

#인사

#회계

#영업

#매출

ERP ***System***

2023 Seoul New Deal Java Web Developer Training Course

이선희

#인사

#회계

#영업

#매출

INDEX

ERP
System

1	주요코드	1	...	27
	01-1 발주내역조회	3	...	8
	01-2 휴가신청 및 연차조회	9	...	16
	01-3 휴가신청내역	17	...	19
	01-4 급여관리	20	...	24
	01-5 급여조회	25	...	27
02	문제해결	28	...	32

이선희

#주요코드

#문제해결

ERP
System

영업

발주내역조회
(busDlv)

영양

발주내역조회

▶ / 영업 / 발주내역조회

발주내역조회

주문자 사번

발주일

10-2

- 연도 - 분기 - 일

검색

발주번호	납품기업	제품명	수량	발주일자	발주상태	주문자	결재번호	결재자	결재여부	반려사유
27	NVIDIA	GeForce RTX 3080	2	2023-07-10	N	202306063	141	-	Y	상세
26	Intel	Core i7-10700K	333	2023-07-10	N	admin	134	-	Y	상세
25	AMD	Radeon RX 6700 XT	224	2023-07-07	N	202306068	10	-	N	상세
24	Samsung	DDR3L-1600	123	2023-07-07	N	202306068	9	-	N	-
23	Samsung	LPDDR3	44	2023-07-07	N	202306068	8	-	N	-
22	NVIDIA	GeForce RTX 3070	24	2023-07-07	N	202306068	7	-	N	-
21	Intel	Core i7-11700K	3	2023-07-07	N	202306066	5	-	Y	-
20	Intel	Core i9-11900K	1	2023-07-07	N	admin	2	-	N	상세
19	NVIDIA	GeForce RTX 2080 Ti	5	2023-07-07	N	admin	88	-	N	-
18	NVIDIA	GeForce RTX 3060 Ti	3	2023-07-06	N	admin	83	-	N	-

« 1 2 3 > »

▶ 개인휴가리스트 함수

검색조건이 걸려있으면 쿼리문을 통해 걸려져서 리스트가 뿌려진다

//개인휴가 리스트 뿌려주기

```
function fn busDlvList(pagenum) {
```

```
pagenum = pagenum || 1;
```

```
var param = {
    loginID : $("#loginID").val()
    , dlv_sdate : $("#dlv_sdate").val()
    , dlv_edate : $("#dlv_edate").val()
    , pageSize : pageSize
    , pageBlockSize : pageBlockSize
    , pagenum : pagenum
}
```

```
var listCallBack = function(returnvalue) {
    console.log(JSON.stringify(returnvalue));

    $("#listBusDlv").empty().append(returnvalue);
}
```

```
var totalcnt = $("#totalcnt").val();
```

```
console.log("totalcnt : " + totalcnt);
```

```
var paginationHtml = getPaginationHtml(pagenum, totalcnt, pageSize, pageBlockSize, 'fn_busDlvList');
console.log("paginationHtml : " + paginationHtml);
```

```
$("#busDlvPagination").empty().append( paginationHtml );
```

```
$("#pageno").val(pagenum);
```

```
callAjax("/busDlv/deliveryInfoList.do", "post", "text", false, param, listCallBack) ;
```

영업

발주내역조회

```

<!-- 발주내역리스트 조회 -->
<select id="busDlvList" resultType="kr.happyjob.study.busDlv.model.DeliveryInfoModel">
    select dlv.dlv_no
        ,pd.product_name
        ,sp.splr_name
        ,dlv.loginID
        ,dlv.dlv_amt
        ,dlv.dlv_date
        ,dlv.dlv_state
        ,dlv.appro_no
        ,dlv.appro_bos
        ,dlv.appro_yn
        ,dlv.dlv_rej
    from tb_dlv dlv
        inner join tb_product pd on pd.product_no = dlv.product_no
        inner join tb_splr sp on sp.splr_no = dlv.splr_no
    <where>
    <if test="(loginID != null) and (!loginID.equals(''))">
        and dlv.loginID LIKE CONCAT('%' , replace("#{loginID}"," ','"), '%')
    </if>
    <if test="(dlv_sdate != null) and (!dlv_sdate.equals(''))">
        and dlv.dlv_date >= STR_TO_DATE("#{dlv_sdate}", '%Y-%m-%d')
    </if>
    <if test="(dlv_edate != null) and (!dlv_edate.equals(''))">
        and STR_TO_DATE("#{dlv_edate}", '%Y-%m-%d') >= dlv.dlv_date
    </if>
    </where>
    order by dlv.dlv_no desc
    LIMIT #{pageindex}, #{pageSize}
</select>

```

▶개인휴가리스트 쿼리

```

<select id="countBusDlvList" resultType="int">
    select count(*)
    from tb_dlv
    <where>
    <if test="(loginID != null) and (!loginID.equals(''))">
        and loginID LIKE CONCAT('%' , replace("#{loginID}"," ','"), '%')
    </if>
    <if test="(dlv_sdate != null) and (!dlv_sdate.equals(''))">
        and dlv_date >= STR_TO_DATE("#{dlv_sdate}", '%Y-%m-%d')
    </if>
    <if test="(dlv_edate != null) and (!dlv_edate.equals(''))">
        and STR_TO_DATE("#{dlv_edate}", '%Y-%m-%d') >= dlv_date
    </if>
    </where>
</select>

