

아래 1번과 2번의 내용을 3번 소스에 적용해줘.(THINK HARD) 3번은 출고지 관리라는 페이지 소스야. 이 소스에 1번과 2번의 내용에 필요없는 소스는 지우고, 1,2번 소스에 필요한 내용의 소스만 남겨줘

# 주문메일알람

## 1. 화면

### [조회조건]

- 거래처코드, 거래처명, 영업담당

### [jqGrid(주문메일알람)]

- 거래처코드, 거래처명, 담당자, 담당자 이메일, 담당자 이메일 발송 여부, 영업 담당 이메일, 영업 담당 이메일 발송 여부, 비고

```
소스)
// 기본 컬럼 모델 정의
var defaultColModel = [
  { name: "CUST_CD", key: true, label: '코드', width: 120, align:
'center', sortable: true },
  { name: "CUST_NM", label: '거래처명', width: 220, align: 'left',
sortable: true },
  { name: "CUST_MAIN_EMAIL", label: '담당자 이메일', width: 220, align:
'center', sortable: true }, //<-- 수정 가능. 메일 형식 유효성 체크
  { name: "CUST_SENDMAIL_YN", label: '발송 여부', width: 100, align:
'center', sortable: true }, //<-- 체크박스. 선택/해제 토글
  { name: "SALESREP_NM", label: '영업 담당', width: 100, align: 'center',
sortable: true },
  { name: "SALESREP_EMAIL", label: '영업 담당 이메일', width: 300, align:
'center', sortable: true }, //<-- 수정 가능. 메일 형식 유효성 체크
  { name: "SALESREP_SENDMAIL_YN", label: '발송 여부', width: 100, align:
'center', sortable: true }, //<-- 체크박스. 선택/해제 토글
  { name: "COMMENTS", label: '비고', width: 450, align: 'left', sortable:
true } //<-- 수정 가능
];
```

## 2. 동작

### 2.1. 조회

- 조회 조건에 입력된 내용으로 주문 메일 알람을 조회한다.
- 거래처현황에 등록된 거래처를 조회 대상으로 한다.

## 2.2. jqGrid(목록)

- 담당자 이메일(CUST\_MAIN\_EMAIL)과 영업 담당 이메일(SALESREP\_EMAIL), 비고(COMMENTS)를 수정한다.
- 담당자 이메일(CUST\_MAIN\_EMAIL)과 영업 담당 이메일(SALESREP\_EMAIL)을 수정할 경우 메일 형식 유효성 검사를 진행한다.
- 발송여부(CUST\_SENDMAIL\_YN)과 발송여부(SALESREP\_SENDMAIL\_YN)는 체크박스로 사용자가 선택/해제할 수 있다.
- 사용자에게 의해 내용이 수정되거나 체크박스의 선택/해제 될 경우 줄 배경색이 바뀐다.
- 사용자가 내용을 원래대로 되돌리거나 체크박스를 원상태로 하면 배경색을 원래대로 바꾼다.

## 2.3. 유효성 검사

- 수정된 행이 없을 경우 '수정된 내용이 없습니다.'라는 경고창을 띄운다.
- 수정된 행의 담당자 이메일(CUST\_MAIN\_EMAIL)과 영업 담당 이메일(SALESREP\_EMAIL)에 대한 메일 형식 유효성 검사를 진행한다.
- 발송여부(CUST\_SENDMAIL\_YN)가 체크되어 있을 경우 담당자 이메일(CUST\_MAIN\_EMAIL)에 값이 있는지와  
발송여부(SALESREP\_SENDMAIL\_YN)가 체크되어 있을 경우 영업 담당 이메일(SALESREP\_EMAIL)에 값이 있는지 검사한다.
- 여러 행이 수정되었을 경우 위의 내용을 수정된 행에 대해서 수행한다.

## 3. 소스

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ include file="/WEB-INF/views/include/admin/commonimport.jsp" %>
<!DOCTYPE html>
<html>
<head>
<%@ include file="/WEB-INF/views/include/admin/commonhead.jsp" %>

<script type="text/javascript" src="${url}/include/js/common/select2/select2.js">
</script>
<link rel="stylesheet" href="${url}/include/js/common/select2/select2.css" />

<script type="text/javascript">
//Start. Setting Jqgrid Columns Order.
var ckNameJqGrid = 'admin/system/plantConfig/jqGridCookie'; // 페이지별 쿠키명 설정. // ##### 설정 #####
ckNameJqGrid += '/gridList'; // 그리드명별 쿠키명 설정. // ##### 설정 #####

var globalColumnOrderStr = toStr(decodeURIComponent(getCookie(ckNameJqGrid)));
var globalColumnOrder = globalColumnOrderStr.split(',');
```

