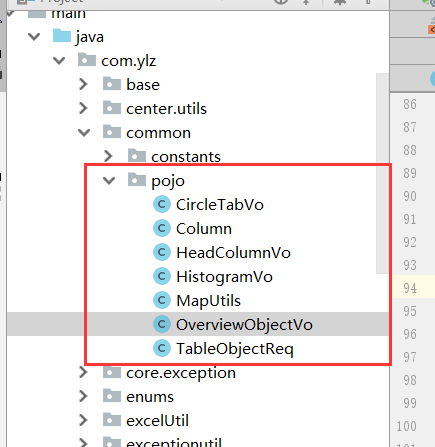
1. 使用到的相关类：
2. 存储列名对象，相关实体类：



**com.ylz.common.pojo.OverviewObjectVo最外层公式 实体类相关属性**

1. .同比公式属性:

com.ylz.common.pojo.OverviewObjectVo#isAnFlag=true

2).是否是重点字段(折线图返回字段设置为true):

com.ylz.common.pojo.OverviewObjectVo#important=true

3).自定义公式(不对公式做任何拼接)

com.ylz.common.pojo.OverviewObjectVo#customFormula=true

4).在图形返回的坐标(枚举类型)

com.ylz.common.pojo.OverviewObjectVo#coordinate

1. .饼图相关公式参数:

*//饼状图形***private boolean isXPercent** = **false**;*//是否第二列***private boolean isTotal** = **false**;*//是否总计***private boolean isTotalAn** = **false**;*//是否是总计同比***private** String **circleName**;*//饼图名称*

1. .自定义更多的参数类型,可以在这个上面给公用方法增加更多的类型参数(在不增加属性的基础上):

**private** Map<?, ?> **object**;

**com.ylz.common.pojo.Column 内层属性公式 实体类相关属性**

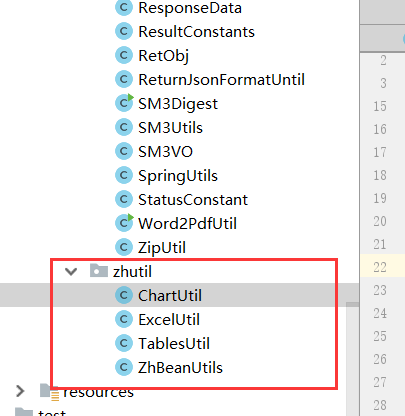
1. .自定义列名

com.ylz.common.pojo.Column#customFlag=true

**com.ylz.sxpt.req.FamilyDoctorInfoReq 接收前端参数,具体的使用方法比较简单,在实体类中就有比较清楚的注释;**

**com.ylz.common.pojo.TableObjectReq 定义表实体类 在实体类中就有比较清楚的注释;**

**2.存储表对象工具类：**



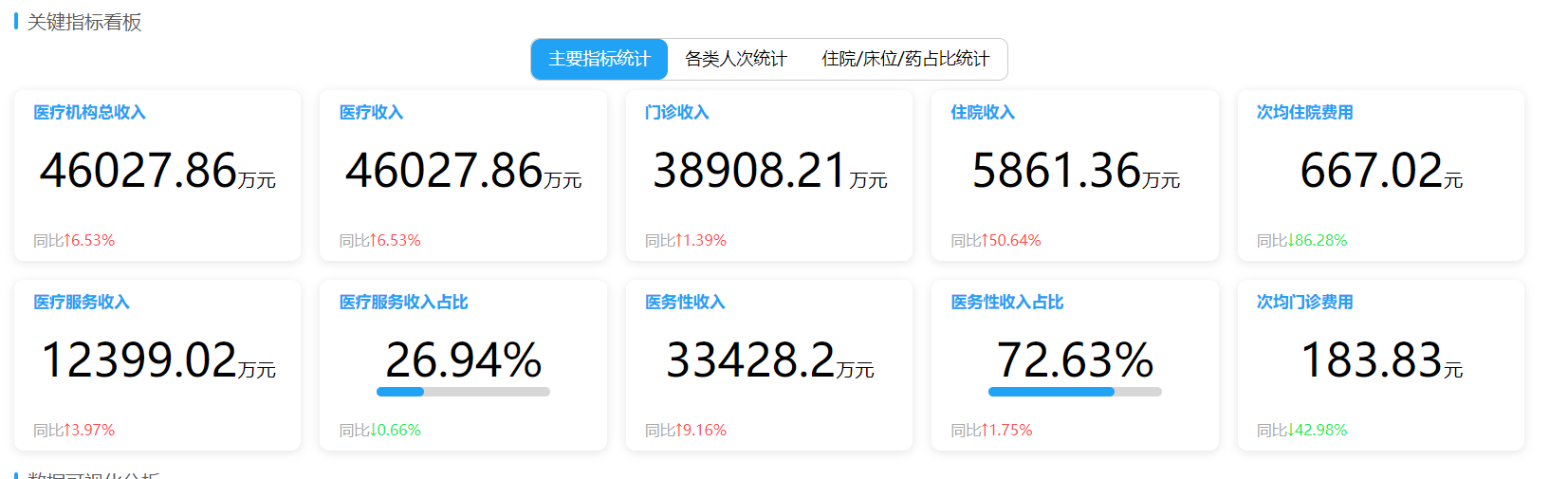
**3.调用公用方法相关接口：**

**com.ylz.sxpt.service.DynamicStatementService**

Map commonOverview(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields);  
*/\*\*  
 \** ***@Description*** *查询饼图  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map queryCircleDate(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields);  
*/\*\*  
 \** ***@Description*** *单值map查询  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map getSingleObject(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields);  
*/\*\*  
 \** ***@Description*** *查询折线图 + 表格接口  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonDataQuery(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<HeadColumnVo> headColumnVos);  
*/\*\*  
 \** ***@Description*** *单独 查询柱状图和折线图  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonPolyAndCol(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<HeadColumnVo> headColumnVos);  
*/\*\*  
 \** ***@Description*** *返回单个表格  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonTableData(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<HeadColumnVo> headColumnVos);  
*/\*\*  
 \** ***@Description*** *查询折线图 + 表格接口 增加自定义聚合函数  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonDataQuery(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<Column> orderByFields, List<Column> groupByFields, List<HeadColumnVo> headColumnVos);  
*/\*\*  
 \** ***@Description*** *查询折线图 + 表格接口 折线图是否按照年份查询  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonDataQuery(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<HeadColumnVo> headColumnVos, **boolean** isTimePoint);  
  