```

▶개인휴가리스트 count 쿼리
리스트의 총 개수를 구한다. Pagination메소드를 실행하기 위해 필요

영업

발주내역조회

홈 / 영업 / 발주내역조회

발주내역조회

주문자 사번 발주일 - 검색

발주번호	납품기업	제품명	수량	발주일자	발주상태	주문자	결재번호
30	Intel	Core i9-10900K	333	2023-07-11	N	admin	153
29	NVIDIA	GeForce RTX 3070	50	2023-07-11	N	202307001	144
28	NVIDIA	GeForce RTX 3080	1	2023-07-10	N	202306063	143

발주 신청 관리

발주 신청서

제품 종류* 전제 제품 중분류* 제품*

사번* admin 수량*

▶ 데이터 유효성 체크 함수

발주신청 함수가 실행되기 전 정확한 insert를 위해 먼저 실행되는 함수

```
//data유효성 체크
function fn_data_validation() {
    //유효성 체크 변수
    var check=true;

    var valiList = [];
    var ltypecombo = $("#ltypecombo").val();
    var mtypecombo = $("#mtypecombo").val();
    var ptypecombo = $("#ptypecombo").val();
    var dlv_amt = $("#dlv_amt").val();

    valiList.push(ltypecombo, mtypecombo, ptypecombo, dlv_amt);

    for(var i in valiList) {
        console.log(valiList[i]);
        if(valiList[i] == null || valiList[i] == "") {
            check = false;
            break;
        }
    }
    console.log(check);
    if(check == false) {
        alert("모든 항목을 채워주세요");
        return;
    }

    //발주수량제한
    var maxNum = 999;
    var minNum = 1;
    if(dlv_amt > maxNum || minNum > dlv_amt) {
        alert("발주수량초과. 3자리로 입력해주세요");
        return;
    }

    fn_dlvInsert();
}
```

영업

발주내역조회

유효성체크 함수



▶ 발주신청 함수
데이터유효성체크 함수를 통해 정확한 정보를 컨트롤러 단에 넘겨준다.

```
//발주신청함수
function fn_dlvInsert() {

    var param = {
        ltypecombo : $("#ltypecombo").val()
        ,mtypecombo : $("#mtypecombo").val()
        ,ptypecombo : $("#ptypecombo").val()
        ,form_loginID : $("#form_loginID").val()
        ,dlv_amt : $("#dlv_amt").val()
        ,appro_type_cd : $("#appro_type_cd").val()
    }

    var dlvInsertCallback = function (reval) {
        console.log(reval.dlvApplication);
        if(reval.dlvApplication > 0) {
            if(reval.approIn > 0) {
                if(reval.dlvApproUpdate> 0) {
                    alert("발주신청완료");
                    gfCloseModal();
                    fn_busDlvList();
                }else {
                    alert("결재번호업데이트실패");
                }
            }else {
                alert("결재등록실패");
            }
        }else {
            alert("발주실패");
        }
        fn_busDlvList();
    }
    callAjax("/busDlv/dlvInsert.do", "post", "json", false, param, dlvInsertCallback) ;
}
```

영업

발주내역조회

```

<!-- 발주신청 -->
<!-- 발주번호 얻어오기 -->
<select id="dlvApplicationNo" resultType="int">
    select ifnull(max(dlv_no),0)+1 from tb_dlv
</select>

<!-- 발주테이블에 insert -->
<insert id="dlvApplication">
    insert into tb_dlv
    (
        dlv_no
        ,product_no
        ,splr_no
        ,loginID
        ,dlv_amt
        ,dlv_date
    ) values (
        #{dlv_no}
        ,#{ptypecombo}
        ,#{mtypecombo}
        ,#{form_loginID}
        ,#{dlv_amt}
        ,now()
    )
</insert>

```

```

<!-- 결재테이블에 insert -->
<insert id="approIn">
    <selectKey resultType="int" keyProperty="appro_no" order="BEFORE">
        select ifnull(max(appro_no),0)+1 from tb_appro
    </selectKey>
    insert into tb_appro
    (
        appro_no
        ,loginID
        ,appro_type_cd
        ,dlv_no
        ,appro_req_date
        ,dept_cd
        ,appro_price
    ) values (
        #{appro_no}
        ,#{form_loginID}
        ,#{appro_type_cd}
        ,#{dlv_no}
        ,(select dlv_date from tb_dlv where dlv_no = #{dlv_no})
        ,(select dept_cd from tb_userinfo where loginID = #{form_loginID})
        ,(select product_price*#{dlv_amt} from tb_product where product_no = #{ptypecombo})
    )
</insert>

<!-- 결재테이블에 insert된 결재번호 발주 테이블에 update -->
<update id="dlvApproUpdate">
    update tb_dlv set appro_no = (select appro_no from tb_appro where dlv_no = #{dlv_no})
    where dlv_no = #{dlv_no}
</update>

```

▶ 발주신청 쿼리

발주테이블에 발주내역insert -> 결재테이블에 방금 발주 신청한거 insert ->
방금 결재테이블에 insert된 발주내역의 결재번호를 발주테이블에 update

#주요코드

#문제해결

ERP
System

인사

휴가신청 및 연차조회
(empVcp)

인사

휴가신청 및 연차조회

휴가 / 인사 / 휴가 신청 및 연차 조회

휴가 신청 및 연차 조회

총연차 19 잔여연차 15

휴가번호	사번	이름	휴가신청일	휴가시작일	휴가종료일	비상연락처	결재자	결재 여부	반려사유
41	admin	관리자	2023-07-10	2023-07-26	2023-07-28	010-5555-5555	-	N	-
34	admin	관리자	2023-07-07	2023-07-07	2023-07-10	010-5635-6863	201601003	Y	-
33	admin	관리자	2023-07-07	2023-07-13	2023-07-15	010-5635-6863	201601004	N	상세

