

动态表格快速实现统计和非实体字段拼接



一、销售订单界面效果案例

销售订单 > 销售订单

新增 已选0项

高级搜索

#	订单编号	业务类型	订单类型	状态	数量	金额	交货日期	客户名称	联系人	联系电话	联系地址	操作
1	SO-20240409-0073	标准销售	销售订单	未审核	0	0	2024-04-09 00:00:00					编辑 删除 操作
2	SO-20240407-0072	标准销售	销售变更单	已审核	0	0	2024-04-23 00:00:00	张一				编辑 删除 操作
3	SO-20240407-0071	标准销售	销售变更单	已审核	0	0	2024-04-23 00:00:00	张一				编辑 删除 操作
4	SO-20240402-0069	标准销售	销售订单	已下达	20	0	2024-04-23 00:00:00	张一				编辑 删除 操作
5	SO-20240402-0064	标准销售	销售订单	已关闭	110	0	2024-04-30 00:00:00	张一				编辑 删除 操作
6	SO-20240402-0065	标准销售	销售订单	待发货	10	0	2024-04-29 00:00:00	王二				编辑 删除 操作
7	SO-20240402-0068	标准销售	销售订单	未审核	30	0	2024-04-30 00:00:00	王二				编辑 删除 操作
8	SO-20240401-0060	标准销售	销售订单	已下达	30	1800	2024-04-30 00:00:00	张一				编辑 删除 操作
9	SO-20240401-0063	标准销售	销售变更单	已审核	0	0	2024-04-30 00:00:00					编辑 删除 操作
10	SO-20240401-0062	标准销售	销售变更单	已审核	0	0	2024-04-30 00:00:00					编辑 删除 操作
11	SO-20240401-0061	标准销售	销售订单	已审核	0	0	2024-04-30 00:00:00					编辑 删除 操作
12	SO-20240328-0054	标准销售	销售订单	待发货	20	30000	2024-03-31 00:00:00	张一	曲奕	13454454334	武汉市洪山区	编辑 删除 操作
13	SO-20240326-0041	标准销售	销售订单	待发货	40	16	2024-04-30 00:00:00					编辑 删除 操作
14	SO-20240329-0059	标准销售	销售订单	未审核	0	0	2024-03-19 00:00:00					编辑 删除 操作
15	SO-20240329-0058	标准销售	销售订单	未审核	0	0	2024-03-31 00:00:00					编辑 删除 操作

销售订单 > 销售订单 > 销售订单详情

操作

基础信息

创建人	创建时间	订单编号	订单类型
cd	2024-04-02 06:06:18	SO-20240402-0069	销售订单
状态	交货日期	客户名称	联系人
已下达	2024-04-23 00:00:00	张一	-
联系电话	联系地址	销售人员	合同名称
-	-	-	-
计划数量	备注		
20	-		

物料明细 生产订单 采购订单 发货记录 进度跟踪

#	明行号	产品物料	产品名称	生产方式	数量	单位	单价	金额	备注	创建人	创建时间	修改人	修改时间
1	1	1.03.01.04.00019	立体塑料袋		10		55			cd	2024-04-02 06:06:18	eric	2024-04-02 09:04:34
2	2	1.03.01.01.00386	鞋盒式纸箱	采购	10		101			cd	2024-04-02 06:06:18	eric	2024-04-02 09:04:34