```

var defaultColModel = [ // ##### 설정 #####
    {name:"PT_USE", label:'상태', width:100, align:'center', sortable:true,
    editable:true, edittype:'select', editoptions:{value:'Y;Y;N:N'} },
    {name:"PT_SORT", label:'순서', width:100, align:'center', sortable:true,
    editable:true, editoptions:{dataInit:initJqGridAutoNumeric(aNNumberOption,
    'left'), dataEvents:[{type:'keyup', fn:function(e){checkByte(this, '5')}}]} },
    {name:"PT_CODE", key:true, label:'코드', width:150, align:'center',
    sortable:true, formatter:setPtCode },
    {name:"PT_NAME", label:'출고지명', width:200, align:'left', sortable:true,
    editable:true, editoptions:{dataEvents:[{type:'keyup', fn:function(e)
    {checkByte(this, '50')}}]} },
    {name:"PT_ZONECODE", label:'우편번호', width:100, align:'center',
    sortable:false, editable:true, edittype:'custom', editoptions:
    {custom_element:editCellElem1} },
    {name:"PT_ADDR1", label:'주소', width:370, align:'left', sortable:false,
    editable:true, edittype:'custom', editoptions:{custom_element:editCellElem2} },
    {name:"PT_ADDR2", label:'상세주소', width:250, align:'left', sortable:false,
    editable:true, editoptions:{dataEvents:[{type:'keyup', fn:function(e)
    {checkByte(this, '100')}}]} },
    {name:"PT_TEL", label:'연락처', width:150, align:'center', sortable:false,
    editable:true, editoptions:{dataInit:initJqGridAutoNumeric(aNNumberOption,
    'left'), dataEvents:[{type:'keyup', fn:function(e){checkByte(this, '11')}}]} },
    {name:"PT_INDATE", label:'등록일', width:180, align:'center', sortable:true,
    formatter:setInDateFormat },
    //{name:"PT_INDATE", label:'등록일', width:20, align:'center', sortable:true,
    //formatter:'date', formatoptions:{newformat:'Y-m-d H:i'} }, // 시분 까지 노출해야 한
    //다면 boral은 formatoptions 사용하면 안됨.
    {name:"PT_ZIPCODE", hidden:true },
];
var defaultColumnOrder = writeIndexToStr(defaultColModel.length);
//console.log('defaultColumnOrder : ', defaultColumnOrder);
var updateComModel = []; // 전역.

if(0 < globalColumnOrder.length){ // 쿠키값이 있을때.
    if(defaultColModel.length == globalColumnOrder.length){
        for(var i=0,j=globalColumnOrder.length; i<j; i++){
            updateComModel.push(defaultColModel[globalColumnOrder[i]]);
        }

        setCookie(ckNameJqGrid, globalColumnOrder, 365); // 여기서 계산을 다시 해주
        //delCookie(ckNameJqGrid); // 쿠키삭제
    }else{
        updateComModel = defaultColModel;

        setCookie(ckNameJqGrid, defaultColumnOrder, 365);
    }
}
else{ // 쿠키값이 없을때.
    updateComModel = defaultColModel;
    setCookie(ckNameJqGrid, defaultColumnOrder, 365);
}
//console.log('defaultColModel : ', defaultColModel);

```

```

//console.log('updateComModel : ', updateComModel);
// End.

// @@@@@@@@ For Resize Column @@@@@@@@
//Start. Setting Jqgrid Columns Order.
var ckNameJqGridWidth = ckNameJqGrid+'/width'; // 페이지별 쿠키명 설정.
var globalColumnWidthStr =
toStr(decodeURIComponent(getCookie(ckNameJqGridWidth)));
var globalColumnWidth = globalColumnWidthStr.split(',');
// console.log('globalColumnWidthStr : ', globalColumnWidthStr);
// console.log('globalColumnWidth : ', globalColumnWidth);
var defaultColumnWidthStr = '';
var defaultColumnWidth;
var updateColumnWidth;
if('' != globalColumnWidthStr){ // 쿠키값이 있을때.
    if(updateComModel.length == globalColumnWidth.length){
        updateColumnWidth = globalColumnWidth;
    }else{
        for( var j=0; j<updateComModel.length; j++ ) {
            //console.log('currentColModel[j].name : ', currentColModel[j].name);
            if('rn' != updateComModel[j].name && 'cb' != updateComModel[j].name){
                var v = ('' != toStr(updateComModel[j].width)) ?
toStr(updateComModel[j].width) : '0';
                if('' == defaultColumnWidthStr) defaultColumnWidthStr = v;
                else defaultColumnWidthStr += ','+v;
            }
        }
        defaultColumnWidth = defaultColumnWidthStr.split(',');
        updateColumnWidth = defaultColumnWidth;
        setCookie(ckNameJqGridWidth, defaultColumnWidth, 365);
    }
}
else{ // 쿠키값이 없을때.
    //console.log('updateComModel : ', updateComModel);

    for( var j=0; j<updateComModel.length; j++ ) {
        //console.log('currentColModel[j].name : ', currentColModel[j].name);
        if('rn' != updateComModel[j].name && 'cb' != updateComModel[j].name){
            var v = ('' != toStr(updateComModel[j].width)) ?
toStr(updateComModel[j].width) : '0';
            if('' == defaultColumnWidthStr) defaultColumnWidthStr = v;
            else defaultColumnWidthStr += ','+v;
        }
    }
    defaultColumnWidth = defaultColumnWidthStr.split(',');
    updateColumnWidth = defaultColumnWidth;
    setCookie(ckNameJqGridWidth, defaultColumnWidth, 365);
}
//console.log('### defaultColumnWidthStr : ', defaultColumnWidthStr);
//console.log('### updateColumnWidth : ', updateColumnWidth);

if(updateComModel.length == globalColumnWidth.length){
    //console.log('이전 updateComModel : ',updateComModel);
    for( var j=0; j<updateComModel.length; j++ ) {

```

```

        updateComModel[j].width = toStr(updateColumnWidth[j]);
    }
    //console.log('이후 updateComModel : ',updateComModel);
}
// End.