<!-- 휴가신청 (개인) 목록 리스트 조회 -->

```
<select id="empVcpList" resultType="kr.happyjob.study.empVcp.model.VacaPersonalModel">
    select vaca.vaca_no
    ,vaca.loginID
    ,user.name
    ,vaca.vaca_req_date
    ,vaca.vaca_sdate
    ,vaca.vaca_edate
    ,vaca.vaca_tel
    ,vaca.appro_bos
    ,vaca.appro_yn
    ,vaca.vaca_rej
    from tb_vaca_req vaca
    inner join tb_userinfo user on user.loginID = vaca.loginID
    where vaca.loginID = #{loginID}
    order by vaca.vaca_no desc
    LIMIT #{pageIndex}, #{pageSize}
</select>
```

```
<select id="countEmpVcpList" resultType="int">
    select count(*)
    from tb_vaca_req vaca
    inner join tb_userinfo user on user.loginID = vaca.loginID
    where vaca.loginID = #{loginID}
</select>
```

//개인휴가 리스트 불러주기

function fn_empVcpList(pagenum) {

IDCheck = false;

pagenum = pagenum || 1;

```
var param = {
    loginID : loginID
    , pageSize : pageSize
    , pageBlockSize : pageBlockSize
    , pagenum : pagenum
}
```

```
var listCallBack = function(returnvalue) {
    console.log(JSON.stringify(returnvalue));
```

```
console.log(returnvalue);
$("#listEmpVcp").empty().append(returnvalue);
```

var totalcnt = \$("#totalcnt").val();

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

```
var paginationHtml = getPaginationHtml(pagenum, totalcnt, pageSize, pageBlockSize, 'fn_empVcpList');
console.log("paginationHtml : " + paginationHtml);
```

\$("#empVcpPagination").empty().append(paginationHtml);

\$("#pageno").val(pagenum);

}

callAjax("/empVcp/vacaPersonalList.do", "post", "text", false, param, listCallBack);

}

▶ 휴가신청 및 연차조회 리스트
개인의 휴가신청 리스트가 조회됨

인사

휴가신청 및 연차조회

휴가 / 인사 / 휴가 신청 및 연차 조회

휴가 신청 및 연차 조회

총연차 19 잔여연차 15

휴가번호	사번	이름	휴가신청일	휴가시작일	휴가종료일	비상연락처	결재자
41	admin	관리자	2023-07-10	2023-07-26	2023-07-28	010-5555-5555	-
34	admin	관리자	2023-07-07	2023-07-07	2023-07-10	010-5635-6863	201601003
33	admin	관리자	2023-07-07	2023-07-13	2023-07-15	010-5635-6863	201601004

// 총휴가와 휴가잔여 표기

```
function fn_vacaRemain () {
    var param = {
        loginID : loginID
    }

    var vacaRemainCallback = function (reval) {
        if(reval.length > 0) {
            $("#vacaTot").empty().append(reval[0].dt_va_tot);
            $("#vacaRemain").empty().append(reval[0].dt_vaca_rest);
        } else {
            $("#vacaTot").empty().append("0");
            $("#vacaRemain").empty().append("0");
        }
    }

    callAjax("/empVcp/vacaRemain.do", "post", "json", false, param, vacaRemainCallback) ;
}
```

```
<!-- 잔여연차조회 -->
<select id="vacaRemain" resultType="kr.happyjob.study.empVcp.model.VacaPersonalRemainModel">
    select dt_vaca_rest, dt_va_tot from tb_dt_vacation where loginID = replace("#{loginID}"," ','')
</select>
```

▶ 휴가신청 및 연차조회 잔여연차
개인의 올해 총 연차와 잔여연차가 표기됨

인사

휴가신청 및 연차조회

휴 / 인사 / 휴가신청 및 연차조회

휴가신청 및 연차조회

총연차 19 잔여

휴가번호 사번

41 admin

34 admin

휴가신청

휴가신청

▶ 휴가신청 및 연차조회에 글자수 제한 알림
총 200자로 제한을 두며 글자를 입력할 때 마다 숫자가 줄어들고 200자를 초과하면 경고 글씨 생성

```
//textarea 글자수 표기
function fn_textArea() {
    var totText = 200;
    //남은 글자 수
    var remainText;
    //현재 작성한 글자 수
    var vacaResonLength = $("#form_vaca_reson").val().length;
    //현재 남은 글자 수
    remainText = totText - vacaResonLength;

    if(remainText < 0 ) {
        $("#textAreaRemain").html("글자 수 초과").css("color","red");
        return;
    }

    $("#textAreaRemain").html(remainText + "/" + totText).css("color","black");
}
```

사번	admin	비상연락처	ex) 010-0000-0000
휴가시작일	연도-월-일	휴가종료일	연도-월-일
휴가사유	<div>200자 이내로 작성가능</div> <div>200/200</div>		

