

用友BIP | iuap平台

YonBuilder-YPD 专业服务开发培训

典型案例功能及在线脚本开发

2023 年 2 月

文档修订摘要

日期	修订号	描述	著者	审阅者
2023-02-23	V20221230	添加YPD函数案例	【卢鹏菲】	
2023-03-29	V202303r2	修改按钮状态配置	【卢鹏菲】	
2023-11-06	V20230930	按照新版本调整截图	【卢鹏菲】	

版权

©2023 用友集团版权所有。

未经用友集团的书面许可，本文档描述任何整体或部分的内容不得被复制、复印、翻译或缩减以用于任何目的。本文档描述的内容在未经通知的情形下可能会发生改变，敬请留意。请注意：本文档描述的内容并不代表用友集团所做的承诺。

数智平台事业部

目录

版权.....	2
1 业务场景.....	5
2 对象建模.....	6
2.1 基本概念	6
2.1.1 业务对象.....	6
2.1.2 场景支持.....	7
2.2 新建枚举	7
2.3 新建业务对象	8
2.4 新建实体	9
2.4.1 核心功能介绍.....	9
2.4.2 添加主实体.....	10
2.4.3 添加子实体.....	14
2.4.4 执行脚本.....	16
2.4.5 场景支持配置.....	16
2.5 特征配置	16
2.5.1 业务对象配置.....	16
2.5.2 特征实体表配置.....	17
2.5.3 特征分类定义.....	24
2.5.1 特征定义.....	25
2.5.2 特征分配.....	25
3 页面建模.....	27
3.1 新建页面	27
3.2 页面布局说明	28
3.3 列表界面配置	30
3.3.1 调整查询区.....	30
3.3.2 列表界面样式调整.....	32
3.3.3 应用效果.....	35
3.4 卡片界面配置	35
3.4.1 业务需求.....	35
3.4.2 主表界面配置.....	35
3.4.3 子表界面配置.....	40
3.4.4 应用效果.....	41
3.5 页面规则功能配置	43
3.5.1 业务场景.....	43
3.5.2 页面规则.....	43
3.5.3 应用效果.....	44
3.6 参照业务功能配置	48
3.6.1 业务场景.....	48

3.6.2	参照过滤.....	48
3.6.3	应用效果.....	49
4	函数式在线脚本.....	50
4.1	前端函数.....	50
4.1.1	业务场景.....	50
4.1.2	卡片页面初始化.....	50
4.1.3	金额计算.....	57
4.1.4	查询区扩展.....	58
4.1.5	前端调试.....	60
4.2	后端函数.....	63
4.2.1	业务场景.....	63
4.2.2	后端函数说明.....	63
4.2.3	后端函数开发.....	65
4.2.4	应用效果.....	69
5	典型业务功能.....	70
5.1	按钮功能开发.....	70
5.1.1	业务需求.....	70
5.1.2	主表按钮开发.....	70
5.1.3	子表行按钮开发.....	74
5.1.4	应用效果.....	77
6	附录.....	79
6.1	MODEL 模型基本概念.....	79

1 业务场景

本章节我们将开发预订单节点，样式为主子表样式。主要目的是让学员了解线上开发过程，如何通过线上提供的函数实现业务功能开发。

预订单展现方式为列表卡片主子表样式，可以录入预订单；节点最终效果如下图

默认方案

销售组织

销售区域

编码

新增

提交

下推

导入

导出

删除

打印

下载

销售组织	编码	销售区域	订货日期	失效日期	客户	销售渠道类型	代销级别	代销公司	业务员	出
<input checked="" type="checkbox"/>	传统销售中心	000003	苏杭区域	2022-05-05	2022-05-21	北方实业	代销	二级	AA公司	--
<input type="checkbox"/>	传统销售中心	000002	东区区域	2022-05-05	2022-05-21	北方实业	代销	二级	AA公司	--
<input type="checkbox"/>	传统销售中心	000001	苏杭区域	2022-05-05	2022-05-21	北方实业	代销	二级	AA公司	--
小计										
合计										

共 3 条 << 1 >> 20 条 页

销售预订单明细

序号	物料编码	物料名称	数量	单位	单价	单品折扣	税率	税额	价税合计	计划发货日期	出
1	01000001	显示器	100.00	台	1,000.00	5.00%	10.00%	10.00	10,100.00	2022-05-21	
2	01000002	台式机	200.00	台	1,000.00	8.00%	12.00%	28.00	20,211.00	2022-05-18	
合计			300.00					38.00	30,311.00		

卡片显示效果

返回

复制

审批

下推

全局联查

销售订单打印

打印

辅助功能

变更历史查询

行关闭

销售信息

销售组织 传统销售中心

销售渠道类型 代销

代销信息

代销公司 AA公司

预订信息

编码 000003

业务员 --

总数量 300.00

单据状态 已审核

备注 --

客户信息

客户 北方实业

开票客户 --

销售区域 苏杭区域

销售部门 销售部

代销级别 二级

币种 人民币

订货日期 2022-05-05

价格合计 3,012,567.00

运输方式 公路运输

失效日期 2022-05-21

预订状态 正常

销售预订单明细

序号	物料编码	物料名称	数量	单位	单价	单品折扣	税率	税额	价税合计	计划
<input type="checkbox"/>	1	01000001	显示器	100.00	台	1,000.00	5.00	10.00	10,100.00	2022-05-21
<input type="checkbox"/>	2	01000002	台式机	200.00	台	1,000.00	8.00	12.00	20,211.00	2022-05-18

创建时间 2022-05-07 19:53:55 创建人 joydev

2 对象建模

2.1 基本概念

2.1.1 业务对象

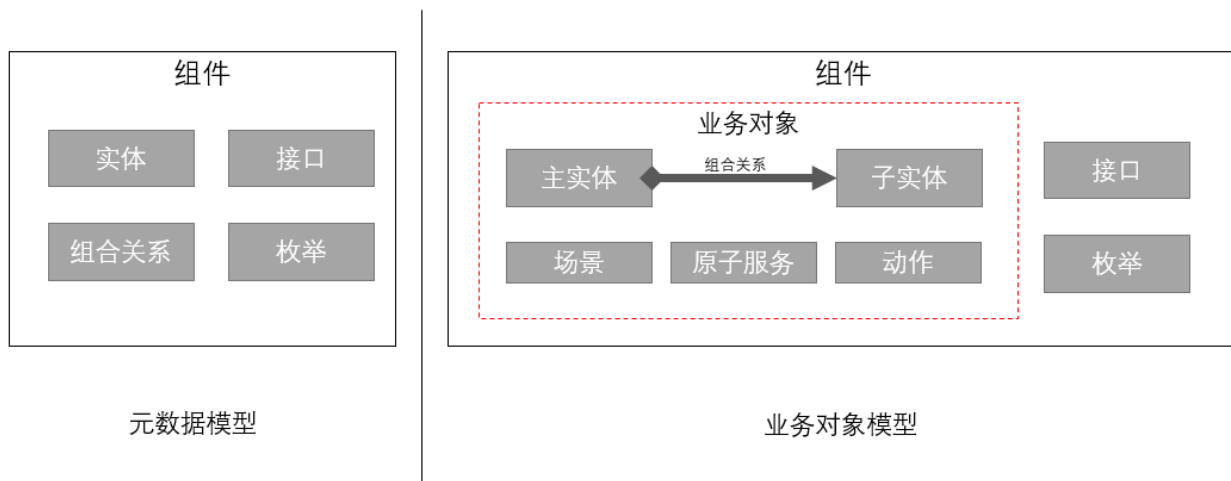
业务对象就是指软件业务建模中的业务主体，一般一个业务对象是指现实世界中有一定生命周期的实体，可以是基础数据，比如员工、供应商、客户、物料等。也可以是交易单据，比如采购订单、销售订单等。

YonBIP 业务对象是对实体元数据的扩展和补充，运行态都是元数据的实体。业务对象是以一个元数据实体作为主实体的聚合实体。一个业务对象包含主实体以及主实体下的子孙实体。

业务对象对元数据实体进行了如下方面的增强：

- 1) 业务对象增加了场景、原子服务、动作等。
- 2) 业务对象对应的实体的属性增加了属性业务类型、属性上的默认参照、规则信息(dataRule)、日期格式等。
- 3) 一个业务对象下有且只有一个主实体。
- 4) 元数据实体上支持定义表达式属性。
- 5) 业务对象增加了业务数据类型,取值范围：文本、数值、整数、日期、日期时间、单选、多选、单选引用、多选引用、联系方式、开关、附件、图片、证件号、定位、多语文本、大文本、评分、时间。
- 6) 支持在实体上支持设置过滤条件（报销单、组织和部门场景）。

元数据模型和业务对象模型的区别如下图：



2.1.2 场景支持

业务对象的可使用范围通过场景来定义，在业务对象上标记场景后就可以支持相关业务功能，如在业务对象上标记报表场景，表示该业务对象可以进行报表分析。场景支持标记在业务对象上，也可以标注在业务对象的属性上。目前版本场景支持如下

实体 场景支持			
序号	场景名称	启用	描述
1	打印	<input type="checkbox"/>	打印场景，当前业务对象支持进行打印模板、打印控制等相关功能配置。
2	编码规则	<input type="checkbox"/>	编码规则场景，支持业务对象实体字段编号的生成规则的定义和维护。
3	审批流	<input type="checkbox"/>	审批流场景，支持对业务对象单据配置审批流程，做到业务单据与业务审批流程联动。
4	业务流	<input type="checkbox"/>	业务流场景，支持多个业务对象（流程单据）进行业务场景流程的定义，处理多个业务对象之间数据流转控制如：触发时机、转换规则等。
5	报表	<input type="checkbox"/>	报表场景，支持对业务对象配置报表，用表格、图表等格式来动态显示数据。
6	电子合同	<input type="checkbox"/>	电子合同场景，支持业务对象数据字段参与电子合同编制，成为销售合同、采购合同、人力合同等合同主体的编制内容。
7	消息模版	<input type="checkbox"/>	消息模版场景，支持对业务对象定义系统发送的各种消息的不同展现样式，包括消息标题和消息内容、消息。
8	交易类型	<input type="checkbox"/>	交易类型场景，支持租户级基于业务的不同性质及管理要求，对同一业务对象下的业务场景进行的细分分类的定义，可作用于流程、模板等。
9	规则引擎	<input type="checkbox"/>	规则引擎是一种嵌入在应用中的组件，使用预先定义好的语义规范来实现剥离出来的业务规则，通过接受输入的环境和业务数据，进行业务评估，执行业务
10	自定义项	<input type="checkbox"/>	自定义项分配场景，支持自定义项分配。
11	数据权限	<input type="checkbox"/>	数据权限控制场景，支持业务对象实体、单据的数据权限配置。
12	字段权限	<input type="checkbox"/>	字段权限控制场景，支持业务对象实体、单据中具体字段的权限配置。
13	业务流起始单据	<input type="checkbox"/>	业务流程比如采购订货(普通采购)流程，流程以核心单据采购订单起始，流程也会从核心起始单据一直向下游携带。
14	高级条码服务	<input type="checkbox"/>	启用高级条码场景，支持按照业务对象属性配置条码规则和打印模板，支持条码规则和打印模板按照业务对象属性定义使用条件。

