  if (status === kakao.maps.services.Status.OK) {

    coords = new kakao.maps.LatLng(result[0].y, result[0].x)

    // 결과값으로 받은 위치를 마커로 표시합니다
    var marker = new kakao.maps.Marker({
      map: map,
      position: coords
    });

    // 인포윈도우로 장소에 대한 설명을 표시합니다
    var infowindow = new kakao.maps.InfoWindow({
      content: '<div style="width:150px;text-align:center;padding:6px 0;">거레 반경 주변 1km</div>'
    });
    infowindow.open(map, marker);

    // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
    map.setCenter(coords);
  }

  // 지도에 표시할 원을 생성합니다
  var circle = new kakao.maps.Circle({
    center : coords, // 원의 중심좌표입니다
    radius: 1000, // 미터 단위의 원의 반지름입니다
    strokeWeight: 5, // 선의 두께입니다
    strokeColor: '#75B8FA', // 선의 색깔입니다
    strokeOpacity: 1, // 선의 불투명도 입니다 1에서 0 사이의 값이며 0에 가까울수록 투명합니다
    strokeStyle: 'dashed', // 선의 스타일 입니다
    fillColor: '#CFE7FF', // 채우기 색깔입니다
    fillOpacity: 0.7 // 채우기 불투명도 입니다
  });

  // 지도에 원을 표시합니다
  circle.setMap(map);
});
</script>
```

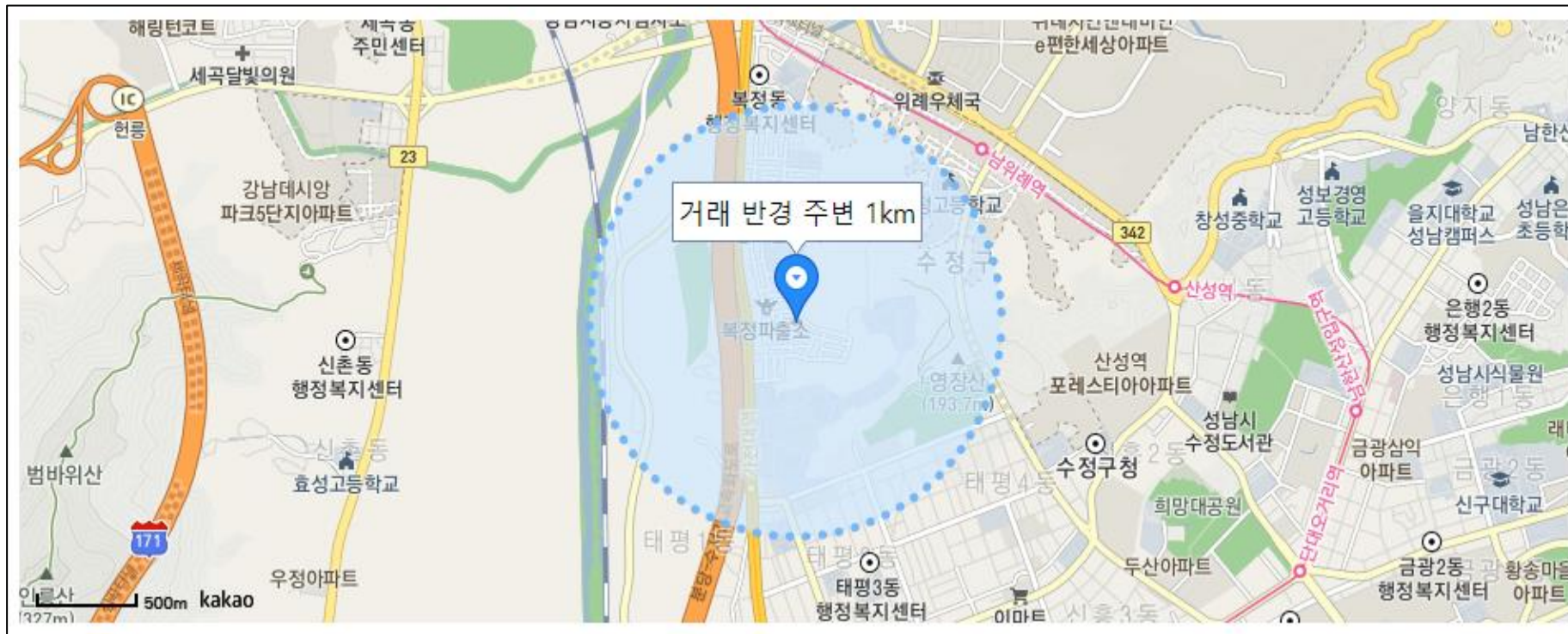
코드 합치는 과정

겹치는 코드들은 하나만 사용

[주소로 장소 표시하기] 샘플에서 위치에 해당하는 coords를
[원 표시하기] 샘플 코드에서도 사용하기 위해 if문 밖에
전역 변수로 세팅하고 [원 표시하기] 샘플에서 중심에
해당하는 center에 coords 삽입

반경을 1km(1000m)로 세팅하기 위해 radius를 1000으로 설정

Map Api 구현 과정 (3/3)



구현한 코드의 결과

Mail Api 구현 과정 (1/6)