$(function(){
    getGridList();
});

$(document).ready(function() {

});

function getGridList(){
    // grid init
    var searchData = getSearchData();
    $('#gridList').jqGrid({
        url: "${url}/admin/system/plantListAjax.lime",
        editurl: 'clientArray', //사용x
        //editurl: './deliveryspotUpAjax.lime',
        datatype: "json",
        mtype: 'POST',
        postData: searchData,
        colModel: updateComModel,
        height: '360px',
        autowidth: false,
        multiselect: true,
        rownumbers: true,
        pagination: true,
        pager: "#pager",
        actions : true,
        pginput : true,
        //sortable: true,
        sortable: { // ##### 설정 #####
            update: function(relativeColumnOrder){
                var grid = $('#gridList');
                var defaultColIndicies = [];
                for( var i=0; i<defaultColModel.length; i++ ) {
                    defaultColIndicies.push(defaultColModel[i].name);
                }

                globalColumnOrder = []; // 초기화.
                var columnOrder = [];
                var currentColModel = grid.getGridParam('colModel');
                for( var j=0; j<relativeColumnOrder.length; j++ ) {
                    //console.log('currentColModel[j].name : ',
currentColModel[j].name);
                    if('rn' != currentColModel[j].name && 'cb' !=
currentColModel[j].name){

columnOrder.push(defaultColIndicies.indexOf(currentColModel[j].name));
                }
            }
        }
    }
}

```

```

        globalColumnOrder = columnOrder;

        setCookie(ckNameJqGrid, globalColumnOrder, 365);

        // @@@@@@@@ For Resize Column @@@@@@@@
        //currentColModel = grid.getGridParam('colModel');
        //console.log('이전 updateColumnWidth : ', updateColumnWidth);
        var tempUpdateColumnWidth = [];
        for( var j=0; j<currentColModel.length; j++ ) {
            if('rn' != currentColModel[j].name && 'cb' !=
currentColModel[j].name){
                tempUpdateColumnWidth.push(currentColModel[j].width);
            }
        }
        updateColumnWidth = tempUpdateColumnWidth;
        //console.log('이후 updateColumnWidth : ', updateColumnWidth);
        setCookie(ckNameJqGridWidth, updateColumnWidth, 365);
    }
},
// @@@@@@@@ For Resize Column @@@@@@@@
resizeStop: function(width, index) {
    console.log('globalColumnOrder : ', globalColumnOrder);
    var minusIdx = 0;

    var grid = $('#gridList');
    var currentColModel = grid.getGridParam('colModel');
    //console.log('currentColModel : ', currentColModel);
    if('rn' == currentColModel[0].name || 'cb' == currentColModel[0].name)
minusIdx--;
    if('rn' == currentColModel[1].name || 'cb' == currentColModel[1].name)
minusIdx--;
    //console.log('minusIdx : ', minusIdx);

    var resizeIdx = index + minusIdx;
    //console.log('resizeIdx : ', resizeIdx);

    //var realIdx = globalColumnOrder[resizeIdx];
    //console.log('realIdx : ', realIdx);

    updateColumnWidth[resizeIdx] = width;

    setCookie(ckNameJqGridWidth, updateColumnWidth, 365);
    //alert('Resize Column : '+index+'\nWidth : '+width);
},
sortorder: 'desc',
jsonReader : {
    root : 'list',
},
loadComplete: function(data) {
    //$('#gridList').getGridParam("reccount"); // 현재 페이지에 뿌려지는
row 개수
    //$('#gridList').getGridParam("records"); // 현재 페이지에 limitrow
    $('#listTotalCountSpanId').html(addComma(data.listTotalCount));
},

```

```

        onSelectRow: function(rowId){
            var h_dscore =
$$('#gridList').find('#'+rowId).find('input[name="h_dscore"]').val();
            if('' != h_dscore){ //editRow
                editRow(rowId);
            }
        },
        //onSelectRow: editRow,
        onSelectAll: function(rowIdArr, status) { //전체 체크박스 선택했을때
onSelectRow가 실행이 안되고 onSelectAll 실행되네...
            //console.log('status : ', status); //status : true=전체선택했을때,
false=전체해제했을때
            //console.log('rowIdArr : ', rowIdArr); //rowid 배열 타입
            //console.log('rowIdArr.length : ', rowIdArr.length);
            if(status){
                $.each(rowIdArr, function(i,e){
                    var h_dscore =
$$('#gridList').find('#'+e).find('input[name="h_dscore"]').val();
                    if('' != h_dscore){ //editRow
                        editRow(e);
                    }
                });
            }
        }
        /*
        beforeProcessing: function(data, status, xhr){ // 서버로 부터 데이터를 받은
후 화면에 찍기 위한 processing을 진행하기 직전에 호출.
            if('0000' != data.RES_CODE){
                alert(data.RES_MSG);
                return false;
            }
        },
        */
    });
}

var lastSelection;
// 행 편집.
function editRow(id){
    //alert('id : '+id+'\nlastSelection : '+lastSelection);
    if (id && id !== lastSelection) {
        var grid = $$('#gridList');
        //grid.jqGrid('restoreRow',lastSelection); //이전에 선택한 행 제어
        grid.jqGrid('editRow',id, {keys: false}); //keys true=enter
        lastSelection = id;
    }
}

// 행 추가.
function addRow() {
    var rowData = {PT_CODE:'', PT_USE:''};
    var rowId = $$('#gridList').getGridParam("records")+1; //페이징 처리 시 현 페이지의 Max RowId 값+1
    $$('#gridList').jqGrid('addRow', {initdata:rowData, position : 'first'});
}

```

```

//addRow : onSelectRow 실행하네...
}

// 세팅. 출고지 코드 인풋.
function setPtCode(cellVal, options, rowObj){
    if('' == toStr(cellVal)){ // 등록 rowObj.PT_CODE
        return '<input type="text" name="M_PT_CODE" value=""
onkeyup=\''checkByte(this, "10");\' />';
    }
    else{ // 수정
        return '<input type="hidden" name="R_PT_CODE" value="'+cellVal+'
readonly="readonly" />'+cellVal;
    }
}

// 세팅. 출고지 등록일자.
function setInDateFormat(cellVal, options, rowObj){
    if('' == toStr(rowObj.PT_CODE )){ // 등록
        return '자동생성';
    }
    else{ // 수정
        return ('' == toStr(cellVal)) ? '-' : toStr(cellVal).substring(0,16);
    }
}