2.2 新建枚举

左侧枚举管理下，进行枚举定义：

业务对象 (3)

新增

测试节点

test

销售区域定义la

salesareala

销售预订单la

salespreorderla

查询对象

枚举管理

新增枚举

编码/名称

序号	编码	名称	URI	类型
1	salesarealevel	区域级别	ypdtrain.ypdtrain.salesarealevel	枚举-字符

共 1 条 < 1 > 10 条 跳至 页

在数据建模-枚举页签中，点击新增枚举，录入数据并保存

1、销售渠道类型 XX saleschanneltypeXX 字符

枚举值	名称	说明
1	代销	
2	直销	

同样方式新建枚举销售渠道

名称

销售渠道类型

编码

saleschannellype

类型

字符

枚举值或名称

新增

序号	枚举值(key)	名称(value)
1	1	代销
2	2	直销

删除

2、预订状态 orderstate

枚举值	名称	说明
1	正常	
2	已预定	
3	关闭	

名称

预订状态

编码

orderstate

类型

字符

枚举值或名称

新增

序号	枚举值(key)	名称(value)
1	1	正常
2	2	已预订
3	3	关闭

删除

3、是否枚举

名称

是否通用枚举

编码

yesno

类型

字符

枚举值或名称

新增

序号	枚举值(key)	名称(value)
1	Y	是
2	N	否

2.3 新建业务对象

对象建模-业务对象，点击新增，录入业务对象信息，点击确定保存

新增业务对象

名称

销售预订单la

编码

salespreorderla

业务编码请保持全局唯一

描述

请输入

取消

确定

2.4 新建实体

2.4.1 核心功能介绍

业务对象下新增实体



界面展示如下



界面核心功能描述如下：

- 实体名称/编码：必填，不可重复；
- 父实体：可选项，下拉选择，可选当前应用已发布的实体名称；
- 表名：当前实体对应的数据库表名；
- 描述信息：可选项，对实体进行简要描述；
- 继承：元数据模型取自于类图，含有继承关系。广义上来说，当一个元数据继承另一个元数据时，被继承的元数据就可以称之为基类，此处可选择创建的基类元数据并继承它的相关属性。
- 引用接口：可选项，接口包括自动编码、审批、档案、业务流、树型结构、交易类型、主组织等接口；

编码	名称	作用
ypdBusiObjbase	基础可选项（ypd）	提供一些基本字段，id,pubts,dr 等
ypdBusiObjCode	自动编码（ypd）	选取后可进行编码，配合编码规则使用
ypdBusiObjApprovalFlow	审批（ypd）	添加审批字段，可在生成的 ui 模板上进行审批
ypdBusiObjEnable	档案（ypd）	是否是档案
ypdBusiObjBusinessFlow	业务流（ypd）	可在生成的 ui 模板上进行推拉单
ypdBusiObjITree	树型结构（ypd）	可生成树页面，记录上下级关系
ypdBusiObjOrg	主组织（ypd）	配合主组织权限使用
busiObjBusinessType	交易类型（ypd）	Ui 模板可以参照交易类型

➤ 实体的属性：

属性编码/名称：可手动修改，首次编辑属性名称时编码会生成对应的拼音字段；

属性类型：默认类型为文本，可根据需要进行重新选择；

常见属性类型：

- 单选引用：当类型为单选引用时，可选实体范围：平台已发布参照、当前租户已购买应用的参照、协同表单发布的参照、自建应用中发布的参照等；
- 单选：配置枚举，可选的枚举范围为：平台已发布枚举和租户自建枚举；
- 表字段名：可录入业务要求表字段名；
- 标签：可多选，从参照中选择，无默认值，从右侧操作面板中进行操作。

- 系统属性：包括基类属性+特性的属性两部分内容，作为系统属性显示在“系统属性”页签，仅查看，不提供编辑和删除功能。系统属性与勾选的引用接口有关，根据选择的接口分成基础可选项、审批、业务流、交易类型等八种，每一种下面包含多个系统属性，可以展开查看。

2.4.2 添加主实体

点击“新增实体”

基本信息：

编码：salespreorderXX

名称：销售预订单 XX

数据库表：xx_salespreorder

引用接口：审批、业务流、自动编码

配置完成后如下图

新增实体

名称

销售预订单la

编码

salespreorderla

父实体

表名

la_salespreorder

描述

继承

受控场景

请选择

引用接口

☒ 自动编码 (ypd)

☒ 审批 (ypd)

☐ 档案 (ypd)

☒ 业务流 (ypd)

☐ 树型结构 (ypd)

☐ 主组织 (ypd)

☐ 交易类型 (ypd)

选择更多

业务属性

系统属性

唯一性校验

非空校验

场景支持

索引

编码	名称	类型	引用	表字段名	标签
<div>+</div> ypdBusiObjCode	自动编码 (ypd)				
<div>+</div> ypdBusiObjApprovalFlow	审批 (ypd)				
<div>+</div> ypdBusiObjBusinessFlow	业务流 (ypd)				
<div>+</div> ypdBusiObjbase	基础可选项 (ypd)				

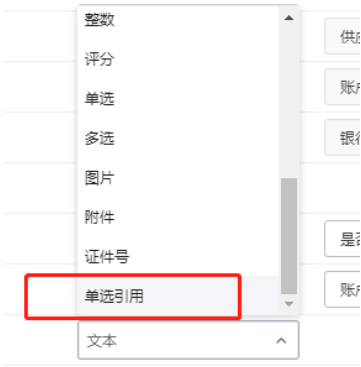
实体属性字段如下

字段编码	字段名称	类型	数据库列名
salesOrg	销售组织	单选引用：销售组织	sales_org
salesarea	销售区域	单选引用：销售区域定义	salesarea
orderdate	订货日期	日期	orderdate
expiredate	失效日期	日期	expiredate
businessman	业务员	单选引用：员工	businessman
salesdept	销售部门	单选引用：部门	salesdept
customer	客户	单选引用：客户档案	customer
settlecust	开票客户	单选引用：客户	settlecust
currtype	币种	单选引用：币种	currtype
totalnum	总数量	数值	totalnum
total	价税合计	数值	total
orderstate	预订状态	单选：预订状态	orderstate
memo	备注	文本	memo
saleschanneltype	销售渠道类型	单选：销售渠道类型	saleschanneltype
agent	代销公司	文本	agent
agentlevel	代销级别	单选：区域级别	agentlevel
vfile	自定义特征	自定义项特征	vfile

1、添加常规属性

12	<div>memo</div>	备注	<div>文本</div>	memo	
13	<div>totalnum</div>	总数量	<div>数值</div>	totalnum	<div>详情</div> <div>标签</div> <div>输入</div> <div>删除</div>
14	<div>total</div>	价税合计	<div>数值</div>	total	

2、配置参照



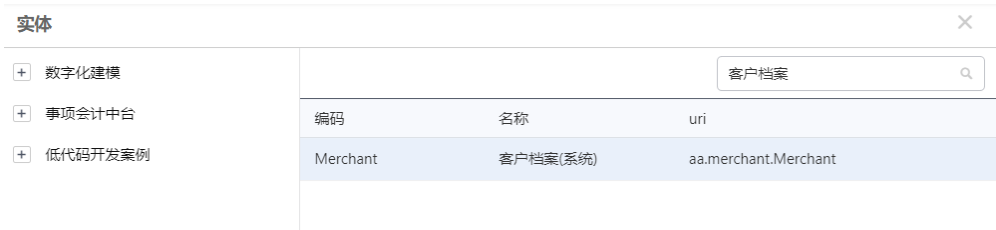
销售组织参照【销售组织】



销售区域参照【自定义参照：销售区域定义】



客户、开票客户参照“客户档案”



业务员参照“员工”



销售部门参照“部门”

实体

数字化建模

事项会计中台

低代码开发案例

部门

编码	名称	uri
DeptType	部门性质(系统)	org.doc.DeptType
AdminOrgVO	部门(系统)	bd.adminOrg.AdminOrgVO

币种参照“币种”

实体

数字化建模

事项会计中台

低代码开发案例

币种

编码	名称	uri
currencyacc	币种精度(系统)	bd.currencyacc.currencyacc
CurrencyTenantVO	币种(系统)	bd.currencytenant.CurrencyTenan...