```
package mail;

import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class SendEmail {
    public static void main(String[] args) {

        String host = "smtp.gmail or naver.com"; // 사용할 사이트
        final String user = "ID@gmail or naver.com"; // ID
        final String password = "PW"; // PW

        String to = "mailAddress"; // 보낼 이메일 주소

        // Get the session object
        Properties props = new Properties();
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", host);
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.port", "587");

        Session session = Session.getDefaultInstance(props, new javax.mail.Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(user, password);
            }
        });

        // Compose the message
        try {
            MimeMessage message = new MimeMessage(session);
            message.setFrom(new InternetAddress(user));
            message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

            // Subject
            message.setSubject("이메일 제목");

            // Text
            message.setText("본문 내용");

            // send the message
            Transport.send(message);
            System.out.println("이메일 전송 성공!");

        } catch (MessagingException e) {
            e.printStackTrace();
        }
    }
}
```

팀 프로젝트에서 구현하고 싶은 회원가입 시 이메일 인증, 비밀번호 찾기 시 임시 비밀번호가 담긴 메일 전송을 하기 위해 Mail Api를 사용해 구현

왼쪽 코드는 메일 api 사용을 위해 구글링한 코드

Mail Api 구현 과정 (2/6)

메일쓰기

내계쓰기

★

📎

TO

안읽음

중요

첨부

받는사람

전체메일

정리하기

받은메일함 23

수신확인

보낸메일함

수신확인

임시보관함

비밀첩부

내계쓰메일함

비밀첩부

VIP

중요연락처 지정하기

내 메일함

1. ⚙️

뉴스레터함

쇼핑레터함

청구서함

스마트메일함

프로모션

청구결제

SMS

사용하기

사용하기

사용하기

환경 설정 | 메일로 돌아가기

기본 환경 설정

메일함 관리

메일 자동 분류

서명/빠른답장

부재 중 설정

새 메일 알림 설정

스팸 설정

외부 메일 가져오기

2. ✓ POP3/IMAP 설정

단축키

POP3/SMTP 설정

3. IMAP/SMTP 설정

스마트폰에서 네이버 메일을 더욱 잘 관리하고 쓸 수 있도록 IMAP/SMTP를 설정합니다.

gunman511님은 현재 IMAP/SMTP를 사용하고 있습니다.