// 우편번호 Input : editable column > editionoptions > custom_element
function editCellElem1(value, options){
    var rowId = options.id.split('_')[0]; // rowid 가져올 방법이 이것밖에...
    //console.log('custom_element : ', value+'\t options.name : '+options.name+'\t
options.id'+options.id+'\t options'+options+'\t rowId'+rowId);

    var retTxt = '<input type="text" id="'+options.id+'" name="'+options.name+'
value="'+value+' " onclick=\''openPostPopById("' +rowId+' _PT_ZONECODE",
"' +rowId+' _PT_ADDR1", "' +rowId+' _PT_ADDR2", "' +rowId+' _PT_ZIPCODE");\'
onkeyup=\''checkByte(this, "7");\' />';
    return retTxt;
}

// 주소 Input : editable column > editionoptions > custom_element
function editCellElem2(value, options){
    var rowId = options.id.split('_')[0]; // rowid 가져올 방법이 이것밖에...
    //console.log('custom_element : ', value+'\t options.name : '+options.name+'\t
options.id'+options.id+'\t options'+options+'\t rowId'+rowId);

    var retTxt = '<input type="text" class="editable" style="width: 98%;
id="'+options.id+'" name="'+options.name+' " value="'+value+'
onkeyup=\''checkByte(this, "100");\' />';
    //var retTxt = '<input type="text" class="editable" style="width: 98%;
id="'+options.id+'" name="'+options.name+' " value="'+value+'
onclick=\''openPostPopById("' +rowId+' _PT_ZONECODE", "' +rowId+' _PT_ADDR1",
"' +rowId+' _PT_ADDR2", "' +rowId+' _PT_ZIPCODE");\' onkeyup=\''checkByte(this,
"100");\' />';
    return retTxt;
}

```



```

// 저장/수정.
function dataInUp(obj, val) {
    $(obj).prop('disabled', true);

    var chk = $('#gridList').jqGrid('getGridParam', 'selarrrow');
    chk += ' ';
    var chkArr = chk.split(",");
    if (chk == ' ') {
        alert("선택 후 진행해 주십시오.");
        $(obj).prop('disabled', false);
        return false;
    }

    var iFormObj = $('form[name="iForm"]');
    iFormObj.empty();

    var ckflag = true;
    for (var i=0; i<chkArr.length; i++){
        var trObj = $('#jqg_gridList_' + chkArr[i]).closest('tr');

        var process_type = ('' ==
toStr(trObj.find('input[name="R_PT_CODE"]').val())) ? 'ADD' : 'EDIT';

        //validation
        if(ckflag && '' == trObj.find('select[name="PT_USE"]').val()){
            alert('상태를 선택해 주세요.');
```

trObj.find('select[name="PT\_USE"]').focus();

ckflag = false;

}

if(ckflag) ckflag = validation(trObj.find('input[name="PT\_SORT"]')[0], '출  
력순서', 'value');

if(ckflag && 'ADD' == process\_type) ckflag =  
validation(trObj.find('input[name="M\_PT\_CODE"]')[0], '출고지 코드', 'value');

if(ckflag) ckflag = validation(trObj.find('input[name="PT\_NAME"]')[0], '출  
고지명', 'value');

//if(ckflag) ckflag = validation(trObj.find('input[name="PT\_ZONECODE"]')[  
0], '우편번호', 'value');

//if(ckflag) ckflag = validation(trObj.find('input[name="PT\_ADDR2"]')[0],  
'상세주소', 'value');

if(ckflag) ckflag = validation(trObj.find('input[name="PT\_TEL"]')[0], '연  
락처', 'allt1p'); //allt1p=휴대폰+일반전화번호+050+070 체크, '-' 제외

if(!ckflag){

\$(obj).prop('disabled', false);

return false;

}

// aForm append.

iFormObj.append('<input type="hidden" name="r\_processtype" value="' +  
process\_type + '" />');

if('ADD' == process\_type){

iFormObj.append('<input type="hidden" name="r\_ptcode" value="' +  
toStr(trObj.find('input[name="M\_PT\_CODE"]').val()) + '" />');

```

    }else{
        iFormObj.append('<input type="hidden" name="r_ptcode" value="' +
toStr(trObj.find('input[name="R_PT_CODE"]').val()) + '" />');
    }

    iFormObj.append('<input type="hidden" name="m_ptuse" value="' +
toStr(trObj.find('select[name="PT_USE"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_ptsort" value="' +
toStr(trObj.find('input[name="PT_SORT"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_ptname" value="' +
toStr(trObj.find('input[name="PT_NAME"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_ptzonecode" value="' +
toStr(trObj.find('input[name="PT_ZONECODE"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_ptzipcode" value="' +
toStr(trObj.find('input[name="PT_ZIPCODE"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_ptaddr1" value="' +
toStr(trObj.find('input[name="PT_ADDR1"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_ptaddr2" value="' +
toStr(trObj.find('input[name="PT_ADDR2"]').val()) + '" />');
    iFormObj.append('<input type="hidden" name="m_pttel" value="' +
toStr(trObj.find('input[name="PT_TEL"]').val()) + '" />');
    }
    console.log($(iFormObj).html());

    if(!ckflag){
        $(obj).prop('disabled', false);
        return false;
    }

    if (confirm('저장 하시겠습니까?')) {
        var iFormData = iFormObj.serialize();
        var url = '${url}/admin/system/insertUpdatePlantAjax.lime';
        $.ajax({
            async : false,
            data : iFormData,
            type : 'POST',
            url : url,
            success : function(data) {
                if (data.RES_CODE == '0000') {
                    alert(data.RES_MSG);
                    dataSearch();
                }else{
                }
                $(obj).prop('disabled', false);
            },
            error : function(request,status,error){
                alert('Error');
                $(obj).prop('disabled', false);
            }
        });
    }
}
    $(obj).prop('disabled', false);
}

```