3、配置枚举

选择“单选”，同时选择上步骤添加的枚举“销售渠道类型”。

枚举

dev1126

systemEnum

销售渠道

编码	名称	uri
saleschanneltype	销售渠道类型	dev1126.dev1126.saleschanneltype

最终效果如下图

业务属性	系统属性	唯一性校验	非空校验	场景支持	索引	编码/名称	快速新增	批量删除	批量设置类型	新增
序号	编码	名称	类型	引用	表字段名	标签				
1	salesOrg	销售组织	单选引用	销售组织(系统)	sales_org					
2	salesarea	销售区域	单选引用	销售区域定义(租户)	salesarea					
3	orderdate	订货日期	日期		orderdate					
4	expiredate	失效日期	日期		expiredate					
5	businessman	业务员	单选引用	员工(系统)	businessman					详细配置 标签 插入 删除
6	salesdept	销售部门	单选引用	部门(系统)	salesdept					
7	customer	客户	单选引用	客户档案(系统)	customer					
8	settlecust	开票客户	单选引用	客户档案(系统)	settlecust					
9	currtype	币种	单选引用	币种(系统)	currtype					
10	totalnum	总数量	数值		totalnum					详细配置 标签 插入 删除
11	total	价税合计	数值		total					
12	orderstate	预订状态	单选	订单状态	orderstate					
13	memo	备注	大文本		memo					
14	saleschanneltype	销售渠道类型	单选	销售渠道类型	saleschanneltype					

低代码案例1126								
数据建模 页面建模 流程&自动化 集成配置 发布管理								
实体 查询 枚举								
新增实体 批量发布 导入 导出								
序号	编码	名称	URI	类型	状态	描述	标签	
1	salespreorder	销售预订单	devcase1126.devcase1126.salespreorder	实体	已发布		报表实体	

4、自定义特征配置

17	vfile	自定义特征	自定义项特征组	vfile	详细配置 标签 插入 删除
----	-------	-------	---------	-------	---------------

2.4.3 添加子实体

选中主实体，新增子实体

业务对象 (3)								
实体 场景支持								
隐藏停用实体 编码/名称								
导出								
编码/名称	序号	编码	名称	URI	类型	状态	描述	标签
测试节点 test	1	salespreorderla	销售预订单la	ypdtrain0824.ypdtrain0824.salespreord...	实体	编辑	新建子实体	新建平行表 停用 查看 标签 更多
销售区域定义la salesareala								
销售预订单la salespreorderla								

基本信息：

编码：preorderdetailXX

名称：销售预订单明细 XX

数据库表：xx_preorderdetail

引用接口：业务流

* 名称	销售预订单明细la	* 编码	preorderdetaila	父实体	销售预订单la	* 表名	la_preorderdetail
描述							
引用接口	<input checked="" type="checkbox"/> 业务流 (ypd)	<input type="checkbox"/> 树型结构 (ypd)	<input type="checkbox"/> 核算信息 (ypd)	<input type="checkbox"/> 期间类 (ypd)	<input type="checkbox"/> 物料信息 (ypd)	<input type="checkbox"/> 分摊分配业务财务事务 (ypd)	<input type="checkbox"/> 分摊分配成本对象 (ypd)
选择更多							

实体属性字段如下

字段编码	字段名称	类型	数据库列名
materialId	物料编码	单选引用：物料	material_id
materialname	物料名称	文本	materialname
num	数量	数值	num
unit	单位	单选引用：物料计量单位	unit
price	单价	数值	price
discount	单品折扣	数值	discount
faxrate	税率	数值	faxrate
fax	税额	数值	fax
money	价税合计	数值	money
plansenddate	计划发货日期	日期	plansenddate
lineclose	是否行关闭	单选：是否枚举	lineclose

1、配置物料参照

+

数字化建模

+

事项会计中台

+

低代码开发案例

物料

编码	名称	uri
UnitGroup	物料计量单位分组(系统)	pc.unit.UnitGroup
TagClass	物料标签分类(系统)	pc.tag.TagClass
Tag	物料标签(系统)	pc.tag.Tag
ManagementClass	物料分类(系统)	pc.cls.ManagementClass
ProductSKU	物料SKU(系统)	pc.product.ProductSKU
Unit	物料计量单位(系统)	pc.unit.Unit
Product	物料(系统)	pc.product.Product
ProductTpl	物料模板(系统)	pc.tpl.ProductTpl
GoodsProductsCompar...	货位物料对照(系统)	aa.goodsposition.GoodsProducts...

取消

确定

配置单位参照

+

数字化建模

+

事项会计中台

+

低代码开发案例

物料计量单位

编码	名称	uri
UnitGroup	物料计量单位分组(系统)	pc.unit.UnitGroup
Unit	物料计量单位(系统)	pc.unit.Unit

完成后效果如下图

业务属性

系统属性

唯一性校验

非空校验

场景支持

索引

物料/名称

快速新增

批量删除

批量设置类型

新增

序号	编码	名称	类型	引用	表字段名	标签
1	materialid	物料编码	单选引用	物料(系统)	material_id	
2	materialname	物料名称	文本		materialname	
3	num	数量	数值		num	
4	unit	单位	单选引用	物料计量单位(系统)	unit	
5	price	单价	数值		price	<div>详细配置</div> <div>标签</div> <div>插入</div> <div>删除</div>
6	discount	单品折扣	数值		discount	
7	faxrate	税率	数值		faxrate	
7	fax	税额	数值		fax	
8	money	价税合计	数值		money	
9	plansenddate	计划发货日期	日期		plansenddate	<div>详细配置</div> <div>标签</div> <div>插入</div> <div>删除</div>
10	lineclose	行关闭	单选	是否枚举	lineclose	

在底部新增

10

行

确定新增

取消

保存并发布

保存

保存后，发布元数据

实体

查询

枚举

发布成功

新增实体

批量发布

导入

导出

序号	编码	名称	URI	描述	标签
1	salesorder	销售订单	devcase1126.devcase1126.salesorder	实体	已发布
	salesorderdetail	销售订单明细	devcase1126.devcase1126.salesorder...	实体	已发布

编辑

新建子实体

停用

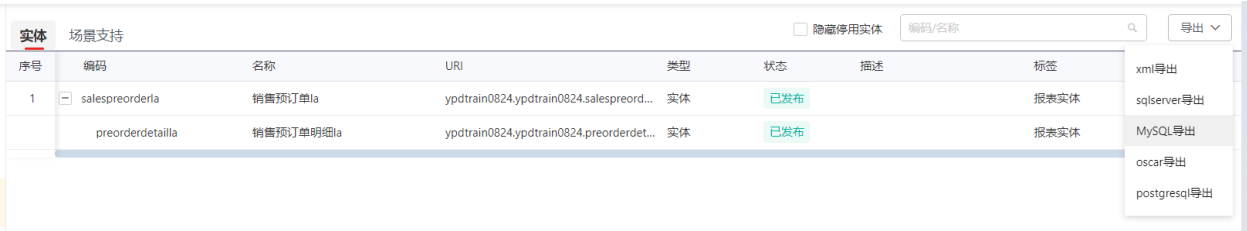
查看

标签

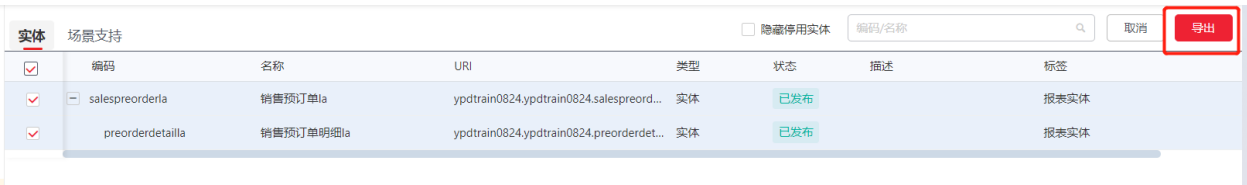
更多

2.4.4 执行脚本

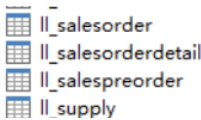
执行 SQL 脚本，找到库示例，可批量导出 SQL 并执行



选择主子实体，点击“导出”



导出销售预订单主表、子表 SQL，在对应的业务库表中执行，执行 SQL 刷新库表即可开始进行页面设计了



2.4.5 场景支持配置

按照当前业务实际场景，配置场景支持



2.5 特征配置

2.5.1 业务对象配置

数据模型添加如下属性



场景勾选自定义项

实体 场景支持			
序号	场景名称	启用	描述
1	打印	<input checked="" type="checkbox"/>	打印场景，当前业务对象支持进行打印模板、打印控制等相关功能配置。
2	自定义项	<input checked="" type="checkbox"/>	用于自定义项分配
3	交易类型	<input checked="" type="checkbox"/>	交易类型场景，支持租户级基于业务的不同性质及管理要求，对同一业务对象下的业务场景进行的细分分类的定义，可作用于流程、模板等。
4	审批流	<input checked="" type="checkbox"/>	审批流场景，支持对业务对象单据配置审批流程，做到业务单据与业务审批流程联动。
5	电子合同	<input type="checkbox"/>	电子合同场景，支持业务对象数据字段参与电子合同编制，成为销售合同、采购合同、人力合同等合同主体的编制内容。
6	业务流起始单据	<input type="checkbox"/>	业务流程比如采购订货(普通采购)流程，流程以核心单据采购订单起始，流程id会从核心起始单据一直向下游携带。
7	报表	<input checked="" type="checkbox"/>	报表场景，支持对业务对象配置报表，用表格、图表等格式来动态显示数据。
8	UI模板	<input checked="" type="checkbox"/>	UI模板场景，支持UI模板左边树基于业务对象下统一管理，并能支持原厂预置、用户自定义交易类型的多模板个性化扩展。
9	编码规则	<input checked="" type="checkbox"/>	编码规则场景，支持业务对象实体字段编号的生成规则的定义和维护。
10	业务流	<input checked="" type="checkbox"/>	业务流场景，支持对业务对象单据进行业务数据流转的规则配置，如：转换规则等。
11	消息模板	<input checked="" type="checkbox"/>	消息模板场景，支持对业务对象定义系统发送的各种消息的不同展现样式，包括消息标题和消息内容、消息。
12	规则引擎	<input checked="" type="checkbox"/>	规则引擎是一种嵌入在应用中的组件，使用预先定义好的语义规范来实现剥离出来的业务规则，通过接受输入的环境和业务数据，进行业务评估，执行业务动作，做出业务决策。

2.5.2 特征实体表配置

一旦实体中引用了特种组，那么特征实体中就会有特征实体，标准版无需建表即可使用，而专业版就必须预置表到对应引擎数据库了。

具体预置过程如下。

1、打开特征实体列表。

业务对象 (9)