Map privatelyOrg(FamilyDoctorInfoReq param, Map<String,OverviewObjectVo> queryMap, List<TableObjectReq> tableObjectReqs, List<Column> columns, List<HeadColumnVo> headColumns);  
*/\*\*  
 \** ***@Description*** *单独返回一个表格  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonCustomQuery(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<Column> orderByFields, List<Column> groupByFields, List<HeadColumnVo> headColumnVos);  
*/\*\*  
 \** ***@Description*** *单独返回一个表格 增加排序参数  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonCustomQuery(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<Column> orderByFields, List<Column> groupByFields, List<HeadColumnVo> headColumnVos,**boolean** isOrderByDesc);  
*/\*\*  
 \** ***@Description*** *查询表格合计列,聚合函数参数 和 排序参数 都放置为空  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*List commonCustomQueryTotal(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<Column> orderByFields, List<Column> groupByFields, List<HeadColumnVo> headColumnVos,**boolean** isOrderByDesc);  
  
*/\*\*  
 \** ***@Description*** *默认是 按照区域分组  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*Map commonCustomQuery(FamilyDoctorInfoReq param, Map<String, OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<HeadColumnVo> headColumnVos);  
*/\*\*  
 \** ***@Description*** *详情列表接口，用于各项数据以列表、分组方法返回数据 单个表格  
 \** ***@Author*** *lcw  
 \** ***@Date*** *2019/12/1   
 \** ***@Param*** *\** ***@Return*** *\*/*List<Map> constDetailListQuery(FamilyDoctorInfoReq param, Map<String,OverviewObjectVo> resultMap, List<TableObjectReq> tables, List<Column> fields, List<Column> orderByFields, List<Column> groupByFields);

二、调用方法例子：

1. 概览返回格式1:



{  
    **"main"**:{  
        **"medicalTotalIncome"**:{  
            **"title"**:**"医疗机构总收入"**,  
            **"data"**:**23456.12**,  
            **"unit"**:**"万元"**,  
            **"an"**:**4.3**,  
            **"status"**:**"1"**  
        },  
        **"medicalRevenue"**:{  
            **"title"**:**"医疗收入"**,  
            **"data"**:**9456.34**,  
            **"unit"**:**"万元"**,  
            **"an"**:**4.3**,  
            **"status"**:**"1"**  
        }  
    },  
    **"number"**:{  
        **"treatment"**:{  
            **"title"**:**"总诊疗人次"**,  
            **"data"**:**233443**,  
            **"unit"**:**"人次"**,  
            **"an"**:**4.3**  
        },  
        **"outNumber"**:{  
            **"title"**:**"门诊人次"**,  
            **"data"**:**93443**,  
            **"unit"**:**"人次"**,  
            **"an"**:**4.3**  
        }  
    }  
}

调用例子:

Map<String, OverviewObjectVo> resultMap = **new** HashMap();  
*//概览*Map<String,String> mainMap = **new** MapUtils().put(**"tab"**,**"main"**).toMap();  
resultMap.put(**"medicalTotalIncome"**, **new** OverviewObjectVo<Double>(**"医疗机构总收入"**, **"万元"**,**"round("**+***medicalALlIncome***+**"/10000,2)"**,mainMap));  
resultMap.put(**"medicalRevenue"**, **new** OverviewObjectVo<Double>(**"医疗收入"**, **"万元"**,**"round("**+***medicalALlIncome***+**"/10000,2)"**,mainMap));  
resultMap.put(**"outRevenue"**, **new** OverviewObjectVo<Double>(**"门(急)诊收入"**, **"万元"**,**"round(sum(outMedicalBusinessIncome)/10000,2)"**,mainMap));

*//number*Map<String,String> numberMap = **new** MapUtils().put(**"tab"**,**"number"**).toMap();  
resultMap.put(**"treatment"**, **new** OverviewObjectVo<Integer>(**"总诊疗人次"**, **"人次"**,**"sum(outAllPatientNum+checkNum+vaccinationNum+registrationCarNum)"**,numberMap));  
resultMap.put(**"outNumber"**, **new** OverviewObjectVo<Integer>(**"门诊人次"**, **"人次"**,**"sum(outOutpatientNum)"**,numberMap));  
resultMap.put(**"emergencyNumber"**, **new** OverviewObjectVo<Integer>(**"急诊人次"**, **"人次"**,**"sum(outEmergencyNum)"**,numberMap));