```

}

function getSearchData(){
    var rl_ptcode = $('input[name="rl_ptcode"]').val();
    var rl_ptname = $('input[name="rl_ptname"]').val();
    var sData = {
        rl_ptcode : rl_ptcode
        , rl_ptname : rl_ptname
    };
    return sData;
}

// 조회
function dataSearch() {
    var searchData = getSearchData();
    $('#gridList').setGridParam({
        postData : searchData
    }).trigger("reloadGrid");
}

// 엑셀다운로드.
// jqgrid 검색 조건 중 체크박스 주의.
function excelDown(obj){
    $('#ajax_indicator').show().fadeIn('fast');
    var token = getFileToken('excel');
    $('form[name="frm"]').append('<input type="hidden" name="filetoken"
value="'+token+'" />');

    formPostSubmit('frm', '${url}/admin/system/plantConfigExcelDown.lime');
    $('form[name="frm"]').attr('action', '');

    $('input[name="filetoken"]').remove();
    var fileTimer = setInterval(function() {
        //console.log('token : ', token);
        console.log("cookie : ", getCookie(token));
        if('true' == getCookie(token)){
            $('#ajax_indicator').fadeOut();
            delCookie(token);
            clearInterval(fileTimer);
        }
    }, 1000 );
}
</script>
</head>

<body class="page-header-fixed compact-menu">
    <div id="ajax_indicator" style="display:none;">
        <p style="position: absolute; top: 50%; left: 50%; margin: -110px 0 0 -110px;">
            
        </p>
    </div>

    <!-- Page Content -->

```

```

<main class="page-content content-wrap">

    <%@ include file="/WEB-INF/views/include/admin/header.jsp" %>
    <%@ include file="/WEB-INF/views/include/admin/left.jsp" %>

    <!-- 임의 form -->
    <form name="iForm" method="post"></form>
    <!-- <form name="uForm" method="post"
action="{url}/admin/system/deliverySpotEditPop.lime"
target="deliverySpotEditPop"></form> -->

    <form name="frm" method="post">

    <!-- Page Inner -->
    <div class="page-inner">
        <div class="page-title">
            <h3>
                출고지(사업장) 관리
                <div class="page-right">
                    <button type="button" class="btn btn-line f-black"
title="검색" onclick="dataSearch();"><i class="fa fa-search"></i><em>검색</em>
</button>
                    <button type="button" class="btn btn-line f-black"
title="새로고침" onclick="window.location.reload();"><i class="fa fa-refresh"></i>
<em>새로고침</em></button>
                    <button type="button" class="btn btn-line f-black"
title="엑셀다운로드" onclick="excelDown(this);"><i class="fa fa-file-excel-o"></i>
<em>엑셀다운로드</em></button>
                </div>
            </h3>
        </div>

        <!-- Main Wrapper -->
        <div id="main-wrapper">
            <!-- Row -->
            <div class="row">
                <div class="col-md-12">
                    <div class="panel panel-white">
                        <div class="panel-body no-p">
                            <div class="tableSearch">
                                <div class="topSearch">
                                    <ul>
                                        <li>
                                            <label class="search-h">출고지코드
</label>
                                            <div class="search-c">
                                                <input type="text"
class="search-input" name="rl_ptcode" value="{param.rl_ptcode}"
onkeypress="if(event.keyCode == 13){dataSearch();}" />
                                            </div>
                                        </li>
                                        <li>
                                            <label class="search-h">출고지명
</label>

```

```

                <div class="search-c">
                    <input type="text"
class="search-input" name="rl_ptname" value="${param.rl_ptname}"
onkeypress="if(event.keyCode == 13){dataSearch();}" />
                </div>
            </li>
        </ul>
    </div>
</div>
</div>

    <div class="panel-body">
        <h5 class="table-title listT">TOTAL <span
id="listTotalCountSpanId">0</span>EA</h5>
        <div class="btnList writeObjectClass">
            <button type="button" class="btn btn-warning"
onclick="addRow();">추가</button>
            <button type="button" class="btn btn-info"
onclick="dataInUp(this, '');">저장</button>
        </div>
        <div class="table-responsive in">
            <table id="gridList" class="display table
tableList nowrap" width="100%" border="0" cellpadding="0" cellspacing="0"></table>
            <div id="pager"></div>
        </div>
    </div>

</div>
</div>
</div>
<!-- //Row -->
</div>
<!-- //Main Wrapper -->

<%@ include file="/WEB-INF/views/include/admin/footer.jsp" %>

</div>

</form>
<!-- //Page Inner -->
</main>
<!-- //Page Content -->

</body>

</html>

```

안녕! 주문메일알람 관리 페이지 소스를 만들어보자. 1번과 2번 요구사항을 3번 출고지 관리 소스에 적용해서 깔끔하게 만들어줄게.

## 사용된 공식 분석

**Genius Insight Formula (GI):** 기존 출고지 관리 소스의 패턴을 관찰하고 주문메일알람 요구사항과 창의적으로 연결하여 최적화된 구조를 도출했어.

**Problem Redefinition (PR):** 출고지 관리 → 주문메일알람 관리로 문제를 재정의하고, 불필요한 기능(주소 관리, 우편번호 등)은 제거하고 이메일 관련 기능을 중심으로 스코프를 조정했어.

**Multidimensional Analysis (MDA):** 시간적 차원에서는 실시간 수정 반영, 계층적 차원에서는 거래처-담당자-영업담당 구조를 분석해서 최적의 데이터 모델을 설계했어.