查询对象

枚举管理

特征实体

编码/名称

新增实体

序号	编码	名称	URI	类型	状态	描述	标签
1	preorderdetailmavm...	销售预订单明细_物料选配特征实体	ypdtrain0824.ypdtrain0824.preorderdet...	实体	已发布		
2	preorderdetailmavm...	销售预订单明细_物料特征组实体	ypdtrain0824.ypdtrain0824.preorderdet...	实体	已发布		
3	salespreordermavfile...	销售预订单ma_自定义特征实体	ypdtrain0824.ypdtrain0824.salespreord...	实体	已发布		全局
4	salesordermavfileUs...	销售订单ma_自定义实体	ypdtrain0824.ypdtrain0824.salesorderm...	实体	已发布		全局

点击编辑，修改表名，保存后发布

编辑实体

* 名称

销售预订单ma_自定义特征实体

描述

* 编码

salespreordermavfileUserDefine

父实体

* 表名

ma_presalesdef

继承

业务属性

序号	编码	名称	类型	引用	表字段名	标签
1	productid	物料id	文本		sysCol2	
2	characteristicsid	特征id	文本		sysCol1	
3	id	id	文本		id	
4	pubts	pubts	日期时间		pubts	
5	def1	客户事项	文本		vc01	
6	def2	客户接洽日期	日期		dcol1	
7	def4	销售客户详情	文本		vc02	
8	def5	销售预测日期	日期		dcol2	

1、创建实体表

需要创建三种表，结构是“表名称_数字（1 到 3）”，例如，ma_salesorderdef_1, ma_salesorderdef_2, ma_salesorderdef_3，表的结构一样，就是表名称序号不一样而已，同时注意，还修改表的索引。

```
CREATE TABLE IF NOT EXISTS `ma_salesorderdef_3` (
```

```
  `id` varchar(64) NOT NULL,
  `object_id` varchar(64) DEFAULT NULL,
  `sysCol0` varchar(255) DEFAULT NULL,
  `sysCol1` varchar(255) DEFAULT NULL,
  `sysCol2` varchar(255) DEFAULT NULL,
  `sysCol3` varchar(255) DEFAULT NULL,
  `sysCol4` varchar(255) DEFAULT NULL,
  `vcol1` varchar(255) DEFAULT NULL,
  `vcol2` varchar(255) DEFAULT NULL,
  `vcol3` varchar(255) DEFAULT NULL,
  `vcol4` varchar(255) DEFAULT NULL,
  `vcol5` varchar(255) DEFAULT NULL,
  `vcol6` varchar(255) DEFAULT NULL,
  `vcol7` varchar(255) DEFAULT NULL,
  `vcol8` varchar(255) DEFAULT NULL,
  `vcol9` varchar(255) DEFAULT NULL,
  `vcol10` varchar(255) DEFAULT NULL,
  `vcol11` varchar(255) DEFAULT NULL,
  `vcol12` varchar(255) DEFAULT NULL,
  `vcol13` varchar(255) DEFAULT NULL,
  `vcol14` varchar(255) DEFAULT NULL,
  `vcol15` varchar(255) DEFAULT NULL,
  `vcol16` varchar(255) DEFAULT NULL,
  `vcol17` varchar(255) DEFAULT NULL,
  `vcol18` varchar(255) DEFAULT NULL,
  `vcol19` varchar(255) DEFAULT NULL,
  `vcol20` varchar(255) DEFAULT NULL,
  `vcol21` varchar(255) DEFAULT NULL,
  `vcol22` varchar(255) DEFAULT NULL,
  `vcol23` varchar(255) DEFAULT NULL,
  `vcol24` varchar(255) DEFAULT NULL,
  `vcol25` varchar(255) DEFAULT NULL,
  `vcol26` varchar(255) DEFAULT NULL,
  `vcol27` varchar(255) DEFAULT NULL,
  `vcol28` varchar(255) DEFAULT NULL,
  `vcol29` varchar(255) DEFAULT NULL,
  `vcol30` varchar(255) DEFAULT NULL,
  `vcol31` varchar(255) DEFAULT NULL,
```

```

`vcol32` varchar(255) DEFAULT NULL,
`vcol33` varchar(255) DEFAULT NULL,
`vcol34` varchar(255) DEFAULT NULL,
`vcol35` varchar(255) DEFAULT NULL,
`vcol36` varchar(255) DEFAULT NULL,
`vcol37` varchar(255) DEFAULT NULL,
`vcol38` varchar(255) DEFAULT NULL,
`vcol39` varchar(255) DEFAULT NULL,
`vcol40` varchar(255) DEFAULT NULL,
`vcol41` varchar(255) DEFAULT NULL,
`vcol42` varchar(255) DEFAULT NULL,
`vcol43` varchar(255) DEFAULT NULL,
`vcol44` varchar(255) DEFAULT NULL,
`vcol45` varchar(255) DEFAULT NULL,
`vcol46` varchar(255) DEFAULT NULL,
`vcol47` varchar(255) DEFAULT NULL,
`vcol48` varchar(255) DEFAULT NULL,
`vcol49` varchar(255) DEFAULT NULL,
`vcol50` varchar(255) DEFAULT NULL,
`vcol51` varchar(255) DEFAULT NULL,
`vcol52` varchar(255) DEFAULT NULL,
`vcol53` varchar(255) DEFAULT NULL,
`vcol54` varchar(255) DEFAULT NULL,
`vcol55` varchar(255) DEFAULT NULL,
`ncol1` decimal(24,8) DEFAULT NULL,
`ncol2` decimal(24,8) DEFAULT NULL,
`ncol3` decimal(24,8) DEFAULT NULL,
`ncol4` decimal(24,8) DEFAULT NULL,
`ncol5` decimal(24,8) DEFAULT NULL,
`ncol6` decimal(24,8) DEFAULT NULL,
`ncol7` decimal(24,8) DEFAULT NULL,
`ncol8` decimal(24,8) DEFAULT NULL,
`ncol9` decimal(24,8) DEFAULT NULL,
`ncol10` decimal(24,8) DEFAULT NULL,
`ncol11` decimal(24,8) DEFAULT NULL,
`ncol12` decimal(24,8) DEFAULT NULL,
`ncol13` decimal(24,8) DEFAULT NULL,
`ncol14` decimal(24,8) DEFAULT NULL,
`ncol15` decimal(24,8) DEFAULT NULL,
`ncol16` decimal(24,8) DEFAULT NULL,
`ncol17` decimal(24,8) DEFAULT NULL,
`ncol18` decimal(24,8) DEFAULT NULL,

```



```

`ncol19` decimal(24,8) DEFAULT NULL,
`ncol20` decimal(24,8) DEFAULT NULL,
`ncol21` decimal(24,8) DEFAULT NULL,
`ncol22` decimal(24,8) DEFAULT NULL,
`ncol23` decimal(24,8) DEFAULT NULL,
`ncol24` decimal(24,8) DEFAULT NULL,
`ncol25` decimal(24,8) DEFAULT NULL,
`ncol26` decimal(24,8) DEFAULT NULL,
`ncol27` decimal(24,8) DEFAULT NULL,
`ncol28` decimal(24,8) DEFAULT NULL,
`ncol29` decimal(24,8) DEFAULT NULL,
`ncol30` decimal(24,8) DEFAULT NULL,
`ncol31` decimal(24,8) DEFAULT NULL,
`ncol32` decimal(24,8) DEFAULT NULL,
`ncol33` decimal(24,8) DEFAULT NULL,
`ncol34` decimal(24,8) DEFAULT NULL,
`ncol35` decimal(24,8) DEFAULT NULL,
`ncol36` decimal(24,8) DEFAULT NULL,
`ncol37` decimal(24,8) DEFAULT NULL,
`ncol38` decimal(24,8) DEFAULT NULL,
`ncol39` decimal(24,8) DEFAULT NULL,
`ncol40` decimal(24,8) DEFAULT NULL,
`ncol41` decimal(24,8) DEFAULT NULL,
`ncol42` decimal(24,8) DEFAULT NULL,
`ncol43` decimal(24,8) DEFAULT NULL,
`ncol44` decimal(24,8) DEFAULT NULL,
`ncol45` decimal(24,8) DEFAULT NULL,
`ncol46` decimal(24,8) DEFAULT NULL,
`ncol47` decimal(24,8) DEFAULT NULL,
`ncol48` decimal(24,8) DEFAULT NULL,
`ncol49` decimal(24,8) DEFAULT NULL,
`ncol50` decimal(24,8) DEFAULT NULL,
`lcol1` bigint(20) DEFAULT NULL,
`lcol2` bigint(20) DEFAULT NULL,
`lcol3` bigint(20) DEFAULT NULL,
`lcol4` bigint(20) DEFAULT NULL,
`lcol5` bigint(20) DEFAULT NULL,
`lcol6` bigint(20) DEFAULT NULL,
`lcol7` bigint(20) DEFAULT NULL,
`lcol8` bigint(20) DEFAULT NULL,
`lcol9` bigint(20) DEFAULT NULL,
`lcol10` bigint(20) DEFAULT NULL,