String[] queryTables = **new** String[]{**"outStatisticsView"**,LastParamEnum.*resultLastYearTable*(**"outStatisticsView"**),  
 **"inpStatisticsView"**,LastParamEnum.*resultLastYearTable*(**"inpStatisticsView"**),  
 **"medicalFilling"**,LastParamEnum.*resultLastYearTable*(**"medicalFilling"**),  
 **"outReservationRegistered"**,LastParamEnum.*resultLastYearTable*(**"outReservationRegistered"**),  
 **"deptWardBedTable"**,LastParamEnum.*resultLastYearTable*(**"deptWardBedTable"**),  
 **"inpOperationSummaryView"**,LastParamEnum.*resultLastYearTable*(**"inpOperationSummaryView"**),  
 **"hrBaseInformationTable"**,LastParamEnum.*resultLastYearTable*(**"hrBaseInformationTable"**)  
};  
Map<String, OverviewObjectVo> map = **dynamicStatementService**.commonOverview(param, resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables));  
Map<String, Map<String,OverviewObjectVo>> main = Maps.*newHashMap*();  
//分组  
**for** (String key: map.keySet()){  
 **if**(map.get(key).getObject() !=**null** && map.get(key).getObject().containsKey(**"tab"**)) {  
 **if**(main.containsKey(map.get(key).getObject().get(**"tab"**))) {  
 main.get(map.get(key).getObject().get(**"tab"**).toString()).put(key,map.get(key));  
 } **else** {  
 Map newMap = Maps.*newHashMap*();  
 newMap.put(key,map.get(key));  
 main.put(map.get(key).getObject().get(**"tab"**).toString(),newMap);  
 }  
 }  
}  
**return** main;

1. 返回单个值(返回格式2):



{  
    **"medicalTotalIncome"**:{  
        **"title"**:**"医疗机构总收入"**,  
        **"data"**:**23456.12**,  
        **"unit"**:**"万元"**,  
        **"an"**:**4.3**,  
        **"status"**:**"1"**  
    },  
    **"medicalRevenue"**:{  
        **"title"**:**"医疗收入"**,  
        **"data"**:**9456.34**,  
        **"unit"**:**"万元"**,  
        **"an"**:**4.3**,  
        **"status"**:**"1"**  
    },  
    **"treatment"**:{  
        **"title"**:**"总诊疗人次"**,  
        **"data"**:**233443**,  
        **"unit"**:**"人次"**,  
        **"an"**:**4.3**  
    },  
    **"outNumber"**:{  
        **"title"**:**"门诊人次"**,  
        **"data"**:**93443**,  
        **"unit"**:**"人次"**,  
        **"an"**:**4.3**  
    }  
}

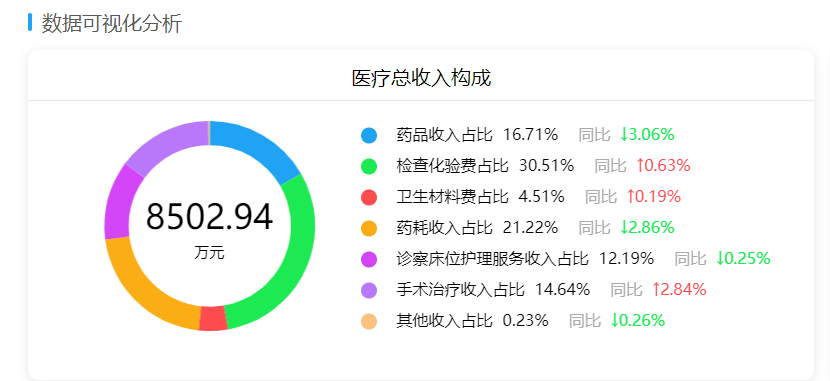
调用例子:

Map<String, OverviewObjectVo> resultMap = **new** HashMap();  
*//医疗指标监管*resultMap.put(**"medicalTotalIncome"**, **new** OverviewObjectVo<Double>(**"医疗机构总收入"**, **"万元"**, **"round("**+***medicalALlIncome***+**"/10000,2)"**));  
resultMap.put(**"medicalTotalIncomeAn"**, **new** OverviewObjectVo<Double>(**"医疗机构总收入同比"**, **"%"**, **"round(("**+***medicalALlIncome***+**"-"**+***medicalALlIncomeLy***+**")/"**+***medicalALlIncomeLy***+**"\*100,2)"**,***medicalALlIncomeLy***, **true**));  
  
resultMap.put(**"medicalRevenue"**, **new** OverviewObjectVo<Double>(**"医疗收入"**, **"万元"**, **"round("**+***medicalALlIncome***+**"/10000,2)"**));  
resultMap.put(**"medicalRevenueAn"**, **new** OverviewObjectVo<Double>(**"医疗收入"**, **"%"**, **"round(("**+***medicalALlIncome***+**"-"**+***medicalALlIncomeLy***+**")/"**+***medicalALlIncomeLy***+**"\*100,2)"**, ***medicalALlIncomeLy***, **true**));

String[] queryTables = **new** String[]{**"hrpBudgetSituationTable"**,**"outStatisticsView"**,**"inpStatisticsView"**,**"outReservationRegistered"**,**"medicalFilling"**,**"deptWardBedTable"**};  
queryTables = LastParamEnum.*addLastEnum*(queryTables, LastParamEnum.***LAST\_YEAR***);  
Map map = **dynamicStatementService**.commonOverview(param, resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables));

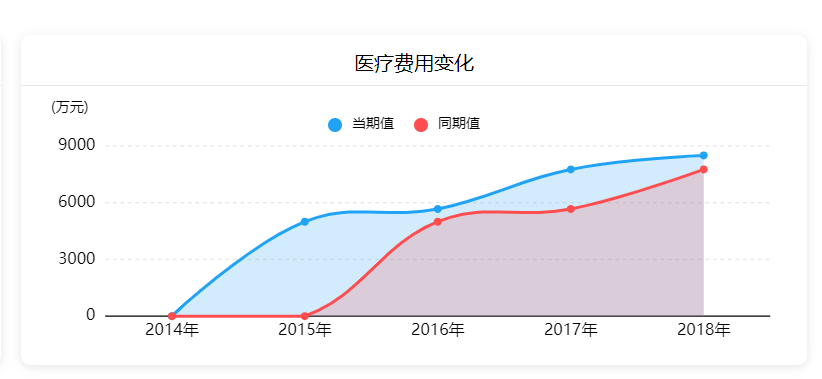
**return** map;