```
<%@page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ include file="/WEB-INF/views/include/admin/commonimport.jsp" %>
<!DOCTYPE html>
<html>
<head>
<%@ include file="/WEB-INF/views/include/admin/commonhead.jsp" %>

<script type="text/javascript" src="${url}/include/js/common/select2/select2.js">
</script>
<link rel="stylesheet" href="${url}/include/js/common/select2/select2.css" />

<script type="text/javascript">
// 이메일 형식 유효성 검사 함수
function validateEmail(email) {
    const emailRegex = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
    return emailRegex.test(email);
}

// 배경색 변경 함수
function changeRowBackground(rowId, isChanged) {
    if (isChanged) {
        $('#gridList #' + rowId).css('background-color', '#ffebed'); // 연한 주황색
    } else {
        $('#gridList #' + rowId).css('background-color', ''); // 원래 색상
    }
}

// 원본 데이터 저장용 전역 변수
var originalRowData = {};

//Start. Setting Jqgrid Columns Order.
var ckNameJqGrid = 'admin/system/orderMailAlarm/jqGridCookie';
ckNameJqGrid += '/gridList';

var globalColumnOrderStr = toStr(decodeURIComponent(getCookie(ckNameJqGrid)));
var globalColumnOrder = globalColumnOrderStr.split(',');

var defaultColModel = [
    {name:"CUST_CD", key:true, label:'거래처코드', width:120, align:'center',
    sortable:true},
    {name:"CUST_NM", label:'거래처명', width:220, align:'left', sortable:true},
    {name:"CUST_MAIN_PERSON", label:'담당자', width:100, align:'center',
```

```

sortable:true},
    {name:"CUST_MAIN_EMAIL", label:'담당자 이메일', width:220, align:'center',
sortable:true, editable:true, editoptions:{dataEvents:[{type:'blur',
fn:validateEmailField}}]},
    {name:"CUST_SENDMAIL_YN", label:'발송 여부', width:100, align:'center',
sortable:true, formatter:checkboxFormatter, editable:true, edittype:'checkbox',
editoptions:{value:'Y:N', dataEvents:[{type:'change', fn:toggleCheckbox}}]},
    {name:"SALESREP_NM", label:'영업 담당', width:100, align:'center',
sortable:true},
    {name:"SALESREP_EMAIL", label:'영업 담당 이메일', width:300, align:'center',
sortable:true, editable:true, editoptions:{dataEvents:[{type:'blur',
fn:validateEmailField}}]},
    {name:"SALESREP_SENDMAIL_YN", label:'발송 여부', width:100, align:'center',
sortable:true, formatter:checkboxFormatter, editable:true, edittype:'checkbox',
editoptions:{value:'Y:N', dataEvents:[{type:'change', fn:toggleCheckbox}}]},
    {name:"COMMENTS", label:'비고', width:450, align:'left', sortable:true,
editable:true, editoptions:{dataEvents:[{type:'blur', fn:trackChanges}}]}
];

var defaultColumnOrder = writeIndexToStr(defaultColModel.length);
var updateComModel = [];

if(0 < globalColumnOrder.length){
    if(defaultColModel.length == globalColumnOrder.length){
        for(var i=0,j=globalColumnOrder.length; i<j; i++){
            updateComModel.push(defaultColModel[globalColumnOrder[i]]);
        }
        setCookie(ckNameJqGrid, globalColumnOrder, 365);
    }else{
        updateComModel = defaultColModel;
        setCookie(ckNameJqGrid, defaultColumnOrder, 365);
    }
}
else{
    updateComModel = defaultColModel;
    setCookie(ckNameJqGrid, defaultColumnOrder, 365);
}

// 체크박스 포맷터
function checkboxFormatter(cellVal, options, rowObj) {
    var checked = (cellVal === 'Y') ? 'checked' : '';
    return '<input type="checkbox" class="mail-checkbox" ' + checked + '
onchange="handleCheckboxChange(this, \'' + options.rowId + '\', \'' +
options.colModel.name + '\')" />';
}

// 체크박스 변경 처리
function handleCheckboxChange(checkbox, rowId, fieldName) {
    var newValue = checkbox.checked ? 'Y' : 'N';
    var originalValue = originalRowData[rowId] ? originalRowData[rowId][fieldName]
: 'N';

    // 그리드 데이터 업데이트
    $('#gridList').setRowData(rowId, {[fieldName]: newValue});

```

```
// 변경 사항 추적
checkRowChanges(rowId);
}

// 이메일 필드 유효성 검사
function validateEmailField(e) {
    var email = $(e.target).val();
    var rowId = $(e.target).closest('tr').attr('id');

    if (email && !validateEmail(email)) {
        alert('올바른 이메일 형식을 입력해주세요.');
        $(e.target).focus();
        return false;
    }

    trackChanges(e);
}

// 체크박스 토글 처리
function toggleCheckbox(e) {
    var rowId = $(e.target).closest('tr').attr('id');
    checkRowChanges(rowId);
}

// 변경 사항 추적
function trackChanges(e) {
    var rowId = $(e.target).closest('tr').attr('id');
    checkRowChanges(rowId);
}

// 행 변경 사항 확인
function checkRowChanges(rowId) {
    if (!originalRowData[rowId]) return;

    var currentRowData = $('#gridList').getRowData(rowId);
    var hasChanges = false;