```

```
`lcol11` bigint(20) DEFAULT NULL,  
`lcol12` bigint(20) DEFAULT NULL,  
`lcol13` bigint(20) DEFAULT NULL,  
`lcol14` bigint(20) DEFAULT NULL,  
`lcol15` bigint(20) DEFAULT NULL,  
`lcol16` bigint(20) DEFAULT NULL,  
`lcol17` bigint(20) DEFAULT NULL,  
`lcol18` bigint(20) DEFAULT NULL,  
`lcol19` bigint(20) DEFAULT NULL,  
`lcol20` bigint(20) DEFAULT NULL,  
`lcol21` bigint(20) DEFAULT NULL,  
`lcol22` bigint(20) DEFAULT NULL,  
`lcol23` bigint(20) DEFAULT NULL,  
`lcol24` bigint(20) DEFAULT NULL,  
`lcol25` bigint(20) DEFAULT NULL,  
`lcol26` bigint(20) DEFAULT NULL,  
`lcol27` bigint(20) DEFAULT NULL,  
`lcol28` bigint(20) DEFAULT NULL,  
`lcol29` bigint(20) DEFAULT NULL,  
`lcol30` bigint(20) DEFAULT NULL,  
`lcol31` bigint(20) DEFAULT NULL,  
`lcol32` bigint(20) DEFAULT NULL,  
`lcol33` bigint(20) DEFAULT NULL,  
`lcol34` bigint(20) DEFAULT NULL,  
`lcol35` bigint(20) DEFAULT NULL,  
`lcol36` bigint(20) DEFAULT NULL,  
`lcol37` bigint(20) DEFAULT NULL,  
`lcol38` bigint(20) DEFAULT NULL,  
`lcol39` bigint(20) DEFAULT NULL,  
`lcol40` bigint(20) DEFAULT NULL,  
`lcol41` bigint(20) DEFAULT NULL,  
`lcol42` bigint(20) DEFAULT NULL,  
`lcol43` bigint(20) DEFAULT NULL,  
`lcol44` bigint(20) DEFAULT NULL,  
`lcol45` bigint(20) DEFAULT NULL,  
`lcol46` bigint(20) DEFAULT NULL,  
`lcol47` bigint(20) DEFAULT NULL,  
`lcol48` bigint(20) DEFAULT NULL,  
`lcol49` bigint(20) DEFAULT NULL,  
`lcol50` bigint(20) DEFAULT NULL,  
  
`dcol1` datetime DEFAULT NULL,
```

```
`dcol2` datetime DEFAULT NULL,  
`dcol3` datetime DEFAULT NULL,  
`dcol4` datetime DEFAULT NULL,  
`dcol5` datetime DEFAULT NULL,  
`dcol6` datetime DEFAULT NULL,  
`dcol7` datetime DEFAULT NULL,  
`dcol8` datetime DEFAULT NULL,  
`dcol9` datetime DEFAULT NULL,  
`dcol10` datetime DEFAULT NULL,  
`dcol11` datetime DEFAULT NULL,  
`dcol12` datetime DEFAULT NULL,  
`dcol13` datetime DEFAULT NULL,  
`dcol14` datetime DEFAULT NULL,  
`dcol15` datetime DEFAULT NULL,  
`dcol16` datetime DEFAULT NULL,  
`dcol17` datetime DEFAULT NULL,  
`dcol18` datetime DEFAULT NULL,  
`dcol19` datetime DEFAULT NULL,  
`dcol20` datetime DEFAULT NULL,  
`dcol21` datetime DEFAULT NULL,  
`dcol22` datetime DEFAULT NULL,  
`dcol23` datetime DEFAULT NULL,  
`dcol24` datetime DEFAULT NULL,  
`dcol25` datetime DEFAULT NULL,  
`dcol26` datetime DEFAULT NULL,  
`dcol27` datetime DEFAULT NULL,  
`dcol28` datetime DEFAULT NULL,  
`dcol29` datetime DEFAULT NULL,  
`dcol30` datetime DEFAULT NULL,  
`dcol31` datetime DEFAULT NULL,  
`dcol32` datetime DEFAULT NULL,  
`dcol33` datetime DEFAULT NULL,  
`dcol34` datetime DEFAULT NULL,  
`dcol35` datetime DEFAULT NULL,  
`dcol36` datetime DEFAULT NULL,  
`dcol37` datetime DEFAULT NULL,  
`dcol38` datetime DEFAULT NULL,  
`dcol39` datetime DEFAULT NULL,  
`dcol40` datetime DEFAULT NULL,  
`dcol41` datetime DEFAULT NULL,  
`dcol42` datetime DEFAULT NULL,  
`dcol43` datetime DEFAULT NULL,
```

```
`dcol44` datetime DEFAULT NULL,  
`dcol45` datetime DEFAULT NULL,  
`dcol46` datetime DEFAULT NULL,  
`dcol47` datetime DEFAULT NULL,  
`dcol48` datetime DEFAULT NULL,  
`dcol49` datetime DEFAULT NULL,  
`dcol50` datetime DEFAULT NULL,  
`bcol1` tinyint(1) DEFAULT NULL,  
`bcol2` tinyint(1) DEFAULT NULL,  
`bcol3` tinyint(1) DEFAULT NULL,  
`bcol4` tinyint(1) DEFAULT NULL,  
`bcol5` tinyint(1) DEFAULT NULL,  
`bcol6` tinyint(1) DEFAULT NULL,  
`bcol7` tinyint(1) DEFAULT NULL,  
`bcol8` tinyint(1) DEFAULT NULL,  
`bcol9` tinyint(1) DEFAULT NULL,  
`bcol10` tinyint(1) DEFAULT NULL,  
`bcol11` tinyint(1) DEFAULT NULL,  
`bcol12` tinyint(1) DEFAULT NULL,  
`bcol13` tinyint(1) DEFAULT NULL,  
`bcol14` tinyint(1) DEFAULT NULL,  
`bcol15` tinyint(1) DEFAULT NULL,  
`bcol16` tinyint(1) DEFAULT NULL,  
`bcol17` tinyint(1) DEFAULT NULL,  
`bcol18` tinyint(1) DEFAULT NULL,  
`bcol19` tinyint(1) DEFAULT NULL,  
`bcol20` tinyint(1) DEFAULT NULL,  
`bcol21` tinyint(1) DEFAULT NULL,  
`bcol22` tinyint(1) DEFAULT NULL,  
`bcol23` tinyint(1) DEFAULT NULL,  
`bcol24` tinyint(1) DEFAULT NULL,  
`bcol25` tinyint(1) DEFAULT NULL,  
`bcol26` tinyint(1) DEFAULT NULL,  
`bcol27` tinyint(1) DEFAULT NULL,  
`bcol28` tinyint(1) DEFAULT NULL,  
`bcol29` tinyint(1) DEFAULT NULL,  
`bcol30` tinyint(1) DEFAULT NULL,  
`bcol31` tinyint(1) DEFAULT NULL,  
`bcol32` tinyint(1) DEFAULT NULL,  
`bcol33` tinyint(1) DEFAULT NULL,  
`bcol34` tinyint(1) DEFAULT NULL,  
`bcol35` tinyint(1) DEFAULT NULL,
```

```
`bcol36` tinyint(1) DEFAULT NULL,
`bcol37` tinyint(1) DEFAULT NULL,
`bcol38` tinyint(1) DEFAULT NULL,
`bcol39` tinyint(1) DEFAULT NULL,
`bcol40` tinyint(1) DEFAULT NULL,
`bcol41` tinyint(1) DEFAULT NULL,
`bcol42` tinyint(1) DEFAULT NULL,
`bcol43` tinyint(1) DEFAULT NULL,
`bcol44` tinyint(1) DEFAULT NULL,
`bcol45` tinyint(1) DEFAULT NULL,
`bcol46` tinyint(1) DEFAULT NULL,
`bcol47` tinyint(1) DEFAULT NULL,
`bcol48` tinyint(1) DEFAULT NULL,
`bcol49` tinyint(1) DEFAULT NULL,
`bcol50` tinyint(1) DEFAULT NULL,
`operator` varchar(255)  DEFAULT NULL,
`domain` varchar(255)  DEFAULT NULL,
`ytenant_id` varchar(64)  NOT NULL,
`pubts` datetime DEFAULT NULL,
`create_date` datetime DEFAULT NULL,
`update_date` datetime DEFAULT NULL,
`dr` smallint(6) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `ma_salesorderdef_3_update_date_IDX` (`update_date`) USING BTREE
) ENGINE=InnoDB ;
```

特征实体 sql 下载

 特征实体sql.txt

2.5.3 特征分类定义

新增特征分类

首页

特征分类

应用构建

自定义项分配

默认方案

编码

名称

状态

重置

高级

新增

删除

	编码	名称	启用状态	排序	备注	
<input type="checkbox"/>	salesbusiness	销售业务特征	启用	0		

2.5.1 特征定义

新增特征，如下图

首页

分配任务列表

特征分类

应用构建

特征

sales1 只看必填项

特征编码

sales1

数据类型

文本

备注

特征名称

销售客户详情

录入长度

255

特征别名

特征分类

销售业务特征

首页

分配任务列表

特征分类

应用构建

特征

salespredate 只看必填项

特征编码

salespredate

数据类型

日期

特征名称

销售预测日期

特征分类

销售业务特征

特征别名

备注

2.5.2 特征分配

启用后点击分配

编码	名称	数据类型	状态	特征分类	
sales1	销售客户详情	文本	启用	销售业务特征	编辑 停用 删除 分配 取消分配 »
salespredate	销售预测日期	日期	启用	销售业务特征	

选择“销售预订单”节点，勾选后点击保存，成功后即可使用

注意:

特征分配时，能够勾选表头是因为做了两个步骤，第一在实体上特征组，选择了自定义项特征组，第二在场景支持上勾选了自定义项，否则表头无法勾选。

分配效果在右上角可查看到成功消息

首页分配任务列表特征特征应用构建

默认方案状态特征名称数据类型成功新增

编码	名称	数据类型	状态	特征分类	
def4	销售客户详情	文本	启用	销售业务特征	
def5	销售预测日期	日期	启用	销售业务特征	
def3	客户业务区域	基本档案	启用	销售业务特征	
def2	客户接洽日期	日期	启用	销售业务特征	
def1	客户事项	文本	启用	--	

点击消息可查看详情

首页分配任务列表特征特征应用构建

默认方案状态任务类型分配类型成功查看详情

序号	ID	描述	创建者	状态	开始时间	结束时间	更新时间	任务类型	
1	15448502585833553...	当前操作为分配:具体信息:自定义项信息,特征Code: def4:处...	joydev	成功	2022-09-13 12:44:02	2022-09-13 12:44:02	2022-09-13 12:44:02	分配	
2	15448484890568294...	当前操作为分配:具体信息:自定义项信息,特征Code: def5:处...	joydev	成功	2022-09-13 12:40:37	2022-09-13 12:40:37	2022-09-13 12:40:37	分配	
3	15448451561622077...	当前操作为分配:具体信息:自定义项信息,特征Code: def1:处...	joydev	成功	2022-09-13 12:36:22	2022-09-13 12:36:23	2022-09-13 12:36:23	分配	