1. 概览返回格式3:



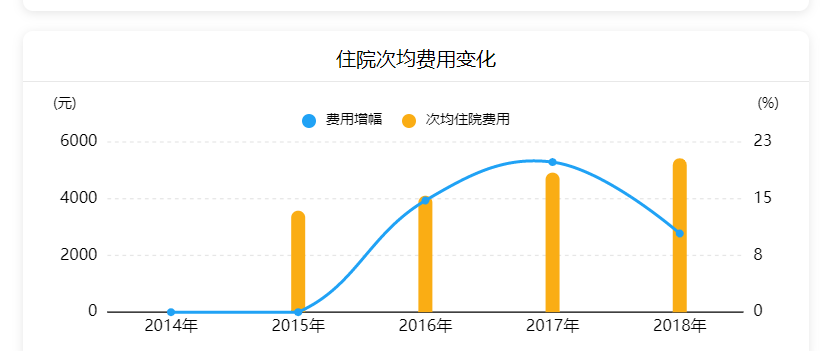
@Override  
**public** Map queryMedicalRevenueInfo2(FamilyDoctorInfoReq param) {  
 Map<String, OverviewObjectVo> resultMap = Maps.*newLinkedHashMap*();  
 *//总计* resultMap.put(**"total"**, **new** OverviewObjectVo<Double>(**"total"**, **"万元"**, **false**,**"round("**+***medicalALlIncome***+**"/10000,2)"**,**true**,**false**,**true**));  
  
 *//按照顺序写 会自动排序  
 //第一行* resultMap.put(**"drugIncome"**, **new** OverviewObjectVo<Double>(**"药品收入占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpDrugIncome+outDrugIncome)/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"drugIncomeXp"**, **new** OverviewObjectVo<Double>(**"药品收入占比"**, **"%"**, **false**,**"round(sum(inpDrugIncome+outDrugIncome)/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"drugIncomeAn"**, **new** OverviewObjectVo<Double>(**"药品收入占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpDrugIncome+outDrugIncome)/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum(inpDrugIncomeLy+outDrugIncomeLy)/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
  
 *//第二行* resultMap.put(**"checkIncome"**, **new** OverviewObjectVo<Double>(**"检查化验费占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(outCheckedIncome+outLabIncome+inpLabIncome+inpCheckedIncome)/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"checkIncomeXp"**, **new** OverviewObjectVo<Double>(**"检查化验费占比"**, **"%"**, **false**,**"round(sum(outCheckedIncome+outLabIncome+inpLabIncome+inpCheckedIncome)/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"checkIncomeAn"**, **new** OverviewObjectVo<Double>(**"检查化验费占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(outCheckedIncome+outLabIncome+inpLabIncome+inpCheckedIncome)/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum(outCheckedIncomeLy+outLabIncomeLy+inpLabIncomeLy+inpCheckedIncomeLy)/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
  
 *//第三行* resultMap.put(**"clfIncome"**, **new** OverviewObjectVo<Double>(**"卫生材料费占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpMaterialIncome+outMaterialIncome)/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"clfIncomeXp"**, **new** OverviewObjectVo<Double>(**"卫生材料费占比"**, **"%"**, **false**,**"round(sum(inpMaterialIncome+outMaterialIncome)/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"clfIncomeAn"**, **new** OverviewObjectVo<Double>(**"卫生材料费占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpMaterialIncome+outMaterialIncome)/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum(inpMaterialIncomeLy+outMaterialIncomeLy)/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
  
 *//第四行* resultMap.put(**"yhsrzbIncome"**, **new** OverviewObjectVo<Double>(**"药耗收入占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpDrugIncome+outDrugIncome+inpMaterialIncome+outMaterialIncome)/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"yhsrzbIncomeXp"**, **new** OverviewObjectVo<Double>(**"药耗收入占比"**, **"%"**, **false**,**"round(sum(inpDrugIncome+outDrugIncome+inpMaterialIncome+outMaterialIncome)/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"yhsrzbIncomeAn"**, **new** OverviewObjectVo<Double>(**"药耗收入占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpDrugIncome+outDrugIncome+inpMaterialIncome+outMaterialIncome)/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum(inpDrugIncomeLy+outDrugIncomeLy+inpMaterialIncomeLy+outMaterialIncomeLy)/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
  
 String jcyphycl = **"inpMedicalBusinessIncome+outMedicalBusinessIncome-inpDrugIncome-outDrugIncome-inpMaterialIncome-outMaterialIncome-outCheckedIncome-outLabIncome-inpLabIncome-inpCheckedIncome"**;  
 String jcyphyclLy = **"inpMedicalBusinessIncomeLy+outMedicalBusinessIncomeLy-inpDrugIncomeLy-outDrugIncomeLy-inpMaterialIncomeLy-outMaterialIncomeLy-outCheckedIncomeLy-outLabIncomeLy-inpLabIncomeLy-inpCheckedIncomeLy"**;  
 *//第五行* resultMap.put(**"zccwhlfwIncome"**, **new** OverviewObjectVo<Double>(**"诊察床位护理服务收入占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum("**+jcyphycl+**")/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"zccwhlfwIncomeXp"**, **new** OverviewObjectVo<Double>(**"诊察床位护理服务收入占比"**, **"%"**, **false**,**"round(sum("**+jcyphycl+**")/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"zccwhlfwIncomeAn"**, **new** OverviewObjectVo<Double>(**"诊察床位护理服务收入占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum("**+jcyphycl+**")/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum("**+jcyphyclLy+**")/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
  