销售进度跟踪 > 交货进度跟踪

高级搜索

#	订单编号	订单类型	客户名称	交货日期	状态	交货状态	销售数量	交货数量	交货完成率	计划生产数量	生产完工数量	生产完成率	计划采购数量
1	SO-20240326-0041	销售订单		2024-04-30 0...	待发货	正常	40	42	NaN%	5	0	0%	80
2	SO-20240401-0060	销售订单	张一	2024-04-30 0...	已下达	超期	30	15	NaN%	30	0	0%	0
3	SO-20240401-0061	销售订单		2024-04-30 0...	已审核	超期	0	0	NaN%	0	0	0%	0
4	SO-20240401-0062	销售变更单		2024-04-30 0...	已审核	超期	0	0	NaN%	0	0	0%	0
5	SO-20240401-0063	销售变更单		2024-04-30 0...	已审核	超期	0	0	NaN%	0	0	0%	0
6	SO-20240402-0068	销售订单	王二	2024-04-30 0...	未审核	超期	30	0	NaN%	0	0	0%	0
7	SO-20240402-0064	销售订单	张一	2024-04-30 0...	已关闭	正常	110	165	NaN%	10	0	0%	100
8	SO-20240402-0066	退货申请单	张一	2024-04-30 0...	未审核	超期	0	0	NaN%	0	0	0%	0
9	SO-20240402-0067	退货申请单	张一	2024-04-30 0...	已审核	超期	0	0	NaN%	0	0	0%	0
10	SO-20240402-0065	销售订单	王二	2024-04-29 0...	待发货	超期	10	5	NaN%	0	0	0%	10
11	SO-20240402-0070	退货申请单	张一	2024-04-28 0...	已审核	超期	0	0	NaN%	0	0	0%	0
12	SO-20240402-0069	销售订单	张一	2024-04-23 0...	已下达	正常	20	20	NaN%	10	0	0%	10
13	SO-20240407-0071	销售变更单	张一	2024-04-23 0...	已审核	超期	0	0	NaN%	0	0	0%	0
14	SO-20240407-0072	销售变更单	张一	2024-04-23 0...	已审核	超期	0	0	NaN%	0	0	0%	0
15	SO-20240409-0073	销售订单		2024-04-09 0...	未审核	超期	0	0	NaN%	0	0	0%	0



二、代码样例

1. 动态表格快速实现数量/金额统计

基于物料维度统计已发货数量

```

//屏蔽前端传的统计字段
SelectData selectData = getProcessData(PageDataUtil.removeColumns(pageData,
"finishQty"));
this.returnData = getSelectService().process(selectData);
GridDataMode gridData = (GridDataMode) returnData.getData();
List<Map<String, Object>> list = (List<Map<String, Object>>) gridData.getRows();
if (CollectionUtils.isNotEmpty(list)) {
    //交货信息统计
    ((ReturnDataAddSum) (dao, keySet) -> {
        String hq1 = "select new map(t.produCode as produCode,t.qty as finishQty)
from SaleDeliveryRecord t left join saleOrder so on t.saleCode = so.code where
t.isDelete = 0 and t.isActive = 0 and so.id = :orderId";
        List<Map<String, Object>> details =
dao.createQuery(hq1).setParameter("orderId",orderId).setReadOnly(true).list();
        return details;
    }).handle(getSelectService().getDao(), list, 0,"produCode", "finishQty");
}

```



2. 动态表格快速实现非实体字段拼接

基于订单维度统计数量、金额、已发货状态、是否延期交货

```

//屏蔽前端传的统计字段
SelectData selectData = getProcessData(PageDataUtil.removeColumns(pageData,
"qty","amount","finishQty","overdueState"));
returnData = process(selectData);
GridDataMode gridData = (GridDataMode) returnData.getData();
List<Map<String, Object>> list = (List<Map<String, Object>>) gridData.getRows();
if (CollectionUtils.isNotEmpty(list)) {
    //交货信息、交货状态统计
    ((ReturnDataAddSum) (dao, keySet) -> {
        String sql = "select o.gid as id,d.qty as qty,d.amount as
amount,r.finish_qty as \"finishQty\",case when o.due_date-date(r.operate_time)>=0
&& finish_qty-qty>=0 then 1 else 0 end as \"overdueState\" from sale_order o " +
        " left join (" +
        " select order_id,sum(qty) as qty,max(amount) as amount from
sale_order_detail where is_delete=0 and is_active=0 group by order_id" +
        " ) d on o.gid = d.order_id" +
        " left join (" +
        " select sale_code,sum(qty) as finish_qty,max(operate_time) as
operate_time from sale_delivery_record where is_delete=0 and is_active=0 group by
sale_code" +
        " ) r on o.code = r.sale_code where o.is_delete=0 and
o.is_active=0 and o.type = 10 and (o.gid in (:orderId))";
        sql = sql.replace(":orderId",
CommonMethodUtils.getInParameter(Lists.newArrayList(keySet), "o.gid"));
        List<Map<String, Object>> details =
dao.getCurrentSession().createSQLQuery(sql).setResultTransformer(Transformers.ALI
AS_TO_ENTITY_MAP).setReadOnly(true).list();
        return details;
    }).handle(dao, list, 0,"id", "qty","amount","finishQty","overdueState");
}