신청

취소

▶ 휴가신청 및 연차조회에 휴가신청 데이터유효성 체크 함수
정확한 데이터 insert를 위해 데이터유효성 체크 함수가 먼저 실행
시작날짜가 종료날짜보다 느린 상황 방지, 빈 항목 방지, 글자 수 초과 방지

```
//data유효성 체크
function fn_data_validation() {
    var check=true;
    var form_vaca_tel = $("#form_vaca_tel").val();
    var form_vaca_sdate = $("#form_vaca_sdate").val();
    var form_vaca_edate = $("#form_vaca_edate").val();
    var form_vaca_reson = $("#form_vaca_reson").val();
    var valiList = [form_vaca_tel, form_vaca_sdate, form_vaca_edate, form_vaca_reson];
    for(var i in valiList) {
        console.log(valiList[i]);
        if(valiList[i] == null || valiList[i] == "") {
            check = false;
            break;
        }
    }
    console.log(check);
    //빈항목체크
    if(check == false) {
        alert("모든 항목을 채워주세요");
        return;
    }
    //휴가사유 글자제한
    if(form_vaca_reson.length > 200) {
        alert("휴가사유를 200자 이내로 적어주세요");
        return;
    }
    //날짜 체크
    form_vaca_sdate = parseInt(form_vaca_sdate.replace(/-/g, ""));
    form_vaca_edate = parseInt(form_vaca_edate.replace(/-/g, ""));
    console.log(form_vaca_sdate);

    if(form_vaca_sdate > form_vaca_edate){
        alert("휴가 시작 날짜와 종료 날짜를 정확히 입력해주세요");
        return;
    }
    fn_vacaFormInsert();
}
```

인사

휴가신청 및 연차조회

유효성체크 함수



▶ 휴가신청 및 연차조회에 휴가신청 함수
데이터 유효성 체크 함수를 거쳐 정확한 데이터가 컨트롤러 단으로 넘어감

```
//휴가신청폼 insert
function fn_vacaFormInsert(){

    var param = {
        appro_type_cd : $("#appro_type_cd").val(),
        form_loginID : $("#form_loginID").val(),
        form_vaca_tel : $("#form_vaca_tel").val(),
        form_vaca_sdate : $("#form_vaca_sdate").val(),
        form_vaca_edate : $("#form_vaca_edate").val(),
        form_vaca_reson : $("#form_vaca_reson").val()
    }

    var vacaInsertCallback = function (reval) {
        console.log(JSON.stringify(reval));
        if(reval.vacaApp > 0) {
            if(reval.approIn > 0) {
                if(reval.vacaApproUpdate > 0) {
                    alert("신청되었습니다");
                    gfCloseModal();
                }else {
                    alert("결재번호등록오류");
                }
            }else {
                alert("결재등록오류");
            }
        }else {
            alert("신청오류");
        }
        fn_empVcpList();
    }

    callAjax("/empVcp/vacaInsertvcp.do", "post", "json", false, param, vacaInsertCallback) ;
}
```

인사

휴가신청 및 연차조회

```

<!-- 휴가테이블에 insert -->
<insert id="vacaApplication">
  <selectKey resultType="int" keyProperty="vaca_no" order="BEFORE">
    select ifnull(max(vaca_no),0) + 1 from tb_vaca_req
  </selectKey>

  insert into tb_vaca_req
  (
    vaca_no
    ,loginID
    ,vaca_tel
    ,vaca_sdate
    ,vaca_edate
    ,vaca_reason
    ,vaca_req_date
  ) values (
    #{vaca_no}
    ,replace("#{form_loginID}',' ','')
    ,replace("#{form_vaca_tel}',' ','')
    ,#{form_vaca_sdate}
    ,#{form_vaca_edate}
    ,#{form_vaca_reson}
    ,DATE_FORMAT(now(), '%Y-%m-%d')
  )
</insert>

```

▶ 휴가신청 및 연차조회 휴가신청 쿼리
 휴가테이블에 휴가내역insert -> 결재테이블에 방금 휴가 신청한거 insert ->
 방금 결재테이블에 insert된 휴가내역의 결재번호를 휴가테이블에 update

```

<!-- 결재테이블에 insert -->
<insert id="approIn">
  <selectKey resultType="int" keyProperty="appro_no" order="BEFORE">
    select ifnull(max(appro_no),0)+1 from tb_appro
  </selectKey>
  insert into tb_appro
  (
    appro_no
    ,loginID
    ,appro_type_cd
    ,vaca_no
    ,appro_req_date
    ,dept_cd
  ) values (
    #{appro_no}
    ,#{form_loginID}
    ,#{appro_type_cd}
    ,(select MAX(vaca_no) from tb_vaca_req)
    ,(select vaca_req_date from tb_vaca_req where vaca_no = (select MAX(vaca_no) from tb_vaca_req))
    ,(select dept_cd from tb_userinfo where loginID = replace("#{form_loginID}',' ',''))
  )
</insert>

<!-- 결재테이블에 insert된 결재번호 휴가 테이블에 update -->
<update id="vacaApproUpdate">
  update tb_vaca_req set appro_no = (select appro_no from tb_appro where vaca_no =
  |(select vaca_no from (select MAX(vaca_no) as vaca_no from tb_vaca_req) vaca))
  where vaca_no = (select vaca_no from (select MAX(vaca_no) as vaca_no from tb_vaca_req) vaca)
</update>

```

#주요코드

#문제해결

ERP
System

인사

휴가신청내역
(empVcs)

인사

휴가신청내역

☞ / 인사 / 휴가신청내역

휴가신청 내역

신청일 연도-월-일 - 연도-월-일 사번

검색

휴가번호	사번	이름	휴가신청일	휴가시작일	휴가종료일	비상연락처
41	admin	관리자	2023-07-10	2023-07-26	2023-07-28	010-5555-5555
40	202307001	신성민	2023-07-07	2023-07-17	2023-07-20	010-5555-5555
39	202306068	최은서	2023-07-07	2023-08-16	2023-08-25	010-5555-5555
38	202306068	최은서	2023-07-07	2023-06-26	2023-07-14	010-5555-5555
37	202306066	서현진	2023-07-07	2023-08-09	2023-08-26	010-5555-5555
36	202306065	권예진	2023-07-07	2023-08-04	2023-08-05	010-5555-5555
35	202306065	권예진	2023-07-07	2023-07-05	2023-07-07	010-5555-5555
34	admin	관리자	2023-07-07	2023-07-07	2023-07-10	010-5635-6863
33	admin	관리자	2023-07-07	2023-07-13	2023-07-15	010-5635-6863