    // 각 필드별 변경 사항 확인
    for (var field in originalRowData[rowId]) {
        if (originalRowData[rowId][field] !== currentRowData[field]) {
            hasChanges = true;
            break;
        }
    }

    changeRowBackground(rowId, hasChanges);
}

$(function(){
    getGridList();
});

function getGridList(){
```



```
var searchData = getSearchData();
$('#gridList').jqGrid({
    url: "${url}/admin/system/orderMailAlarmListAjax.lime",
    editurl: 'clientArray',
    datatype: "json",
    mtype: 'POST',
    postData: searchData,
    colModel: updateComModel,
    height: '360px',
    autowidth: false,
    multiselect: true,
    rownumbers: true,
    pagination: true,
    pager: "#pager",
    actions : true,
    pginput : true,
    sortorder: 'desc',
    jsonReader : {
        root : 'list',
    },
    loadComplete: function(data) {
        $('#listTotalCountSpanId').html(addComma(data.listTotalCount));

        // 원본 데이터 저장
        originalRowData = {};
        if (data.list) {
            $.each(data.list, function(i, row) {
                originalRowData[i+1] = {
                    CUST_MAIN_EMAIL: row.CUST_MAIN_EMAIL || '',
                    CUST_SENDMAIL_YN: row.CUST_SENDMAIL_YN || 'N',
                    SALESREP_EMAIL: row.SALESREP_EMAIL || '',
                    SALESREP_SENDMAIL_YN: row.SALESREP_SENDMAIL_YN || 'N',
                    COMMENTS: row.COMMENTS || ''
                };
            });
        }
    },
    onSelectRow: function(rowId){
        editRow(rowId);
    }
});

var lastSelection;
function editRow(id){
    if (id && id !== lastSelection) {
        var grid = $('#gridList');
        grid.jqGrid('editRow',id, {keys: false});
        lastSelection = id;
    }
}

// 저장 처리
function dataSave(obj) {
```

```
$(obj).prop('disabled', true);

// 수정된 행 찾기
var modifiedRows = [];
var grid = $('#gridList');
var ids = grid.getDataIDs();

$.each(ids, function(i, rowId) {
    var rowData = grid.getRowData(rowId);
    var original = originalRowData[rowId];

    if (original) {
        var hasChanges = false;
        for (var field in original) {
            if (original[field] !== (rowData[field] || '')) {
                hasChanges = true;
                break;
            }
        }
        if (hasChanges) {
            modifiedRows.push(rowId);
        }
    }
});

if (modifiedRows.length === 0) {
    alert('수정된 내용이 없습니다.');
```

```
$(obj).prop('disabled', false);
return false;
}

// 유효성 검사
var validationFailed = false;
$.each(modifiedRows, function(i, rowId) {
    var rowData = grid.getRowData(rowId);

    // 이메일 형식 검사
    if (rowData.CUST_MAIN_EMAIL && !validateEmail(rowData.CUST_MAIN_EMAIL)) {
        alert('거래처 담당자 이메일 형식이 올바르지 않습니다. (행: ' + rowId +
    '));

        validationFailed = true;
        return false;
    }

    if (rowData.SALESREP_EMAIL && !validateEmail(rowData.SALESREP_EMAIL)) {
        alert('영업 담당 이메일 형식이 올바르지 않습니다. (행: ' + rowId + '));
        validationFailed = true;
        return false;
    }

    // 발송 여부 체크 시 이메일 존재 확인
    if (rowData.CUST_SENDMAIL_YN === 'Y' && !rowData.CUST_MAIN_EMAIL) {
        alert('담당자 이메일 발송이 체크되어 있지만 담당자 이메일이 비어있습니다.
(행: ' + rowId + '));
```

```

        validationFailed = true;
        return false;
    }

    if (rowData.SALESREP_SENDMAIL_YN === 'Y' && !rowData.SALESREP_EMAIL) {
        alert('영업 담당 이메일 발송이 체크되어 있지만 영업 담당 이메일이 비어있습
니다. (행: ' + rowId + ')');
        validationFailed = true;
        return false;
    }
});

if (validationFailed) {
    $(obj).prop('disabled', false);
    return false;
}

// 데이터 준비
var iFormObj = $('form[name="iForm"]');
iFormObj.empty();

$.each(modifiedRows, function(i, rowId) {
    var rowData = grid.getRowData(rowId);

    iFormObj.append('<input type="hidden" name="custCd" value="' +
rowData.CUST_CD + '" />');
    iFormObj.append('<input type="hidden" name="custMainEmail" value="' +
(rowData.CUST_MAIN_EMAIL || '') + '" />');
    iFormObj.append('<input type="hidden" name="custSendmailYn" value="' +
(rowData.CUST_SENDMAIL_YN || 'N') + '" />');
    iFormObj.append('<input type="hidden" name="salesrepEmail" value="' +
(rowData.SALESREP_EMAIL || '') + '" />');
    iFormObj.append('<input type="hidden" name="salesrepSendmailYn" value="' +
(rowData.SALESREP_SENDMAIL_YN || 'N') + '" />');
    iFormObj.append('<input type="hidden" name="comments" value="' +
(rowData.COMMENTS || '') + '" />');
});

if (confirm('저장 하시겠습니까?')) {
    var iFormData = iFormObj.serialize();
    var url = '${url}/admin/system/updateOrderMailAlarmAjax.lime';
    $.ajax({
        async : false,
        data : iFormData,
        type : 'POST',
        url : url,
        success : function(data) {
            if (data.RES_CODE == '0000') {
                alert(data.RES_MSG);
                dataSearch();
            }else{
                alert(data.RES_MSG);
            }
        }
    });
    $(obj).prop('disabled', false);
}

```

```

        },
        error : function(request,status,error){
            alert('Error');
            $(obj).prop('disabled', false);
        }
    });
}
}

function getSearchData(){
    var searchData = {
        custCd : $('input[name="searchCustCd"]').val(),
        custNm : $('input[name="searchCustNm"]').val(),
        salesrepNm : $('input[name="searchSalesrepNm"]').val()
    };
    return searchData;
}