IMAP/SMTP 사용

4. ☒ 사용함 ☐ 사용 안 함?

IMAP 동기화 메일 제한

1,000통

IMAP으로 동기화할 수 있는 최대 메일 통수를 지정할 수 있습니다.
너무 많은 메일을 동기화하는 경우 데이터 과금을 주의해 주세요.

기본 설정으로

확인

취소

스마트폰 메일 애플리케이션 계정 설정

스마트폰의 메일 계정 설정에 아래와 같이 등록해 주세요.

IMAP 서버명 : imap.naver.com

SMTP 서버명 : smtp.naver.com

IMAP 포트 : 993, 보안연결(SSL) 필요

SMTP 포트 : 587, 보안 연결(TLS) 필요
(TLS가 없는 경우 SSL로 연결)

아이디 :

비밀번호 : 네이버 로그인 비밀번호

TIP

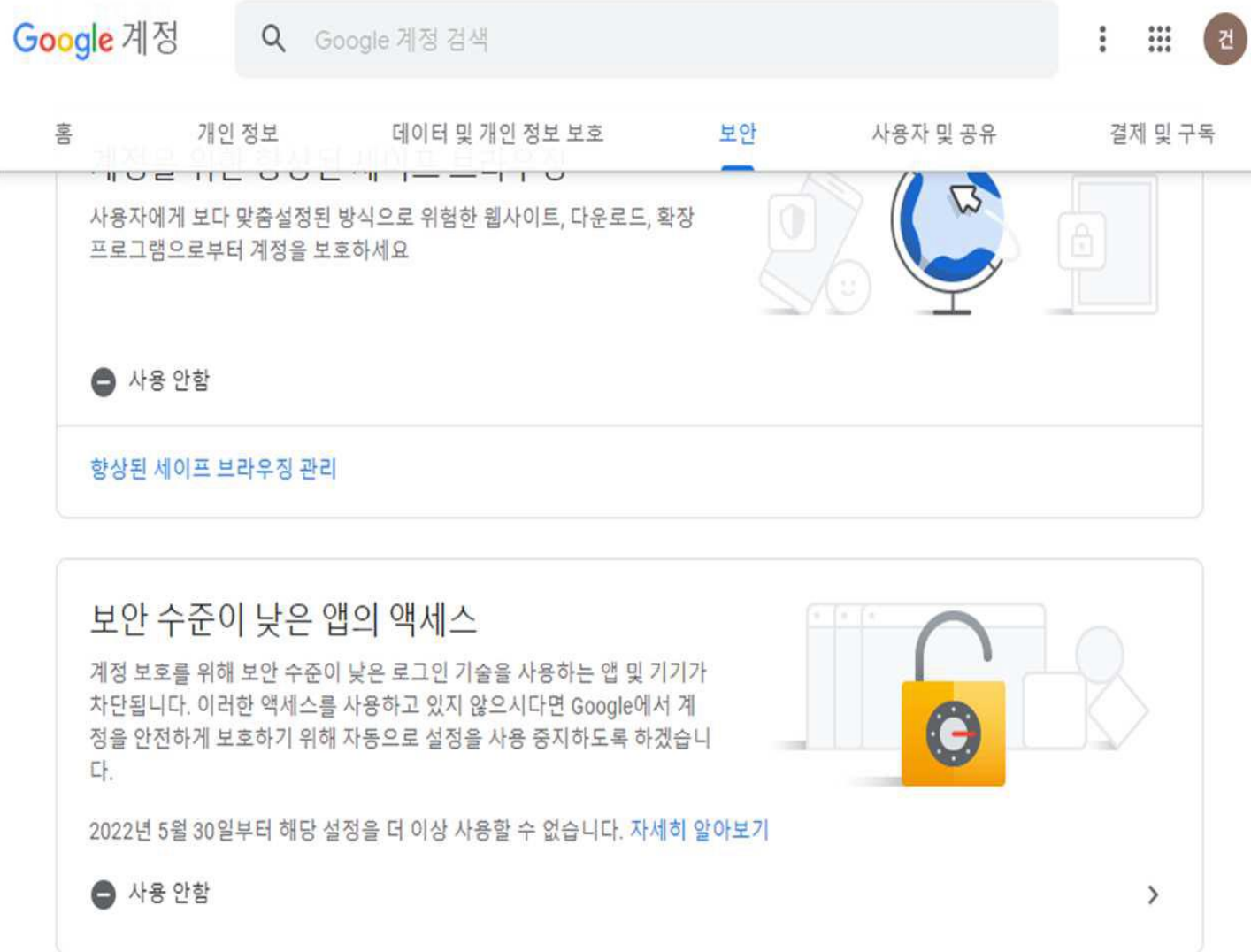
IMAP/SMTP 설정 및 해제 방법

IMAP이란 무엇인가요?

메일 api를 사용하기 전 초기 설정이 필요

네이버로 메일을 전송할 경우
메일 환경설정에서 IMAP/SMTM 설정을
사용함으로 해줘야 함

Mail Api 구현 과정 (3/6)



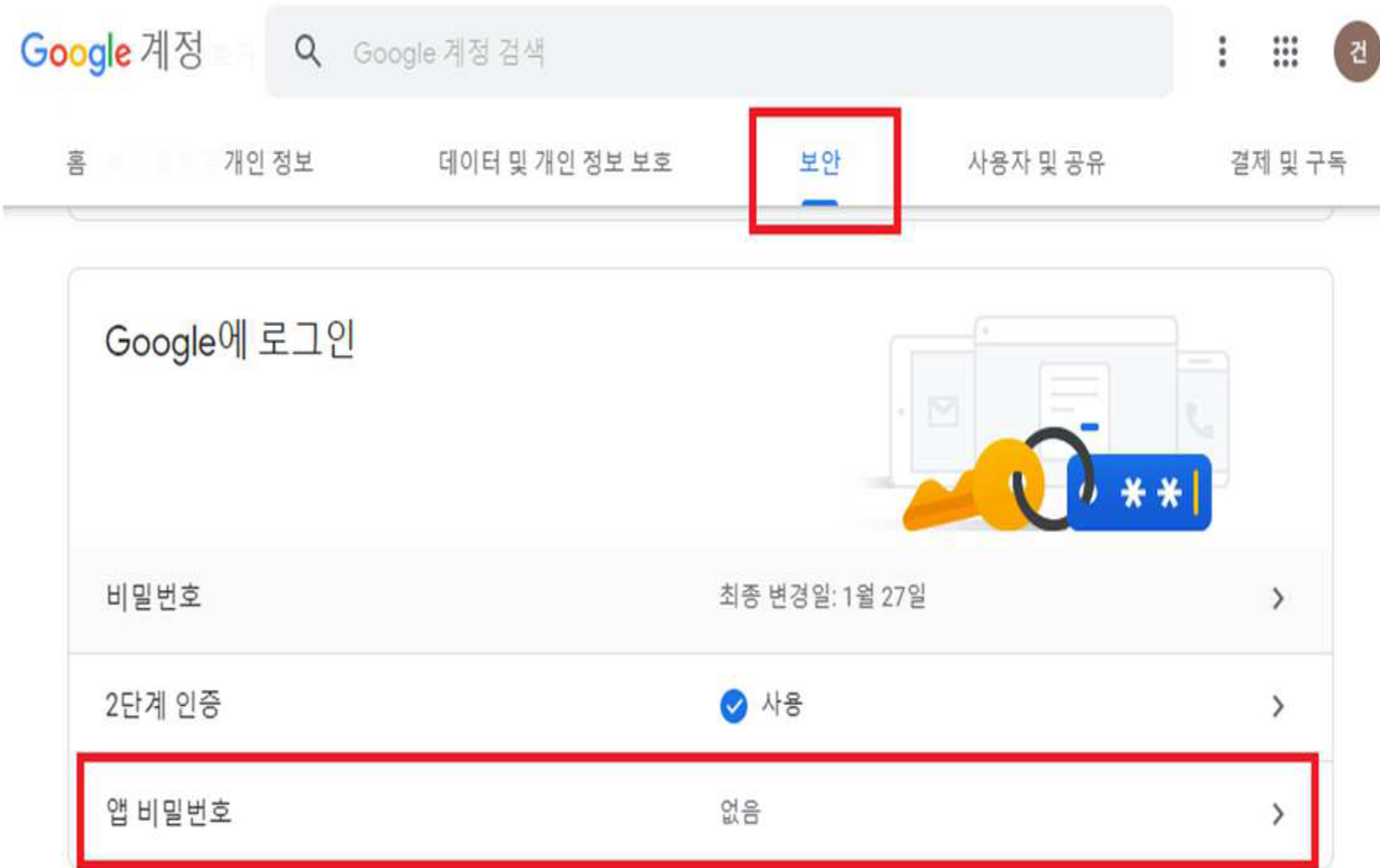
구글로 메일을 보낼 경우

2단계 인증을 안 한 사람과
2단계 인증을 한 사람이 다르게
설정해줘야 함

2단계 인증을 안 한 사람의 경우

계정 설정에서 보안으로 들어간 후
보안 수준이 낮은 앱의 액세스를
사용 안함으로 설정

Mail Api 구현 과정 (4/6)



2단계 인증을 한 사람의 경우

똑같이 계정 설정에서 보안으로
들어간 후 앱 비밀번호를 발급
받아야 함

발급 받은 앱 비밀번호를
코드에 기입할 때
띄어쓰기까지 포함해서 기입

* 구글 관리자 계정이 아닌 경우
(메일 주소가 @gmail.com이
아닌 경우) 메일 api를 이용하지
못할 수도 있음

Mail Api 구현 과정 (5/6)