<< < 1 > >>

```
//휴가신청리스트 뿌려주기
function fn_empVcsList(pagenum) {

    IDCheck = false;

    pagenum = pagenum || 1;

    var param = {
        vaca_req_sdate : $("#vaca_req_sdate").val()
        , vaca_req_edate : $("#vaca_req_edate").val()
        , loginID : $("#loginID").val()
        , pageSize : pageSize
        , pageBlockSize : pageBlockSize
        , pagenum : pagenum
    }

    var listCallBack = function(returnvalue) {
        console.log(JSON.stringify(returnvalue));

        console.log(returnvalue);
        $("#listEmpVcs").empty().append(returnvalue);

        var totalcnt = $("#totalcnt").val();

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

        var paginationHtml = getPaginationHtml(pagenum, totalcnt, pageSize, pageBlockSize, 'fn_empVcsList');
        console.log("paginationHtml : " + paginationHtml);

        $("#empVcsPagination").empty().append( paginationHtml );

        $("#pageno").val(pagenum);
    }

    callAjax("/empVcs/vacaSearchList.do", "post", "text", false, param, listCallBack) ;

}
```

▶ 휴가신청내역 리스트 조회 함수
휴가를 신청한 모든 사원의 신청 내역을 조회 할 수 있다.

인사

휴가신청내역

```

<!-- 휴가신청 목록 리스트 조회 -->
<select id="empVcsList" resultType="kr.happyjob.study.empVcs.model.VacaSearchModel">
    select vaca.vaca_no
        ,vaca.loginID
        ,user.name
        ,vaca.vaca_req_date
        ,vaca.vaca_sdate
        ,vaca.vaca_edate
        ,vaca.vaca_tel
        ,vaca.appro_bos
        ,vaca.appro_yn
        ,vaca.vaca_rej
    from tb_vaca_req vaca
        inner join tb_userinfo user on user.loginID = vaca.loginID
    <where>
    <if test="(vaca_req_sdate != null) and (!vaca_req_sdate.equals(''))">
        and STR_TO_DATE(vaca.vaca_req_date,'%Y-%m-%d') >= STR_TO_DATE("#{vaca_req_sdate}','%Y-%m-%d')
    </if>
    <if test="(vaca_req_edate != null) and (!vaca_req_edate.equals(''))">
        and STR_TO_DATE("#{vaca_req_edate}','%Y-%m-%d') >= STR_TO_DATE(vaca.vaca_req_date,'%Y-%m-%d')
    </if>

    <if test="(loginID != null) and (!loginID.equals(''))">
        and vaca.loginID Like CONCAT('%',replace("#{loginID}',' ','') , '%')
    </if>
    </where>
    order by vaca.vaca_no desc
    LIMIT #{pageIndex}, #{pageSize}
</select>

```

▶ 휴가신청내역 리스트 조회 쿼리

▶ 휴가리스트 count 쿼리
리스트의 총 개수를 구한다. Pagination메소드를 실행하기 위해 필요

```

<select id="countEmpVcsList" resultType="int">
    select count(*)
    from tb_vaca_req vaca
        inner join tb_userinfo user on user.loginID = vaca.loginID
    <where>
    <if test="(vaca_req_sdate != null) and (!vaca_req_sdate.equals(''))">
        and STR_TO_DATE(vaca.vaca_req_date,'%Y-%m-%d') >= STR_TO_DATE("#{vaca_req_sdate}','%Y-%m-%d')
    </if>
    <if test="(vaca_req_edate != null) and (!vaca_req_edate.equals(''))">
        and STR_TO_DATE("#{vaca_req_edate}','%Y-%m-%d') >= STR_TO_DATE(vaca.vaca_req_date,'%Y-%m-%d')
    </if>

    <if test="(loginID != null) and (!loginID.equals(''))">
        and vaca.loginID Like CONCAT('%',replace("#{loginID}',' ','') , '%')
    </if>
    </where>
</select>

```

#주요코드

#문제해결

ERP
System

인사
급여관리
(empSam)

인사

급여관리

인사 / 급여관리

급여관리

----년 --월 □ 이번달 급여계산

부서 ▼ 직급 ▼ 전체 ▼ 전체 ▼ 검색

장년월	사번	사원명	부서명	직급	연봉	기본급	국민연금	건강보험	산재보험	고용보험	실급여	퇴직금	비고	알림지급
23-06	202307002	이하나	회계팀	부장	50,000,000	4,166,667	65,000	33,333	142,917	187,500	3,737,917	0	-	지급
23-06	202306067	김수환	영업팀	사원	30,000,000	2,500,000	39,000	20,000	85,750	112,500	2,242,750	0	-	미지급
23-06	202306065	권예진	인사팀	부장	50,000,000	4,166,667	65,000	33,333	142,917	187,500	3,737,917	0	-	미지급
23-06	202307001	산성민	인사팀	과장	45,000,000	3,750,000	58,500	30,000	128,625	168,750	3,364,125	0	-	미지급
23-06	202307004	윤주희	회계팀	상무	55,000,000	4,583,333	71,500	36,667	157,208	206,250	4,111,708	0	-	미지급
23-06	202307003	송태호	회계팀	주임	35,000,000	2,916,667	45,500	23,333	100,042	131,250	2,616,542	0	-	미지급
23-06	202306064	이서연	임원팀	전무	60,000,000	5,000,000	78,000	40,000	171,500	225,000	4,485,500	0	-	미지급
23-06	202306068	최은서	영업팀	대리	40,000,000	3,333,333	52,000	26,667	114,333	150,000	2,990,333	0	-	미지급
23-06	202306063	배우비	임원팀	대표	70,000,000	5,833,333	91,000	46,667	200,083	262,500	5,233,083	0	-	미지급
23-06	202306066	서현진	영업팀	대리	40,000,000	3,333,333	52,000	26,667	114,333	150,000	2,990,333	0	-	미지급