```



3. 动态表格快速实现比值计算

计算百分比

```
//屏蔽前端传的统计字段
SelectData selectData =
getProcessData(PageDataUtil.removeColumns(pageData, "purProgress"));
returnData = process(selectData);
GridDataMode gridData = (GridDataMode) returnData.getData();
List<Map<String, Object>> list = (List<Map<String, Object>>) gridData.getRows();
PageDataUtil.caculateRatio(list, "purProgress", "receivedQty", "demandQty");
```



三、工具类源码解析

1. 快速统计工具类

```
/**
 * Copyright (c) 2023, Chd9 All Rights Reserved.
 *
 * @ProjectName UniMax-A
 * @ClassName: PageDataSum
 * @Author: donnie
 * @Date: 2024/3/27 17:21
 * @Description: 页面增加汇总数据列
 * @Version: 1.8
 * @Since JDK 1.8
 */
public interface ReturnDataAddSum {
    /**
     * @methodName: handle
     * @Description: 核心计算方法
     * @paramType: [com.epichust.mestar.platform.dao.OperateDao,
     java.util.List<java.util.Map<java.lang.String, java.lang.Object>>,
     java.lang.String, java.lang.String...]
     * @param: [dao, list, joinKey, keys]
     * @return: void
     * @Author: donnie
     * @Date: 2024/3/27 18:42
     */
    default void handle(OperateDao dao, List<Map<String, Object>> list, String
    joinKey, String... keys){
        Set<String> keySet = PageDataUtil.getValuesForKey(list, joinKey);
        Map<String, JSONObject> joinMap = sum(getData(dao, keySet), joinKey,
    keys);
        PageDataUtil.addValueForKeys(list, joinMap, joinKey, keys);
    }

    /**
     * @methodName: handle
     * @Description: 核心计算方法
     * @paramType: [com.epichust.mestar.platform.dao.OperateDao,
     java.util.List<java.util.Map<java.lang.String, java.lang.Object>>,
     java.lang.Object, java.lang.String, java.lang.String...]
     * @param: [dao, list, nullFillValue, joinKey, keys]
     * @return: void
     * @Author: donnie
     */
}
```



```

    * @Date: 2024/3/30 0:21
    */
    default void handle(OperateDao dao, List<Map<String, Object>> list, Object
nullFillValue, String joinKey, String... keys) {
        Set<String> keySet = PageDataUtil.getValuesForKey(list, joinKey);
        Map<String, JSONObject> joinMap = sum(getData(dao, keySet), joinKey,
keys);
        PageDataUtil.addValueForKeys(list, joinMap, nullFillValue, joinKey,
keys);
    }
    /**
    * @methodName: getData
    * @Description: 查询数据接口
    * @paramType: [com.epichust.mestar.platform.dao.OperateDao,
java.util.Set<java.lang.String>]
    * @param: [dao, keySet]
    * @return: java.util.List<java.util.Map<java.lang.String,java.lang.Object>>
    * @Author: donnie
    * @Date: 2024/3/27 22:49
    */
    List<Map<String, Object>> getData(OperateDao dao, Set<String> keySet);
    /**
    * @methodName: sum
    * @Description: 数据汇总
    * @paramType: [com.epichust.mestar.platform.dao.OperateDao,
java.util.Set<java.lang.String>]
    * @param: [dao, keySet]
    * @return: java.util.Map<java.lang.String,com.alibaba.fastjson.JSONObject>
    * @Author: donnie
    * @Date: 2024/3/27 18:42
    */
    default Map<String, JSONObject> sum(List<Map<String, Object>> details, String
joinKey, String... keys) {
        Map<String, JSONObject> sumMap = new HashMap<>();
        for (Map<String, Object> detail : details) {
            String joinKeyValue = (String) detail.get(joinKey);
            JSONObject sum = sumMap.get(joinKeyValue);
            if (sum == null) {
                sum = new JSONObject();
                sumMap.put(joinKeyValue, sum);
            }
            for (int i = 0; i < keys.length; i++) {
                BigDecimal sumValue = (BigDecimal) sum.get(keys[i]);
                if (sumValue == null) {
                    sumValue = BigDecimal.ZERO;
                }
                BigDecimal newValue = BigDecimal.ZERO;
                if (null != detail.get(keys[i])) {
                    newValue = new
BigDecimal(String.valueOf(detail.get(keys[i])));
                }
                sumValue = sumValue.add(newValue);
                sum.put(keys[i], sumValue);
            }
        }
        return sumMap;
    }
}