```
package mail;

import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class SendEmail {
    public static void main(String[] args) {

        String host = "smtp.gmail or naver.com"; // 사용할 사이트
        final String user = "ID@gmail or naver.com"; // ID
        final String password = "PW"; // PW

        String to = "mailAddress"; // 보낼 이메일 주소

        // Get the session object
        Properties props = new Properties();
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", host);
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.port", "587");

        Session session = Session.getDefaultInstance(props, new javax.mail.Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(user, password);
            }
        });

        // Compose the message
        try {
            MimeMessage message = new MimeMessage(session);
            message.setFrom(new InternetAddress(user));
            message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

            // Subject
            message.setSubject("이메일 제목");

            // Text
            message.setText("본문 내용");

            // send the message
            Transport.send(message);
            System.out.println("이메일 전송 성공!");

        } catch (MessagingException e) {
            e.printStackTrace();
        }
    }
}
```

mail-1.4.7.jar

```
Exception in thread "main" java.lang.NoClassDefFoundError: javax/activation/DataSource
    at mail.SendEmail.main(SendEmail.java:38)
Caused by: java.lang.ClassNotFoundException: javax.activation.DataSource
    at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:581)
    at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:178)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
    ... 1 more
```



activation-1.1.1.jar

```
javax.mail.MessagingException: Could not convert socket to TLS;
    nested exception is:
    javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or cipher suites are inappropriate)
    at com.sun.mail.smtp.SMTPTransport.startTLS(SMTPTransport.java:1907)
    at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:666)
    at javax.mail.Service.connect(Service.java:317)
    at javax.mail.Service.connect(Service.java:176)
    at javax.mail.Service.connect(Service.java:125)
    at javax.mail.Transport.send0(Transport.java:194)
    at javax.mail.Transport.send(Transport.java:124)
    at mail.SendEmail.main(SendEmail.java:49)
Caused by: javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or cipher suites are inappropriate)
    at java.base/sun.security.ssl.HandshakeContext.<init>(HandshakeContext.java:170)
    at java.base/sun.security.ssl.ClientHandshakeContext.<init>(ClientHandshakeContext.java:103)
    at java.base/sun.security.ssl.TransportContext.kickstart(TransportContext.java:237)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:404)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:383)
    at com.sun.mail.util.SocketFetcher.configureSSLSocket(SocketFetcher.java:549)
    at com.sun.mail.util.SocketFetcher.startTLS(SocketFetcher.java:486)
    at com.sun.mail.smtp.SMTPTransport.startTLS(SMTPTransport.java:1902)
    ... 7 more
```