 *//第六行* resultMap.put(**"shzlzbIncome"**, **new** OverviewObjectVo<Double>(**"手术治疗收入占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(outSurgeryIncome+outTreatmentIncome+inpTreatmentIncome+inpSurgeryIncome)/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"shzlzbIncomeXp"**, **new** OverviewObjectVo<Double>(**"手术治疗收入占比"**, **"%"**, **false**,**"round(sum(outSurgeryIncome+outTreatmentIncome+inpTreatmentIncome+inpSurgeryIncome)/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"shzlzbIncomeAn"**, **new** OverviewObjectVo<Double>(**"手术治疗收入占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(outSurgeryIncome+outTreatmentIncome+inpTreatmentIncome+inpSurgeryIncome)/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum(outSurgeryIncomeLy+outTreatmentIncomeLy+inpTreatmentIncomeLy+inpSurgeryIncomeLy)/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
  
 *//第七行* resultMap.put(**"otherIncome"**, **new** OverviewObjectVo<Double>(**"其他收入占比"**, **"%"**, **false**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpOtherIncome+outOtherIncome)/"**+***medicalALlIncome***+**"\*100,2))"**,**true**,**false**,**false**));  
 resultMap.put(**"otherIncomeXp"**, **new** OverviewObjectVo<Double>(**"其他收入占比"**, **"%"**, **false**,**"round(sum(inpOtherIncome+outOtherIncome)/10000,2)"**,**true**,**true**,**false**));  
 resultMap.put(**"otherIncomeAn"**, **new** OverviewObjectVo<Double>(**"其他收入占比同比"**, **"%"**,**true**,**"decode("**+***medicalALlIncome***+**",0,0,round(sum(inpOtherIncome+outOtherIncome)/"**+***medicalALlIncome***+**"\*100,2))-decode("**+***medicalALlIncomeLy***+**",0,0,round(sum(inpOtherIncomeLy+outOtherIncomeLy)/"**+***medicalALlIncomeLy***+**"\*100,2))"**,**true**,**false**,**false**));  
 String[] queryTables = **new** String[]{  
 **"outStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"outStatisticsView"**),  
 **"inpStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"inpStatisticsView"**)  
 };  
 Map map = **dynamicStatementService**.queryCircleDate(param,resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables));  
 **return** map;  
}

1. 概览返回格式4:



@Override  
**public** Map queryMedicalRevenueVariety2(FamilyDoctorInfoReq param) {  
 Map<String, OverviewObjectVo> resultMap = Maps.*newLinkedHashMap*();  
 Map typeMap = **new** MapUtils().put(**"orderType"**,**"1"**).toMap();  
 resultMap.put(**"nowData"**, **new** OverviewObjectVo<Double>(**"当期值"**, **"万元"**, **"round("**+***medicalALlIncome***+**"/10000,2)"**,**true**, **true**, CoordinateEnum.***Y***,typeMap));  
 resultMap.put(**"beforeData"**, **new** OverviewObjectVo<Double>(**"同期值"**, **"万元"**, **"round("**+***medicalALlIncomeLy***+**"/10000,2)"**,**true**, **true**, CoordinateEnum.***Y***,typeMap));  
  
 List<HeadColumnVo> headColumns = **new** ArrayList<>();  
 String[] queryTables = **new** String[]{  
 **"outStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"outStatisticsView"**),  
 **"inpStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"inpStatisticsView"**)  
 };  
 Map map = **dynamicStatementService**.commonPolyAndCol(param, resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables), headColumns);  
 Map dataMap = ((HashMap)((ArrayList)map.get(**"chartList"**)).get(0));  
 **return** dataMap;  
}

1. 概览返回格式5:



@Override  
**public** Map queryInAverageRevenueInfo2(FamilyDoctorInfoReq param){  
 Map<String, OverviewObjectVo> resultMap = Maps.*newLinkedHashMap*();  
 resultMap.put(**"nowData"**, **new** OverviewObjectVo<Double>(**"次均住院费用"**, **"%"**, **"decode(sum(inpDischargedNum),0,0,round(sum(inpMedicalBusinessIncome)/sum(inpDischargedNum),2))"**,**true**, **true**, CoordinateEnum.***DL***));  
 resultMap.put(**"beforeData"**, **new** OverviewObjectVo<Double>(**"费用增幅"**, **"%"**, **"round((decode(sum(inpDischargedNum),0,0,round(sum(inpMedicalBusinessIncome)/sum(inpDischargedNum),2))-decode(sum(inpDischargedNumLy),0,0,round(sum(inpMedicalBusinessIncomeLy)/sum(inpDischargedNumLy),2)))/decode(sum(inpDischargedNumLy),0,0,round(sum(inpMedicalBusinessIncomeLy)/sum(inpDischargedNumLy),2))\*100,2)"**,**"decode(sum(inpDischargedNumLy),0,0,round(sum(inpMedicalBusinessIncomeLy)/sum(inpDischargedNumLy),2))"**, **true**, CoordinateEnum.***DR***));  
  
 List<HeadColumnVo> headColumns = **new** ArrayList<>();  
 String[] queryTables = **new** String[]{  
 **"outStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"outStatisticsView"**),  
 **"inpStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"inpStatisticsView"**)  
 };  
 Map map = **dynamicStatementService**.commonPolyAndCol(param, resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables), headColumns);  
 Map dataMap = ((HashMap)((ArrayList)map.get(**"chartList"**)).get(0));  
 **return** dataMap;  
}

1. 概览返回格式6:



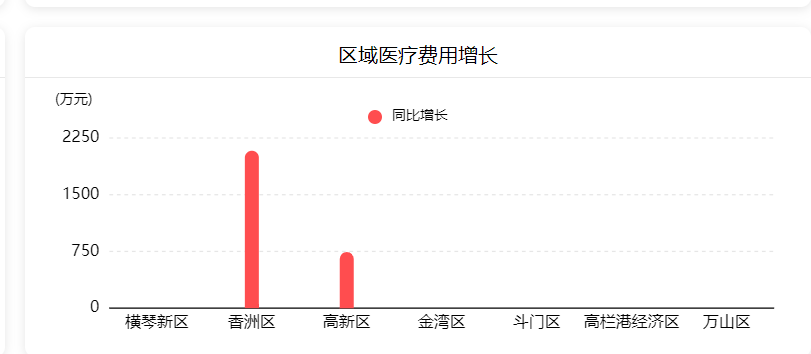
@Override  
 **public** Map orgIncome(FamilyDoctorInfoReq param) {  
 Map<String, OverviewObjectVo> resultMap = **new** HashMap();resultMap.put(**"ylsr"**, **new** OverviewObjectVo<Double>(**"医疗机构总收入"**, **"万元"**, **"round("**+***medicalALlIncome***+**"/10000,2)"**,  
 **true**, **true**, **true**, **false**, **true**, CoordinateEnum.***DL***));  
 resultMap.put(**"ylsrtb"**, **new** OverviewObjectVo<Double>(**"同比"**, **"%"**,**"round(("**+***medicalALlIncome***+**"-"**+***medicalALlIncomeLy***+**")/"**+***medicalALlIncomeLy***+**"\*100,1)"**,  
 ***medicalALlIncomeLy***, **true**, CoordinateEnum.***DR***));  
 resultMap.put(**"ywzsr"**, **new** OverviewObjectVo<Double>(**"医疗业务收入"**, **"万元"**, **"round("**+***medicalALlIncome***+**"/10000,2)"**));  
 resultMap.put(**"zysr"**, **new** OverviewObjectVo<Double>(**"财政补助收入"**, **"万元"**, **"round(sum(finSubIncome)/10000,2)"**));  
 resultMap.put(**"jkjcsr"**, **new** OverviewObjectVo<Double>(**"科教项目收入"**, **"万元"**, **"round(sum(scienceIncome)/10000,2)"**));  
 resultMap.put(**"jmsr"**, **new** OverviewObjectVo<Double>(**"其他收入"**, **"万元"**, **"0"**));  
  
 List<HeadColumnVo> headColumns = **new** ArrayList<>();  
 headColumns.add(**new** HeadColumnVo(**"sort"**, **"序号"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"areaName"**, **"区域/机构名称"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"ylsr"**, **"医疗机构总收入（万元）"**, **""**, **false**, **false**, ShowProgressEnum.***T***, **true**));  
 headColumns.add(**new** HeadColumnVo(**"ylsrtb"**, **"同比"**, **"%"**, **false**, **true**));  
 headColumns.add(**new** HeadColumnVo(**"ywzsr"**, **"医疗业务收入"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"zysr"**, **"财政补助收入"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"jkjcsr"**, **"科教项目收入"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"jmsr"**, **"其他收入"**, **""**));  
 String[] queryTables = **new** String[]{**"outStatisticsView"**, **"inpStatisticsView"**, **"hrpBudgetSituationTable"**};  
 queryTables = LastParamEnum.*addLastEnum*(queryTables, LastParamEnum.***LAST\_YEAR***);  
 **return dynamicStatementService**.commonDataQuery(param, resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables), headColumns);  
 }

1. 概览返回格式7（多层聚合）:



@Override  
 **public** Map privatelyOrgNumDetail(FamilyDoctorInfoReq param) {  
 Map<String, OverviewObjectVo> queryMap = Maps.*newHashMap*();  
 param.setOrgFlag(**"1"**);  
 param.setPrivateAndPublicOrgFlag(**"1"**);  
 queryMap.put(**"parentId"**, **new** OverviewObjectVo<String>(**"机构id"**, **""**, **"orgId"**, **true**));  
 queryMap.put(**"orgId"**, **new** OverviewObjectVo<String>(**"机构id"**, **""**, **"orgId"**, **true**));  
 queryMap.put(**"areaRank"**, **new** OverviewObjectVo<Integer>(**"排名"**, **""**, **"dense\_rank() over(order by addressId asc) "**, **true**));  
 queryMap.put(**"addressId"**, **new** OverviewObjectVo<String>(**"区域id"**, **""**, **"addressId"**, **true**,**true**));  
 queryMap.put(**"addressName"**, **new** OverviewObjectVo<String>(**"区县"**, **""**, **"addressName"**, **true**));  
*// queryMap.put("orgSum", new OverviewObjectVo<Integer>("民营医疗机构数量", "", "sum(privateOrgNum)", true));  
// queryMap.put("orgRate", new OverviewObjectVo<Double>("占比", "%", "round(sum(privateOrgNum)/sum(publicOrgNum)\*100,2)", "sum(publicOrgNum)"));* queryMap.put(**"orgName"**, **new** OverviewObjectVo<String>(**"机构名称"**, **""**, **"orgName"**, **true**));  
 queryMap.put(**"orgGrade"**, **new** OverviewObjectVo<String>(**"机构等级"**, **""**, **"orgGrade"**, **true**));  
 queryMap.put(**"address"**, **new** OverviewObjectVo<String>(**"地址"**, **""**, **"address"**, **true**));  
 queryMap.put(**"linkman"**, **new** OverviewObjectVo<String>(**"联系人"**, **""**, **"linkman"**, **true**));  
 queryMap.put(**"president"**, **new** OverviewObjectVo<String>(**"院长"**, **""**, **"president"**, **true**));  
 queryMap.put(**"foundingTime"**, **new** OverviewObjectVo<String>(**"创建时间"**, **""**, **"foundingTime"**, **true**));  
  