```

```
}
```

2. 比值计算方法

```
/**
 * @methodName: removeColumns
 * @Description: 删除界面传过来实体不需要的列名
 * @paramType: [com.epichust.mestar.operation.controller.def.IPageData,
java.lang.String]
 * @param: [pageData, columnStr]
 * @return: com.epichust.mestar.operation.controller.def.IPageData
 * @Author: donnie
 * @Date: 2024/3/22 14:15
 */
public static IPageData removeColumns(IPageData pageData, @NotNull String...
columnStr) {
    String columns = pageData.getParams().get(PARAM_COLUMNS);
    List<JSONObject> l = (List) JSON.parse(columns);
    l = l.stream().filter(o ->
!Arrays.asList(columnStr).contains(o.getString("name"))).collect(Collectors.toList());
    pageData.getParams().put(PARAM_COLUMNS, JSON.toJSONString(l));
    return pageData;
}
/**
 * @methodName: caculateRatio
 * @Description: Map对象数组比值计算
 * @paramType: [java.util.List<java.util.Map < java.lang.String,
java.lang.Object>>, java.lang.String, java.lang.String, java.lang.String]
 * @param: [list, newColumn, key1, key2]
 * @return: void
 * @Author: donnie
 * @Date: 2024/3/27 22:39
 */
public static void caculateRatio(List<Map<String, Object>> list, String
newColumn, String key1, String key2) {
    for (Map<String, Object> map : list) {
        Object obj1 = map.get(key1);
        BigDecimal value1 = obj1 == null ? BigDecimal.ZERO : new
BigDecimal(String.valueOf(obj1));
        Object obj2 = map.get(key2);
        BigDecimal value2 = obj2 == null ? BigDecimal.ZERO : new
BigDecimal(String.valueOf(obj2));
        if (value2.compareTo(BigDecimal.ZERO) == 0) {
            map.put(newColumn, value1.equals(BigDecimal.ZERO) ? BigDecimal.ZERO :
BigDecimal.ONE);
        }else {
            map.put(newColumn, value1.divide(value2, 4, RoundingMode.HALF_UP));
        }
    }
}
/**
 * @methodName: caculateRatio
 * @Description: Map对象数组比值计算
 * @paramType: [java.util.List<java.util.Map < java.lang.String,
java.lang.Object>>, java.lang.String[], java.lang.String[], java.lang.String[]]
 * @param: [list, newColumn, key1, key2]
```



```

* @return: void
* @Author: donnie
* @Date: 2024/3/27 22:37
*/
public static void caculateRatio(List<Map<String, Object>> list, String[]
newColumn, String[] key1, String[] key2) {
    for (Map<String, Object> map : list) {
        for (int i = 0; i < newColumn.length; i++) {
            Object obj1 = map.get(key1);
            BigDecimal value1 = obj1 == null ? BigDecimal.ZERO : new
BigDecimal(String.valueOf(obj1));
            Object obj2 = map.get(key2);
            BigDecimal value2 = obj2 == null ? BigDecimal.ZERO : new
BigDecimal(String.valueOf(obj2));
            if (value2.compareTo(BigDecimal.ZERO) == 0) {
                map.put(newColumn[i], BigDecimal.ONE);
            }else {
                map.put(newColumn[i], value1.divide(value2, 4,
RoundingMode.HALF_UP));
            }
        }
    }
}
}

```