« < 1 2 > »

▶ 급여계산전확인 쿼리

```

<!-- 최근 급여 확인/급여계산전 확인 -->
<select id="calSalChk" resultType="Integer">
    select CAST(CONCAT(SUBSTRING(sal.sal_date, 1, 4),SUBSTRING(sal.sal_date, 6, 2)) AS UNSIGNED) as sal_date
    from (select sal_date from tb_sal order by sal_date desc) sal
    group by sal.sal_date
</select>

```

```

//급여계산전 확인
function fn_calSalChk() {
    if($("#calDate").val() != null && $("#calDate").val() != '') {
        var param = {};
        var calSalChkCallback = function (reval) {

            //입력한 달 숫자로 변경
            var calDate = $("#calDate").val();
            calDate = Number(calDate.substring(0,4) + calDate.substring(5,7));

            console.log(reval);

            //db에 저장되어있는 이미 급여계산이 끝난달과 입력한 달 비교
            var check = true;

            for(var i in reval) {
                var a = reval[i];
                console.log(a);
                if(a == calDate){
                    alert("이미 계산된 목록 입니다.");
                    check = false;
                    break;
                }
            }

            if(check) { //입력한 달이 db에 없으면 급여 계산 함수 실행
                fn_calSal();
            }

            callAjax("/empSam/calSalChk.do", "post", "json", false, param, calSalChkCallback);
        }else {
            var calSalHtml = "년도랑 월을 선택해주세요";
            $("#calSalSpan").empty().append( calSalHtml );
        }
    }
}

```

▶ 급여계산전확인 함수

중복계산을 피하기 위해 급여 테이블에 계산된 내역이 있는지 확인한다

인사

급여관리

```
//급여계산(tb_emp 테이블에서 조인해오기 지급여부는 N로 해서 insert)
function fn_calSal() {

    var param = {
        calDate : $("#calDate").val()
    }
    var calSalCallback = function(reval) {

        if(reval>0) {
            //리스트 다시 뿌려주기
            fn_empSamList();
        }else {
            alert("급여계산 실패");
        }
    }

    callAjax("/empSam/calSal.do", "post", "json", false, param, calSalCallback) ;
}
```

▶급여계산 함수

급여계산전 확인 함수를 거쳐서 급여가 계산된 내역이 없다면 실행

```
<!-- 급여계산 -->
<insert id="calSal">
    insert into tb_sal(
        loginID
        ,sal_date
        ,sal_kun_com
        ,sal_kun_emp
        ,sal_ko_com
        ,sal_ko_emp
        ,sal_san_com
        ,sal_san_emp
        ,sal_kuk_com
        ,sal_kuk_emp
        ,sal_pre
        ,sal_after
    ) select loginID
        ,#{calDate}
        ,ROUND((sal_kun/2),2) as sal_kun_com
        ,ROUND((sal_kun/2),2) as sal_kun_emp
        ,ROUND((sal_ko/2),2) as sal_ko_com
        ,ROUND((sal_ko/2),2) as sal_ko_emp
        ,ROUND((sal_san/2),2) as sal_san_com
        ,ROUND((sal_san/2),2) as sal_san_emp
        ,ROUND((sal_kuk/2),2) as sal_kuk_com
        ,ROUND((sal_kuk/2),2) as sal_kuk_emp
        ,sal_pre
        ,sal_after
    from tb_userinfo
        where emp_work_yn = 'Y'|
</insert>
```

▶급여계산 쿼리

계산식을 작성하여 급여 테이블에 insert해준다

인사

급여관리

일괄지급

지급

미지급

미지급

미지급

미지급

미지급

미지급

미지급

미지급

미지급

```
//일괄지급 (지급여부 N인것들만 지급하기 (N>Y로 변경))
function fn_salPayAll() {
    var param= {
        dept_cd : $("#dept_cd").val()
        , level_cd : $("#level_cd").val()
        , sal_pay_yn : $("#sal_pay_yn").val()
        , emp_info : $("#emp_info").val()
        , sname : $("#sname").val()
    }

    var payAllCallback = function(reval) {
        console.log( JSON.stringify(reval) );

        if(reval.salPaysf > 0) {
            alert("일괄 지급되었습니다.");
            fn_empSamList(currentPage);
        }else {
            alert("이미 모두 지급되었습니다.");
            gfCloseModal();
        }
    }

    callAjax("/empSam/salPayAll.do", "post", "json", false, param, payAllCallback);
}
```

▶ 급여일괄지급 함수
이번달 급여만 일괄지급 된다. 검색조건이 걸려있으면 그 조건에 걸려져 나온 사원들만 계산된다.