 List<HeadColumnVo> headColumns = **new** ArrayList<>();  
 headColumns.add(**new** HeadColumnVo(**"sort"**, **"序号"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"areaRank"**, **"排名"**, **""**, **"addressName"**));  
 headColumns.add(**new** HeadColumnVo(**"addressName"**, **"区县"**, **""**, **"addressName"**));  
 headColumns.add(**new** HeadColumnVo(**"orgSum"**, **"民营医疗机构数量"**, **""**, **"addressName"**));  
 headColumns.add(**new** HeadColumnVo(**"orgRate"**, **"占比"**, **"%"**, **"addressName"**));  
 headColumns.add(**new** HeadColumnVo(**"orgName"**, **"机构名称"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"orgGrade"**, **"机构等级"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"address"**, **"地址"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"linkman"**, **"联系人"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"president"**, **"院长"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"foundingTime"**, **"创建时间"**, **""**));  
  
 List<Column> groupByColumn = Lists.*newArrayList*();  
 groupByColumn.add(**new** Column(**"addressId"**,**"addressId"**,**"区域id"**, **true**));  
 groupByColumn.add(**new** Column(**"addressName"**,**"addressName"**,**"区域名称"**, **true**));  
 groupByColumn.add(**new** Column(**"orgName"**,**"orgName"**,**"机构名称"**, **true**));  
 groupByColumn.add(**new** Column(**"orgGrade"**,**"orgGrade"**,**"机构等级"**, **true**));  
 groupByColumn.add(**new** Column(**"address"**,**"address"**,**"地址"**, **true**));  
 groupByColumn.add(**new** Column(**"linkman"**,**"linkman"**,**"联系人"**, **true**));  
 groupByColumn.add(**new** Column(**"president"**,**"foundingTime"**,**"院长"**, **true**));  
 groupByColumn.add(**new** Column(**"foundingTime"**,**"foundingTime"**,**"创建时间"**, **true**));  
 groupByColumn.add(**new** Column(**"orgId"**,**"orgId"**,**"机构id"**, **true**));  
  
 List<Column> orderByColumn = Lists.*newArrayList*();  
 *//列名可以取 函数名加列名 也可以去别名 'patientNum'* orderByColumn.add(**new** Column(**"to\_number(addressId)"**,**"to\_number(addressId)"**,**"区域id"**,**true**));  
  
 String[] queryTables = **new** String[]{**"privateOrgTable"**};  
  
 Map<String,Map<String,List>> dataMap = **dynamicStatementService**.commonCustomQuery(param, queryMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables),orderByColumn,groupByColumn, headColumns,**false**);  
*// //查询 区域级别的 占比 和 数量* queryMap.clear();  
 queryMap.put(**"orgSum"**, **new** OverviewObjectVo<Integer>(**"民营医疗机构数量"**, **""**, **"sum(privateOrgNum)"**, **true**));  
 queryMap.put(**"orgRate"**, **new** OverviewObjectVo<Double>(**"占比"**, **"%"**, **"round(sum(privateOrgNum)/sum(publicOrgNum)\*100,2)"**, **"sum(publicOrgNum)"**));  
 param.setOrgFlag(**"0"**);  
 queryTables = **new** String[]{**"orgTable"**};  
 Map<String,Map<String,List>> dataMap2 = **dynamicStatementService**.commonTableData(param, queryMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables), headColumns);  
 List areaList = dataMap2.get(**"tableData"**).get(**"list"**);  
 List dataList = dataMap.get(**"tableData"**).get(**"list"**);  
 dataList.forEach(dataDetail-> {  
 JSONObject data = JSON.*parseObject*(JSON.*toJSONString*(dataDetail));  
 **for** (Object o: areaList){  
 JSONObject object = JSON.*parseObject*(JSON.*toJSONString*(o));  
 **if** (data.getString(**"addressId"**).equals(object.getString(**"areaId"**))){  
 data.put(**"orgSum"**,object.get(**"orgSum"**));  
 data.put(**"orgRate"**,object.get(**"orgRate"**));  
 dataDetail = data;  
 }  
 }  
 });  
 dataMap.get(**"tableData"**).put(**"list"**,dataList);  
 **return** dataMap;  
 }

1. 概览返回格式8（按照区域分组图形）:



@Override  
 **public** Map queryRegionalMedicalFee2(FamilyDoctorInfoReq param){  
 param.setCounty(**"440400000000"**);  
 Map<String, OverviewObjectVo> resultMap = Maps.*newLinkedHashMap*();  
 resultMap.put(**"add"**, **new** OverviewObjectVo<Double>(**"同比增长"**, **"万元"**, **"round(("**+***medicalALlIncome***+**"-"**+***medicalALlIncomeLy***+**")/10000,2)"**,**true**));  
 resultMap.put(**"group"**, **new** OverviewObjectVo<Double>(**"名称"**, **"名称"**, **"'同比增长'"**,**true**));  
  
 List<HeadColumnVo> headColumns = **new** ArrayList<>();  
 headColumns.add(**new** HeadColumnVo(**"areaName"**, **"区域/机构名称"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"add"**, **"同比增长"**, **"万元"**, **false**, **false**, ShowProgressEnum.***P***, **true**));  
 String[] queryTables = **new** String[]{  
 **"outStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"outStatisticsView"**),  
 **"inpStatisticsView"**, LastParamEnum.*resultLastYearTable*(**"inpStatisticsView"**),  
 };  
 Map map = **dynamicStatementService**.commonTableData(param, resultMap, TablesUtil.*queryTables*(queryTables), TablesUtil.*fieldsList*(queryTables), headColumns);  
 Map dataMap = (HashMap)map.get(**"tableData"**);  
 List list = (ArrayList)dataMap.get(**"list"**);List dataList = Lists.*newArrayList*();**for** (Object obj: list){  
 Map newMap = (HashMap)obj;  
 newMap.put(**"x"**,newMap.get(**"areaName"**));  
 newMap.put(**"y"**,newMap.get(**"add"**));  
 dataList.add(newMap);  
 }  
 Map mapList = Maps.*newHashMap*();  
 mapList.put(**"data"**,dataList);  
 **return** mapList;  
 }