分配成功后，在页面生成可以看到了，如下图可录入，同时业务流支持配置下推

返回000006开立案

编辑复制提交下推全局联查销售预订单打印打印删除变更历史查询

销售信息

销售组织销售事业部销售部门业务1销售区域东部

销售渠道类型代销

代销信息

代销公司8888代销级别重点

预订信息

编码000006订货日期2022-09-13失效日期2022-09-16

业务员开心的JOY币种人民币总数量8,888.00

价税合计0.00预订状态关闭单据状态开立案

客户事项注意事项123客户接洽日期2022-09-21销售客户详情客户接洽正常,预期如常

销售预测日期2022-09-22

备注

客户信息

客户开票客户

销售预订单明细

批改增行删行

序号	物料编码	物料名称	单位	数量	单价	单品折扣	税率	物料特征组	物料选配特征	税额	
1	001000002	手机	Centimeters	8,888.00	8,888.00	--	--			--	
合计				8,888.00						0.00	

创建时间2022-09-13 19:57:41修改时间2022-09-22 14:02:22创建人joydev修改人joydev

3 页面建模

页面建模提供应用的功能页面设计器、组合页面设计器、报表设计器，支持不同功能业务页面的设计，同时支持前后端扩展，实现单据页面、画布页面、业务事件的构建和交互驱动过程的个性化扩展。

3.1 新建页面

单据设计器提供 7 种预置模板和多种布局容器和组件，支持表格、表单、网格、树等多种页面的组装能力：

预置模板：提供单表、单卡、左树右表、左树右卡、树形表、一主多子、主子孙表 7 个常用模板。



点击新增页面，选择“一主多子”，选择元数据为上步骤生成的“预订单”



点击下一步可以对界面进行业务配置。列表显示效果可以选择“主表列表”或者“主子表列表”，如果需要在列表显示主子数据，则可选择“主子表列表”，如下图

新建页面

示例页面



一主多子

* 页面名称

销售预订单

* 页面编码

salespreorderla

元数据

销售预订单la

PC子表展示

☒ 页签

☐ 平铺

子表元数据

PC端

销售预订单明细la ×

其他设置

☒ 同时生成列表

☐ 主表列表

☒ 主子表列表

上一步

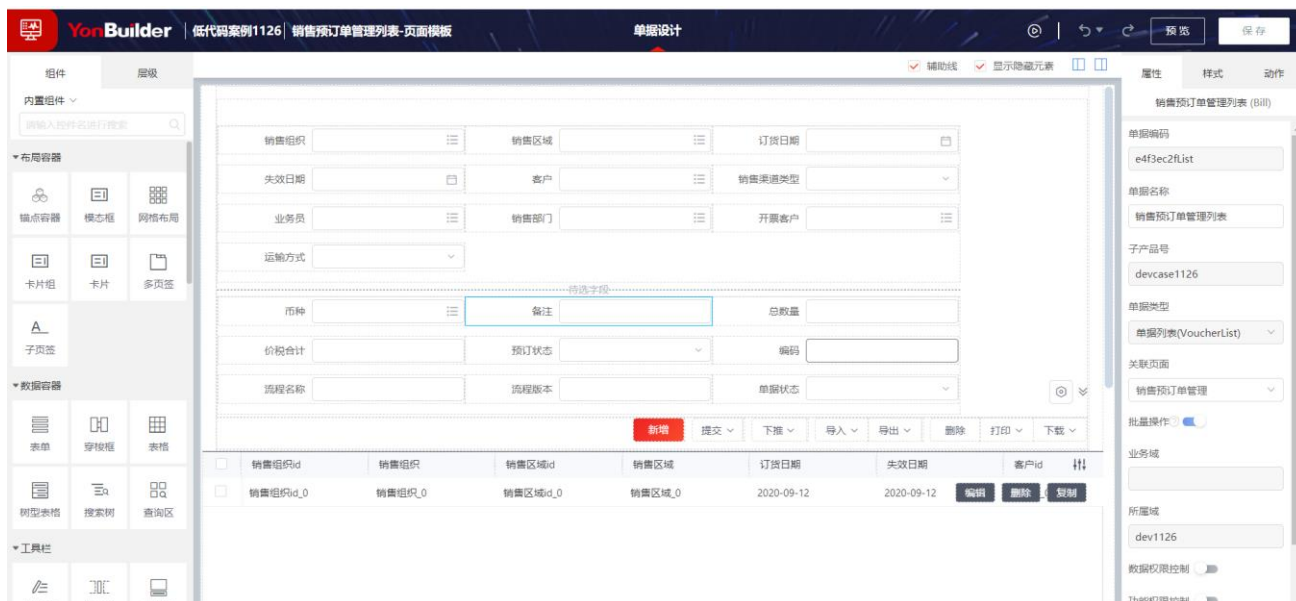
完成

点击完成。



3.2 页面布局说明

新建好的页面，我们可以打开设计器，配置界面显示效果



模板由三个区域构成：组件区、设计区和属性区。

- 组件区：提供构成页面的元素，包括布局容器、数据容器、工具栏、按钮控件、基础控件、业务控件、协作控件等。
- 设计区：单据页面的设计区域。列表模板包括列表头部区域、查询区域、表格区域、列表底部栏按钮区。卡片模板包括卡片头部按钮区、卡片底部按钮区、数据显示区域等，根据需要来设计。
- 属性区：选中的页面元素的属性设置区域。

布局容器：构成页面布局的元素，系统提供了网格、卡片组和多页签三种布局容器，容器支持样式包括大小和位置，背景图边框颜色等属性设置；布局容器中拖入组件控件来实现单据页面的组装。

1. 布局容器：定义单据页面中的用于布局类的控件样式。
2. 数据容器：定义页面模板中常用于数据展示的界面控件。
 - 1) 表单：卡片容器中常用样式，可作为分组的容器；
 - 2) 表格：列表常用数据批量化展示的容器，卡片中常用于子表中的数据展示。
 - 3) 树型表格/搜索树：常用于树类型样式模板。
3. 工具栏：定义单据页面中的操作按钮区域及按钮区域的样式。
4. 按钮控件：定义单据页中按钮区域中的按钮，包括按钮和下拉按钮两个控件，可定义按钮类型、规格类型及单击按钮的动作，包括 新增、编辑、删除、提交、撤回、启用、停用等。
5. 基础控件：定义单据页面的基础控件，包括本框、多行文本、富文本、多语文本、手机、电话、邮箱、证件号、数值、金额、选项、开关、日期框、日期时间、时间框、评分、地图定位、图片上传等。

业务控件：定义单据页面中需要的业务类控件，包括附件管理、参照框、自动编码、下推按钮、打印、描述信息等。

6. 协作控件：与协作套件相关的控件，包括协作套件和沟通套件。
7. 业务控件：与业务中台企业业务支撑组件对接及展示的相关控件。

3.3 列表界面配置

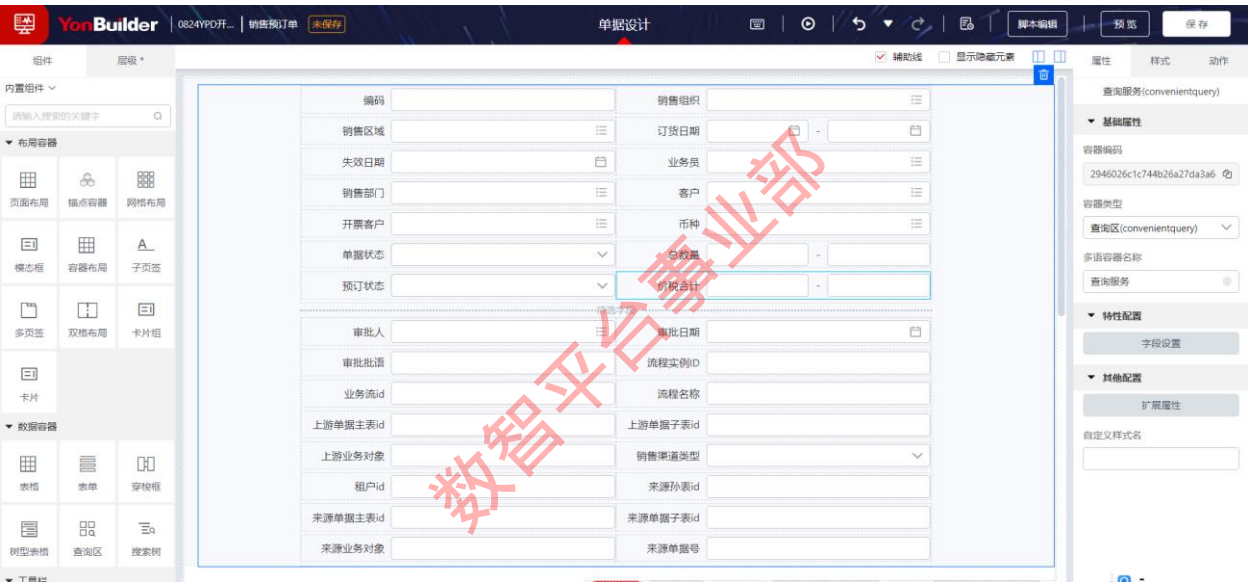
3.3.1 调整查询区

3.3.1.1 配置查询条件

业务需求：

- 1、 调整核心业务信息到查询 “选择区域”
- 2、 调整订货日期查询条件为 “区间”