```
props.put("mail.smtp.ssl.protocols", "TLSv1.2");
```

코드를 실행할 때 기본적으로
mail-1.4.7.jar 파일을 사용

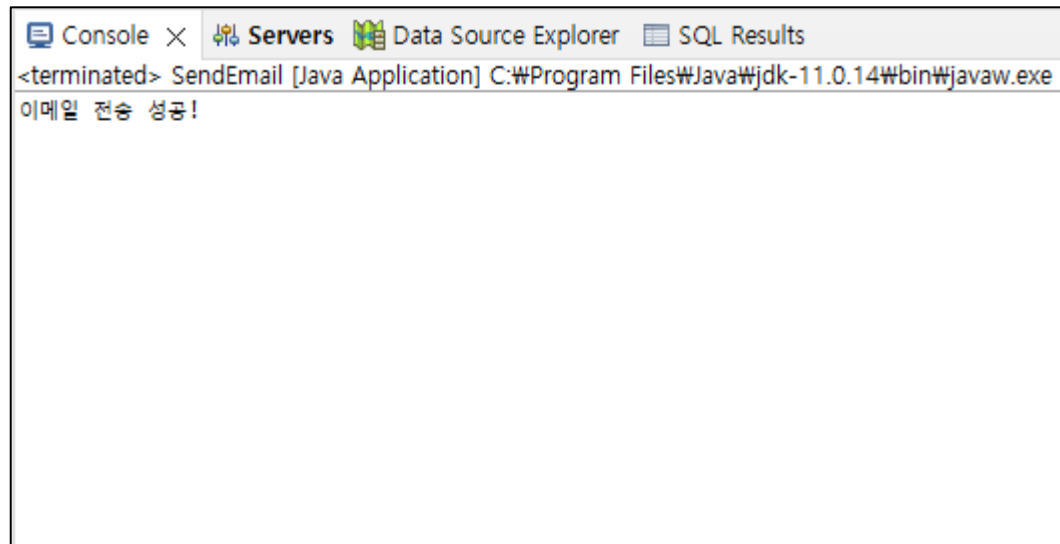
컴파일 했을 때 activation이 없다는
ClassNotFoundException 에러를 확인

activation-1.1.1.jar 파일을
다운 받아 사용

컴파일 했을 때 소켓을 TLS로
변환할 수 없다는 에러를 확인

에러 해결을 위해 TLS 버전을 1.2로
바꿔주는 코드 삽입

Mail Api 구현 과정 (6/6)



The screenshot shows an IDE interface with tabs for Console, Servers, Data Source Explorer, and SQL Results. The Console tab is active and displays the following text:
<terminated> SendEmail [Java Application] C:#Program Files#Java#jdk-11.0.14#bin#javaw.exe
이메일 전송 성공!

Mail Api를 사용해 구현한 결과

☆ 이메일 제목

보낸사람 VIP <sungkwang0908@naver.com>

받는사람 <sungkwang0908@naver.com>

본문 내용

팀 프로젝트 소감

- 소감

팀프로젝트의 뷰를 맡으면서 기존에 잘 만들어진 템플릿을 가져다 쓰는 것에도 불구하고 원하는 구상대로 작업하는 것이 어려웠고 새로운 페이지를 만드는 시간보다 페이지를 분석하는 시간이 더 길었다.

프로젝트를 통해 태그나 속성, 자바스크립트, 제이쿼리, ajax등을 다루면서 숙달될 수 있었고
프론트에서 ajax로 처리해야 하는 부분들에 있어 뷰도 로직을 생각하며 작업을 해야 한다는 것을 느꼈다.
뷰 작업량이 많다고 느낄 때도 있었는데 다른 파트의 팀원들이 많이 도와줘서 힘 내서 작업할 수 있었다.

- 추후 개선사항

처음 웹 어플리케이션 프로젝트를 해보다 보니 View 코드의 경우 가독성이 떨어져 원하는 부분을 찾기 힘들다는 것이 문제였다.
주석 처리와 코드 정리를 통해 코드의 가독성을 높이고 유지보수를 편히 할 수 있도록 개선 예정이다.
ajax를 사용할 때 있어 컨트롤러를 거치지 않고 로직용 JSP 페이지에서 로직을 처리할 수 있도록 구현했는데 View를 View 답게 사용하기 위해
JSP 페이지의 자바 로직을 컨트롤러로 옮겨 로직은 모델이나 컨트롤러에서 다루고
View의 경우 단순히 데이터만을 다루고 처리할 수 있도록 구현할 예정이다.