1. 概览返回格式9（自定义聚合列）:



@Override  
**public** Map queryOutDoctorDrugRanking(FamilyDoctorInfoReq param) {  
 Map<String,OverviewObjectVo> resultMap = **new** HashMap<>();  
 resultMap.put(**"parentId"**,**new** OverviewObjectVo(**"key"**,**""**,**"drugCode"**,**true**));  
 resultMap.put(**"areaName"**,**new** OverviewObjectVo(**"药品名称"**,**""**,**"drugName"**,**true**));  
 **if** (StringUtils.isNotEmpty(param.getDrugCode())){  
 resultMap.put(**"parentId"**,**new** OverviewObjectVo(**"key的id"**,**""**,**"parentId"**,**true**));  
 resultMap.put(**"areaName"**,**new** OverviewObjectVo(**"机构名称"**,**""**,**"parentName"**,**true**));  
 }  
 resultMap.put(**"deptName"**,**new** OverviewObjectVo(**"科室"**,**""**,**"deptName"**,**true**));  
 resultMap.put(**"doctorName"**,**new** OverviewObjectVo(**"医生名称"**,**""**,**"doctorName"**,**true**));  
 resultMap.put(**"userTimes"**,**new** OverviewObjectVo(**"人次"**,**""**,**"sum(outUserTimes)"**,**true**));  
 resultMap.put(**"outpatientMoney"**,**new** OverviewObjectVo(**"费用"**,**"元"**,**"sum(outpatientDrugMoney)"**,**true**));  
 List<HeadColumnVo> headColumns = **new** ArrayList<>();  
 String areaName = StringUtils.isEmpty(param.getDrugCode()) ? **"药品名称"**:**"区域/机构名称"**;  
 headColumns.add(**new** HeadColumnVo(**"sort"**, **"序号"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"areaName"**, areaName, **""**));;  
 headColumns.add(**new** HeadColumnVo(**"deptName"**, **"科室"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"doctorName"**, **"医生名称"**, **""**));  
 headColumns.add(**new** HeadColumnVo(**"userTimes"**,**"人次"**,**""**,**false**,**false**, ShowProgressEnum.F,**true**));  
 headColumns.add(**new** HeadColumnVo(**"outpatientMoney"**,**"费用(元)"**,**"元"**,**false**,**false**, ShowProgressEnum.F,**true**));  
  
 List<Column> orderByColumn = Lists.newArrayList();  
 orderByColumn.add(**new** Column(**"sum(outUserTimes)"**,**"sum(outUserTimes)"**,**"人次"**));  
 List<Column> groupByColumn = Lists.newArrayList();  
 groupByColumn.add(**new** Column(**"drugName"**,**"drugName"**,**"药品名称"**, **true**));  
 groupByColumn.add(**new** Column(**"drugCode"**,**"drugCode"**,**"药品编码"**, **true**));  
 groupByColumn.add(**new** Column(**"deptName"**,**"deptName"**,**"科室名称"**, **true**));  
 groupByColumn.add(**new** Column(**"doctorName"**,**"doctorName"**,**"医生名称"**, **true**));  
 **if** (StringUtils.isNotEmpty(param.getDrugCode())){  
 groupByColumn.add(**new** Column(**"parentId"**,**"parentId"**,**"父级id"**, **true**));  
 groupByColumn.add(**new** Column(**"parentName"**,**"parentName"**,**"机构名称"**, **true**));  
 }  
 String[] queryTables = **new** String[]{**"outpDrugUsedRecordView2"**};  
  
 *//增加动态条件* **if** (StringUtils.isNotEmpty(param.getDrugCode())){  
 String drugCode = param.getDrugCode();  
 String drugName = param.getDrugName();  
 Map<String,String> filterSql = Maps.newHashMap();  
 filterSql.put(**"OUTP\_DRUG\_USED\_RECORD\_VIEW"**,**" a.drug\_code = '"** + drugCode + **"'"** + **" and a.drug\_name = '"** + drugName +**"'"**);  
 param.setFilterSql(filterSql);  
 }  
 PageHelper.startPage(1,10);  
 Map result = dynamicStatementService.commonCustomQuery(param, resultMap, TablesUtil.queryTables(queryTables), TablesUtil.fieldsList(queryTables),orderByColumn,groupByColumn, headColumns);  
 **if** (StringUtils.isNotEmpty(param.getDrugCode())){  
 Map<String,OverviewObjectVo> resultMapCol = **new** HashMap<>();  
 resultMapCol.put(**"userTimes"**,**new** OverviewObjectVo(**"人次"**,**""**,**"sum(outUserTimes)"**,**true**,**true**, CoordinateEnum.Y));  
 queryTables = **new** String[]{**"outpDrugUsedRecordView3"**};  
 Map map = dynamicStatementService.commonPolyAndCol(param, resultMapCol, TablesUtil.queryTables(queryTables), TablesUtil.fieldsList(queryTables), headColumns);  
 Map dataMap = ((HashMap)((ArrayList)map.get(**"chartList"**)).get(0));  
 dataMap.put(**"unit"**,**"人次"**);  
 result.putAll(map);  
 }  
 **return** result;  
}

注:更多返回格式请参考:

<https://www.yuque.com/usersystem/vqxo5n/oolx42>

代码位置:

<https://3mylz.com/svn/jksm/广东省项目/珠海項目/zhygjc>

项目地址:

[http://121.204.176.144:8077](http://121.204.176.144:8077/cockpit/main)

zhygjcadmin

123456