▶ 급여일괄지급 쿼리.

```
<!-- 일괄지급 쿼리 -->
<update id="salPayAll">
/*kr.happyjob.study.empSam.dao.EmpSamDao.salPayAll*/
update tb_userinfo user
inner join tb_sal sal on user.loginID = sal.loginID
set sal.sal_pay_yn = 'Y'
<where>
    <if test="(dept_cd != null) and (!dept_cd.equals(''))">
        and user.dept_cd = #{dept_cd}
    </if>
    <if test="(level_cd != null) and (!level_cd.equals(''))">
        and user.level_cd = #{level_cd}
    </if>
    <if test="(sal_pay_yn != null) and (!sal_pay_yn.equals(''))">
        and sal.sal_pay_yn = #{sal_pay_yn}
    </if>
    <if test="(sname != null) and (!sname.equals(''))">
        <choose>
            <when
                test="emp_info eq 'loginID'.toString()"
                and user.loginID LIKE CONCAT('%', replace(#{sname}, ' ', ''), '%')
            </when>
            <when
                test="emp_info eq 'emp_name'.toString()"
                and user.name LIKE CONCAT('%', replace(#{sname}, ' ', ''), '%')
            </when>
        </choose>
    </if>
    and sal.sal_date LIKE CONCAT('%', DATE_FORMAT(now(), '%Y-%m'), '%')
    and sal.sal_pay_yn = 'N'
</where>
</update>
```

인사

급여관리

입원지급

지급

미지급

미지급

미지급

미지급

미지급

미지급

미지급

미지급

미지급

```
//개별지급
function fn_salPayOne(userInfo) { //사번이 날라옴
    console.log(userInfo)
    var param = {
        loginID : userInfo,
        sal_date : sal_date
    }

    var payOneCallback = function(reval) {
        console.log( JSON.stringify(reval) );

        if(reval.salPaysf > 0) {
            alert("지급되었습니다");
            gfCloseModal();
            fn_empSamList(currentPage);
        } else {
            alert("지급 실패하였습니다.");
        }
    }

    callAjax("/empSam/salPayOne.do", "post", "json", false, param , payOneCallback) ;
}
```

▶ 개별지급 쿼리

```
<!-- 개별지급쿼리 -->
<update id="salPayOne">
    update tb_sal
        set sal_pay_yn = 'Y'
        where loginID = #{loginID} and sal_date = #{sal_date}
</update>
```

▶ 개별지급 함수

누른 버튼에 해당하는 행 정보만 급여가 계산이 됨

#주요코드

#문제해결

ERP
System

인사

급여조회
(empSas)

인사

급여조회

☞ / 인사 / 급여조회

급여조회

사번 202307004 2023년 07월 검색

공제항목	회사부담금	개인부담금	지급항목
국민연금	103,125	103,125	기본급
건강보험	78,604	78,604	실급여
고용보험	18,334	18,334	연봉
산재보험	35,750	35,750	

보험료율(근로자부담)
 국민연금: 4.5%
 건강보험: 3.43%
 고용보험: 0.8%
 산재보험: 1.56%

```
//보험료를 팝업 띄우기
$('#taxPer').hover(function() {
    $('#taxPopup').css("visibility", "visible");
}, function() {
    $('#taxPopup').css("visibility", "hidden");
});
```

▶ 보험료를 팝업 띄우기 함수
 hover를 이용하여 마우스를 올리면 보험료를 팝업창이 보임

```
//개인급여내역서 리스트 뿌려주기1
function fn_empSasList1() {
    var searchLoginID = $("#searchLoginID").val();
    var searchDate = $("#searchDate").val();

    if(searchLoginID != null && searchLoginID != "" &&
        searchDate != null && searchDate != "") {
        var param = {
            searchLoginID : searchLoginID
            , searchDate : searchDate
        }

        $("#empSasTax").css("visibility", "visible");

        var listCallBack = function(returnvalue) {
            console.log(JSON.stringify(returnvalue));

            //salSerchGrd append
            $("#listEmpSas1").empty().append(returnvalue);
        }
        callAjax("/empSas/salSerarchList1.do", "post", "text", false, param, listCallBack);
    } else {
        alert("사번이랑 날짜를 입력해 주세요");
    }
}
```

▶ 급여지급내역(개인)함수
 사번은 자동으로 들어가며
 조회하고 싶은 달을 선택 시
 그 달에 해당하는
 급여와 보험료를 조회할 수 있다.

```
//개인급여내역서 리스트 뿌려주기2
function fn_empSasList2() {
    var searchLoginID = $("#searchLoginID").val();
    var searchDate = $("#searchDate").val();

    if(searchLoginID != null && searchLoginID != "" &&
        searchDate != null && searchDate != "") {
        var param = {
            searchLoginID : searchLoginID
            , searchDate : searchDate
        }

        $("#empSasResult").css("visibility", "visible");

        var listCallBack = function(returnvalue) {
            console.log(JSON.stringify(returnvalue));

            //salSerchGrd append
            $("#listEmpSas2").empty().append(returnvalue);
        }
        callAjax("/empSas/salSerarchList2.do", "post", "text", false, param, listCallBack);
    } else {
        alert("사번이랑 날짜를 입력해 주세요");
    }
}
```


인사

급여조회

```

<!-- 목록 조회 -->
<select id="empSasList1" resultType="kr.happyjob.study.empSas.model.SalSerarchModel">
    /*kr.happyjob.study.empSas.dao.EmpSasDao*/
    select sal.sal_kun_com
           ,sal.sal_kun_emp
           ,sal.sal_ko_com
           ,sal.sal_ko_emp
           ,sal.sal_san_com
           ,sal.sal_san_emp
           ,sal.sal_kuk_com
           ,sal.sal_kuk_emp

    from tb_sal sal
    where sal.loginID = replace("#{searchLoginID}, ' ','') and sal.sal_date LIKE CONCAT('%', #{searchDate} , '%')

</select>

<!-- 목록 조회 -->
<select id="empSasList2" resultType="kr.happyjob.study.empSas.model.SalSerarchModel">
    /*kr.happyjob.study.empSas.dao.EmpSasDao*/
    select
        sal.sal_pre
        ,sal.sal_after
        ,user.emp_yr_sal
    from tb_sal sal
        inner join tb_userinfo user on user.loginID = sal.loginID
    where sal.loginID = replace("#{searchLoginID}, ' ','') and sal.sal_date LIKE CONCAT('%', #{searchDate} , '%')

</select>

```

▶ 급여지급내역(개인) 쿼리

#주요코드

#문제해결

ERP
System

문제해결

문제1. 문자열형식의 날짜를 매개변수로 받으니 2023-12라는 매개변수가 2011로 출력됨.