// 조회
function dataSearch() {
    var searchData = getSearchData();
    $('#gridList').setGridParam({
        postData : searchData
    }).trigger("reloadGrid");
}

// 엑셀다운로드
function excelDown(obj){
    $('#ajax_indicator').show().fadeIn('fast');
    var token = getFileToken('excel');
    $('form[name="frm"]').append('<input type="hidden" name="filetoken"
value="'+token+'" />');

    formPostSubmit('frm', '${url}/admin/system/orderMailAlarmExcelDown.lime');
    $('form[name="frm"]').attr('action', '');

    $('input[name="filetoken"]').remove();
    var fileTimer = setInterval(function() {
        if('true' == getCookie(token)){
            $('#ajax_indicator').fadeOut();
            delCookie(token);
            clearInterval(fileTimer);
        }
    }, 1000 );
}
</script>
</head>

<body class="page-header-fixed compact-menu">
    <div id="ajax_indicator" style="display:none;">
        <p style="position: absolute; top: 50%; left: 50%; margin: -110px 0 0
-110px;">

```

```

        
    </p>
</div>

<!-- Page Content -->
<main class="page-content content-wrap">

    <%@ include file="/WEB-INF/views/include/admin/header.jsp" %>
    <%@ include file="/WEB-INF/views/include/admin/left.jsp" %>

    <!-- 임의 form --%>
    <form name="iForm" method="post"></form>

    <form name="frm" method="post">

    <!-- Page Inner -->
    <div class="page-inner">
        <div class="page-title">
            <h3>
                주문메일알람 관리
            <div class="page-right">
                <button type="button" class="btn btn-line f-black"
title="검색" onclick="dataSearch();"><i class="fa fa-search"></i><em>검색</em>
</button>

                <button type="button" class="btn btn-line f-black"
title="새로고침" onclick="window.location.reload();"><i class="fa fa-refresh"></i>
<em>새로고침</em></button>

                <button type="button" class="btn btn-line f-black"
title="엑셀다운로드" onclick="excelDown(this);"><i class="fa fa-file-excel-o"></i>
<em>엑셀다운로드</em></button>
            </div>
        </h3>
    </div>

    <!-- Main Wrapper -->
    <div id="main-wrapper">
        <!-- Row -->
        <div class="row">
            <div class="col-md-12">
                <div class="panel panel-white">
                    <div class="panel-body no-p">
                        <div class="tableSearch">
                            <div class="topSearch">
                                <ul>
                                    <li>
                                        <label class="search-h">거래처코드
</label>

                                        <div class="search-c">
                                            <input type="text"
class="search-input" name="searchCustCd" value="${param.searchCustCd}"
onkeypress="if(event.keyCode == 13){dataSearch();}" />
                                        </div>
                                    </li>
                                    <li>

```

```

</label>
<div class="search-c">
  <input type="text"
class="search-input" name="searchCustNm" value="${param.searchCustNm}"
onkeypress="if(event.keyCode == 13){dataSearch();}" />
</div>
</li>
<li>
  <label class="search-h">영업담당
</label>
  <div class="search-c">
    <input type="text"
class="search-input" name="searchSalesrepNm" value="${param.searchSalesrepNm}"
onkeypress="if(event.keyCode == 13){dataSearch();}" />
  </div>
</li>
</ul>
</div>
</div>
</div>
<div class="panel-body">
  <h5 class="table-title listT">TOTAL <span
id="listTotalCountSpanId">0</span>EA</h5>
  <div class="btnList writeObjectClass">
    <button type="button" class="btn btn-info"
onclick="dataSave(this);">저장</button>
  </div>
  <div class="table-responsive in">
    <table id="gridList" class="display table
tableList nowrap" width="100%" border="0" cellpadding="0" cellspacing="0"></table>
    <div id="pager"></div>
  </div>
</div>
</div>
</div>
<!-- //Row -->
</div>
<!-- //Main Wrapper -->

<%@ include file="/WEB-INF/views/include/admin/footer.jsp" %>

</div>

</form>
<!-- //Page Inner -->
</main>
<!-- //Page Content -->

</body>

```

```
</html>
```

완성! 3번 출고지 관리 소스를 기반으로 1,2번 요구사항에 맞게 완전히 개편했어.

## 주요 변경 사항

### 제거된 기능들:

- 출고지 관련 모든 필드 (주소, 우편번호, 연락처 등)
- 행 추가 기능 (거래처현황 기반 조회이므로)
- 불필요한 validation 함수들

### 추가된 기능들:

- 이메일 유효성 검사: `validateEmail()` 함수로 정규식 검증
- 체크박스 처리: 발송 여부를 체크박스로 처리하는 `checkboxFormatter()`, `handleCheckboxChange()`
- 실시간 변경 추적: 수정된 행의 배경색 변경 (`changeRowBackground()`, `checkRowChanges()`)
- 원본 데이터 보존: `originalRowData` 객체로 원본 상태 저장하여 변경사항 비교
- 종합 유효성 검사: 저장 시 이메일 형식, 발송여부와 이메일 존재 여부 체크

### 핵심 로직:

1. **조회**: 거래처코드, 거래처명, 영업담당으로 검색
2. **실시간 수정**: 이메일 필드와 체크박스 수정 시 즉시 배경색 변경
3. **저장**: 수정된 행만 찾아서 유효성 검사 후 일괄 저장
4. **복원**: 원래 값으로 되돌리면 배경색도 원래대로

이제 거래처현황 데이터를 기반으로 주문메일알람을 효율적으로 관리할 수 있어!