配置效果如下图所示



列表，点击预览

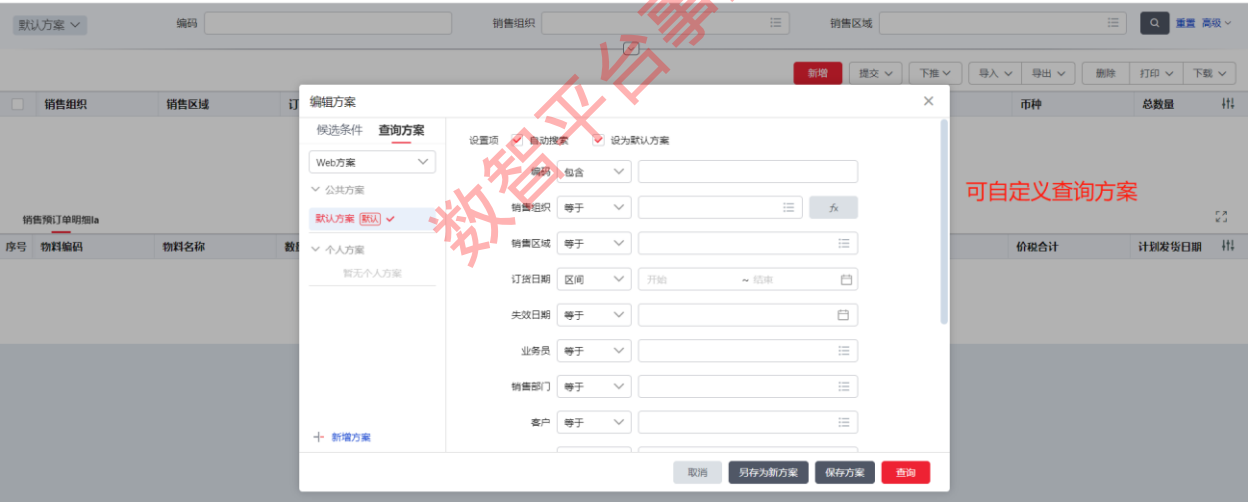


查询区界面效果如下图

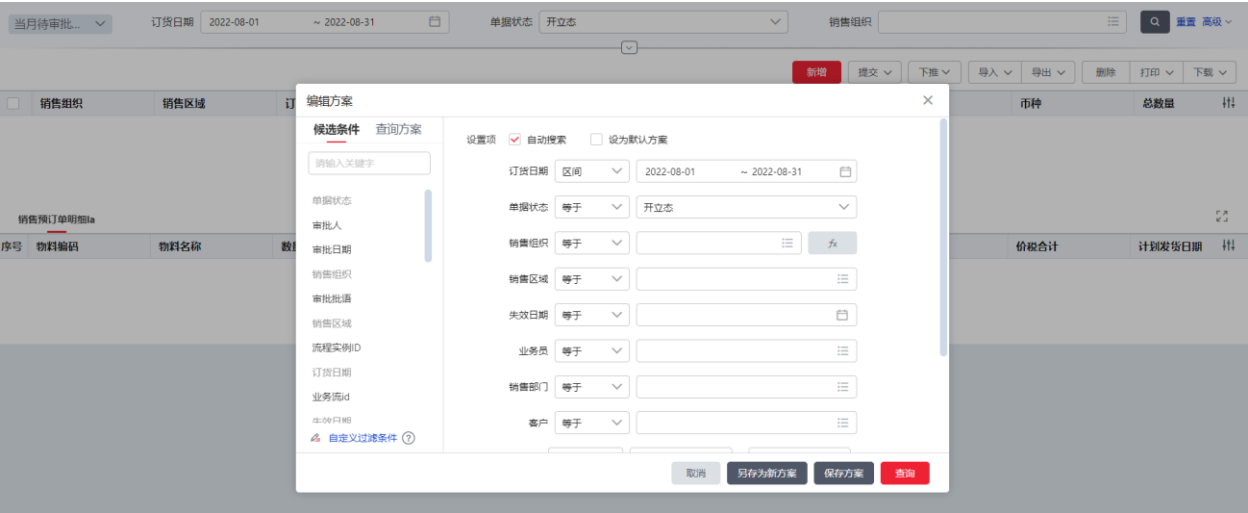


3.3.1.2 自定义查询方案

在预览下，使用用户可直接点击“高级”可自定义查询方案



可配置查询方案，保存即可



可配置查询条件“销售组织”显示为导航区。
预览时可切换查询方案为“当月待复核单据”效果如下图



3.3.2 列表界面样式调整

3.3.2.1 业务需求

- 1、 调整字段顺序：按照业务要求调整字段顺序；
- 2、 配置固定列：配置编码、销售组织在列表区域固定；
- 3、 调整主表数量、价税合计显示合计；
- 4、 配置子表物料显示为物料编码；
- 5、 调整子表单品折扣、税率为百分数；
- 6、 调整子表数量、税额、价税合计显示合计；

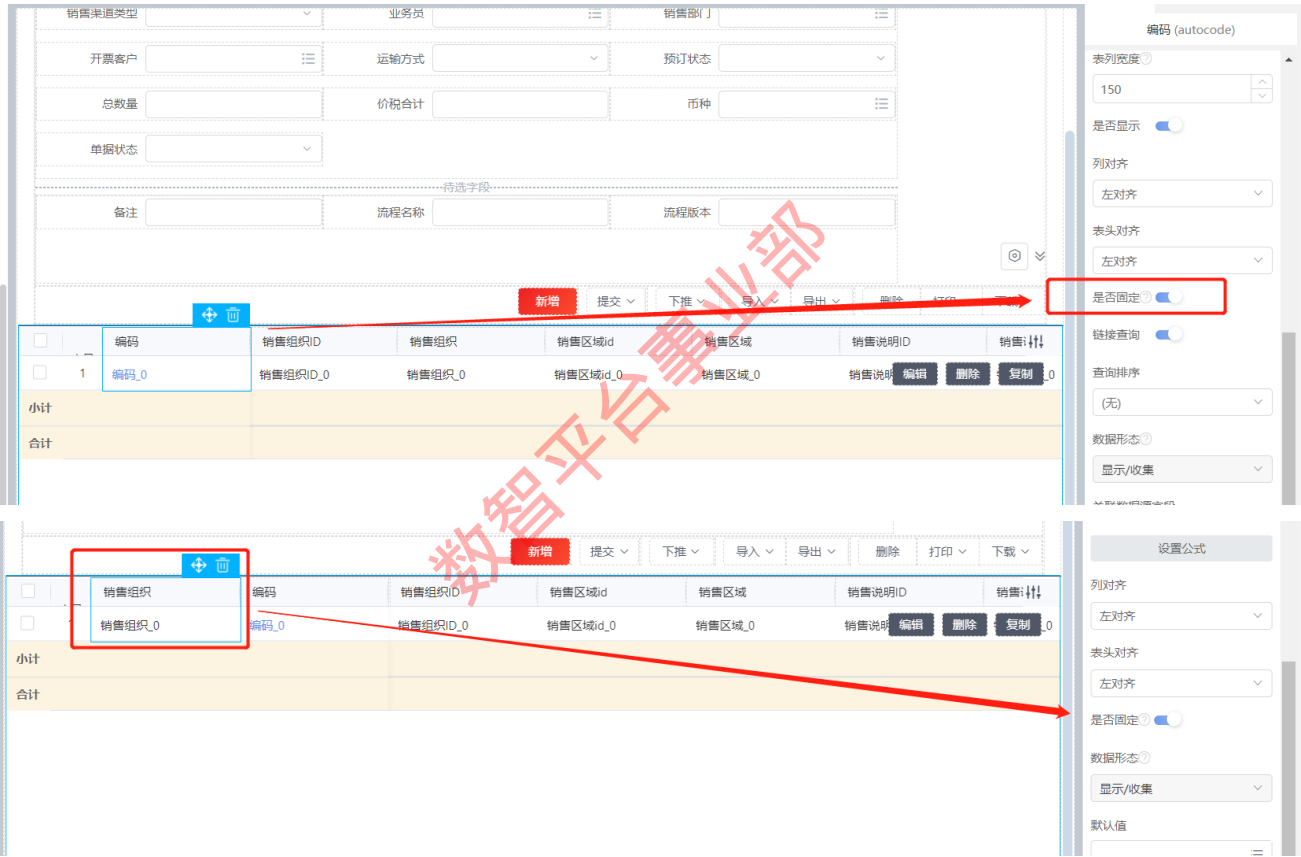
3.3.2.2 配置实现

主表页面调整：

- 1、 配置显示序号



- 2、通过界面拖拽实现顺序调整
- 3、配置固定列



- 4、配置合计

上游单据主表id

上游单据子表id

上游业务对象

销售渠道类型

租户id

来源孙表id

来源单据主表id

来源单据子表id

来源业务对象

来源单据号

新增

提交

下推

导入

导出

删除

打印

下载

业务处理

序号	编码	销售组织	总数量	价税合计	预订状态	销售渠道类型	备注
1	编码_0	销售组织_0	总数量_0	价税合计_0	预订状态_0	全局联查	编辑删除复制

编码: cSumType

只有启用"是"合计列时, 此属性才生效。

缩放取值

表格配置

列对齐

左对齐

居中

右对齐

链接查询

查询排序

(无)

升序

降序

是否固定

检查

表头对齐

左对齐

居中

右对齐

是否合计

合计方式

未指定类型

其他配置

5、 百分数调整

单品折扣

税率

税额

价税合计

计划发货日期

单品折扣

数据形态

显示/收集

必传

表列宽度

150

格式类型

自定义

整数

百分比

常用配置

系统级隐藏

6、 子表配置物料参照显示编码

公共领域

本领域

基础服务

物料参照 (过滤主组织)

物料参照 (过滤主组织)

通用物料参照

物料查询参照

物料参照 (过滤主组织)

参照配置

基本信息

参照名称

通用物料参照

参照编码

productcenter.pc_nomalproductref

参照类型

树表参照

* 参照显示

物料编码 code

参照可用字段

主键 id

可带入的字段

物料编码ID materialid

点击添加一行

配置单位参照

参照配置

公共领域

本领域

基础服务

计量单位

基本信息

参照名称 计量单位

参照编码 productcenter.pc_unitref

参照类型 表参照

*参照显示

名称 name

参照可用字段

可带入的字段

参照带入

主键 id

单位id unit

点击添加一行

3.3.3 应用效果

列表展示页面

默认方案 编码 销售组织 销售区域

新增 提交 下推 导入 导出 删除 打印 下载 业务处理

序号	编码	销售组织	币种	总数量	价税合计	预订状态	销售渠道类型	备注	单据状态	出
1	000001	销售事业部	人民币	10.00	10,000.00	正常	代销	--	开立态	
小计				10.00	10,000.00					
合计				10.00	10,000.00					

共 1 条 << 1 >> 20 条 跳至 页

预订单明细

序号	物料编码	物料名称	数量	单位	单价	单品折扣	税率	税额	价税合计	计划发货日期	出
1	--	--	100.00	--	2,000.00	25.00%	10.00%	1,000.00	1,000,000.00	--	
合计								1,000.00	1,000,000.00		