해결: 구글에 검색하여 여러 개의 문자열 추출 함수(slice,charAt...) 중 원하는 범위를 지정하여 추출 할 수 있는 'substring' 함수를 사용하여 매개변수로 받을 때 숫자 부분만 추출 -> 다시 'substring' 함수로 년과 월을 분리 후 '-' 와 합쳐서 화면에 출력

```
href="javascript:fn_openpopup({list.loginID}, ${list.sal_date.substring(0,4).concat(list.sal_date.substring(5))});"
```

```
function fn_openpopup(info, date) { //사번 이 날라온다
    console.log(info);
    userInfo=info;
    date=date.toString();
    sal_date=date.substring(0,4)+'-'+date.substring(4);

    // 모달 팝업
    gfModalPop("#layer1");
}
function fn_openpopup2() {

    // 모달 팝업
    gfModalPop("#layer2");
}
```

문제2. 휴가 신청 시 휴가 시작날짜가 종료날짜보다 느려도 휴가신청 함수가 실행됨

해결: 데이터 유효성 체크 함수에 휴가날짜 검사 코드 추가 - 정확한 비교를 위해 날짜를 '-'를 빼주어 숫자형으로 변환하여 비교

```
//날짜 체크
form_vaca_sdate = parseInt(form_vaca_sdate.replace(/-/g, ""));
form_vaca_edate = parseInt(form_vaca_edate.replace(/-/g, ""));
console.log(form_vaca_sdate);

if(form_vaca_sdate > form_vaca_edate){
    alert("휴가 시작 날짜와 종료 날짜를 정확히 입력해주세요");
    return;
}
```

문제3. update문 작성시 조건에 update할 테이블과 같은 테이블을 줄 수 없음.

해결: 서브 쿼리를 이용하여 가상테이블을 생성해준 뒤 alias(별칭) 지정하여 그 가상테이블을 사용하여 조건을 작성

```
<update id="vacaApproUpdate">
    update tb_vaca_req set appro_no = (select appro_no from tb_appro where vaca_no =
    (select vaca_no from (select MAX(vaca_no) as vaca_no from tb_vaca_req) vaca))
    where vaca_no = (select vaca_no from (select MAX(vaca_no) as vaca_no from tb_vaca_req) vaca)
</update>
```

문제4. 테이블의 컬럼 도메인이 **int(15)**인데 데이터를 **insert**할때 지정된 컬럼길이를 초과해서 **insert** 오류발생

해결: 데이터 유효성 체크 함수에 숫자 범위를 제한하는 코드 작성

```
//발주수량제한
var maxNum = 999;
var minNum = 1;
if(dlv_amt > maxNum || minNum > dlv_amt) {
    alert("발주수량초과. 3자리로 입력해주세요");
    return;
}
```

문제5. 급여조회 페이지에서 본인이 아닌 다른 사람의 사번을 입력해도 조회가 가능(연봉정보 유출 가능성)

해결: **controller**단에서 세션값으로 등록되어 있는 로그인 된 사원의 사번을 가져와
jsp단에서 **input**태그에 **value**값으로 입력. 태그는 **readonly**로 수정불가능

controller

```
String loginID = (String)session.getAttribute("loginId");
model.addAttribute("loginID", loginID);
```

jsp

```
<input type="text" style="height:30px; width:100px; margin-right:5px;" id="searchLoginID" name="searchLoginID" value="${loginID}" readonly>
```

문제6. 신청 폼 작성 후 폼의 내용이 초기화가 안되고 남아있음

해결: 폼 초기화 함수를 작성하여 폼 화면을 열 때 초기화 함수 먼저 실행

```
//모달 초기화
function popupinit() {
    $("#ltypecombo").val("");
    $("#mtypecombo").val("");
    $("#ptypecombo").val("");
    $("#dlv_amt").val("");
}
```

```
//발주신청모달창 오픈
function fn_openpopup1() {
    //모달 초기화
    popupinit();
    // 모달 팝업
    gfModalPop("#dlvForm");
}
```

문제7. 신청 폼 작성 시 빈 항목이 있음에도 신청이 됨

해결: 데이터 유효성 체크 함수를 작성하여 공백이 있을 시 컨트롤러가 실행되지 않고 alert로 경고

```
//data유효성 체크
function fn_data_validation() {
    //유효성 체크 변수
    var check=true;

    var valiList = [];
    var ltypecombo = $("#ltypecombo").val();
    var mtypecombo = $("#mtypecombo").val();
    var ptypecombo = $("#ptypecombo").val();
    var dlv_amt = $("#dlv_amt").val();

    valiList.push(ltypecombo, mtypecombo, ptypecombo, dlv_amt);

    for(var i in valiList) {
        console.log(valiList[i]);
        if(valiList[i] == null || valiList[i] == "") {
            check = false;
            break;
        }
    }
    console.log(check);
    if(check == false) {
        alert("모든 항목을 채워주세요");
        return;
    }
}
```