3.4 卡片界面配置

3.4.1 业务需求

卡片页面为浏览和编辑功能，界面需求描述如下：

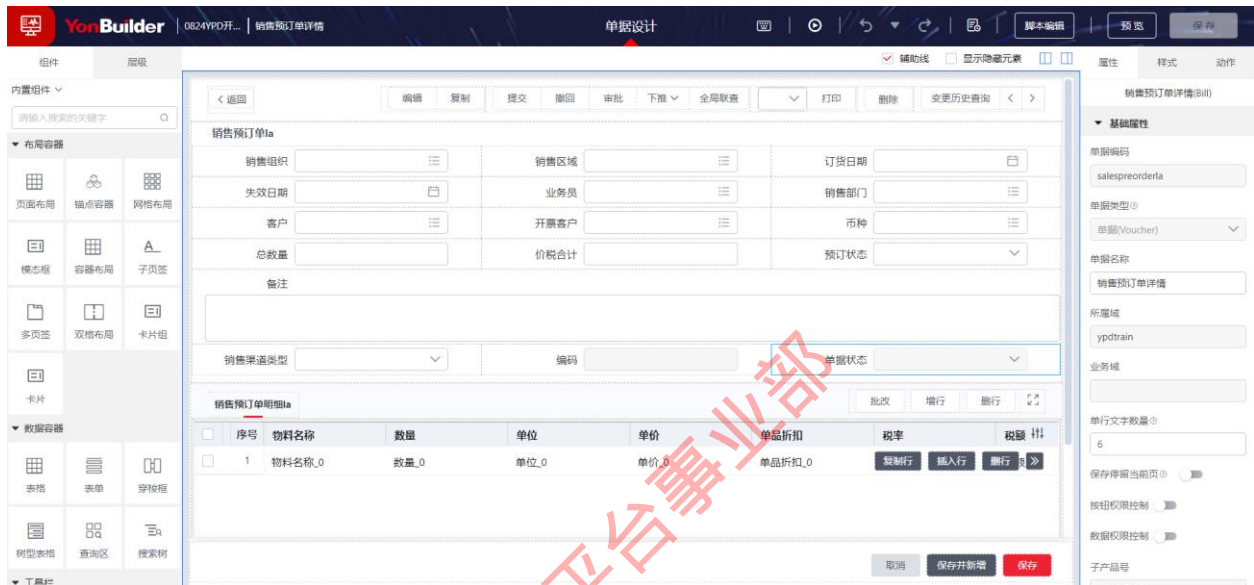
- 1、按照需求调整界面样式，按业务分区；
- 2、调整是否可编辑属性；
- 3、添加界面默认值；
- 4、配置参照过滤及带出

3.4.2 主表界面配置

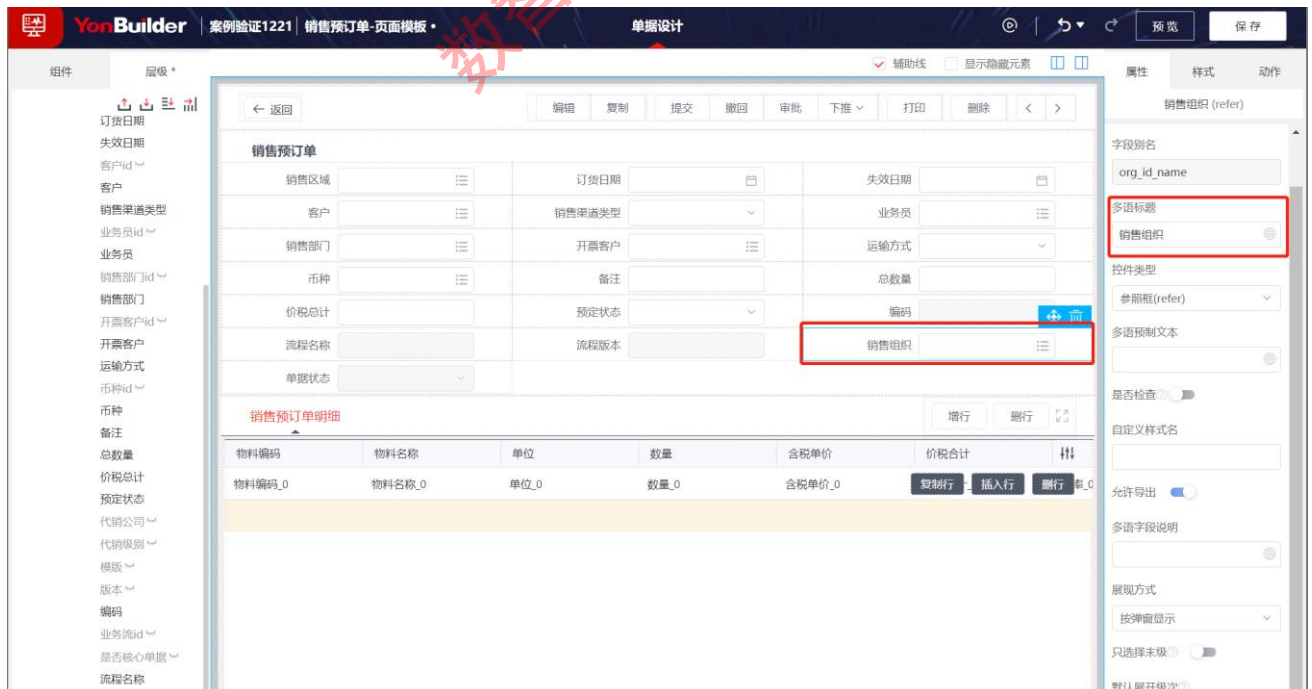
在详情页面点击“设计”



显示卡片详情页设计器

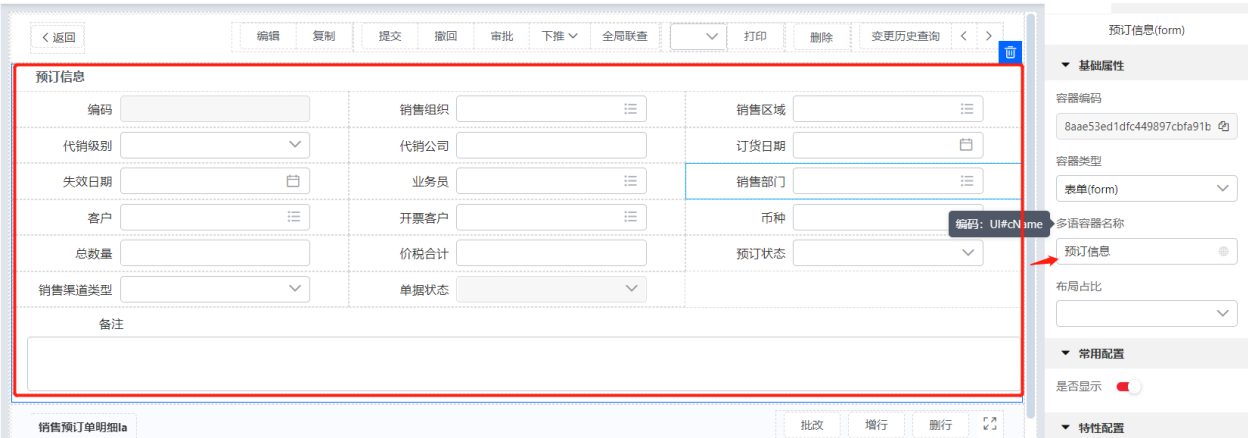


修改组织名称为“销售组织”

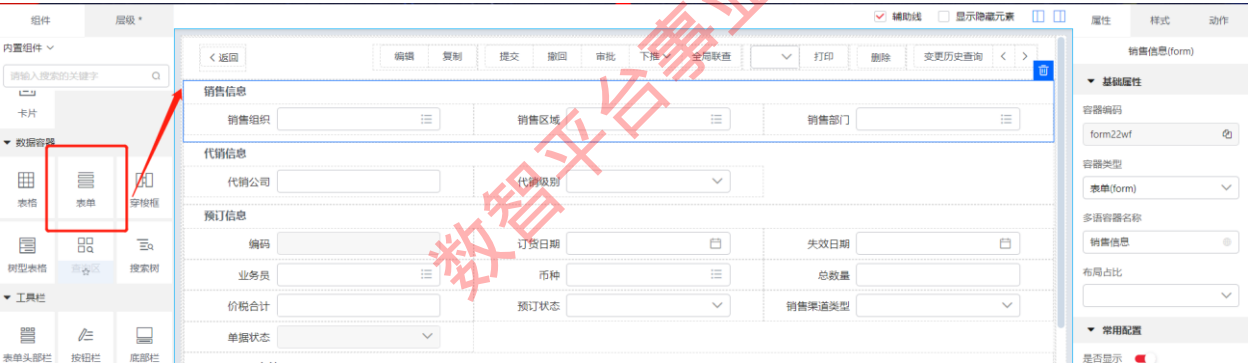


3.4.2.1 调整业务分组

- 1) 修改表单名称为“预订信息”
修改容器名称



- 2) 添加表单分区
添加表单信息：“客户信息”、“销售信息”、“代销信息”。
在组件区域，拖拽“表单”到卡片组，插入三个表单。



- a) 修改多语容器名称为“销售信息”，同时拖拽销售组织、销售部门、销售区域、销售渠道类型字段到当前表单
b) “代销信息”，包含代销公司、代销级别字段
c) “客户信息”，包含客户、开票客户字段

最终效果如下图

< 返回

编辑复制提交撤回审批下推√全局联查√打印删除

变更历史查询<>

+

销售信息

销售组织

销售区域

销售部门

销售渠道类型

代销信息

代销公司

代销级别

预订信息

编码

订货日期

失效日期

业务员

币种

总数量

价税合计

预订状态

单据状态

备注

客户信息

客户

开票客户

销售信息(form)

基础属性

容器编码

form22wf

容器类型

表单(form)

多语容器名称

销售信息

布局占比

常用配置

是否显示

特性配置

字段设置

其他配置

多语布局说明

取消

保存并新增

保存

3.4.2.2 调整编辑控制

- 1、调整字段顺序
- 2、配置必输项，销售组织、销售部门、订货日期、失效日期、币种、客户
- 3、配置不可编辑
配置总数量、价税总计、预订状态不可编辑
- 4、配置默认值
销售渠道类型配置默认为“直销”，预订状态默认“正常”

配置预订状态默认值、不允许复制

代货级别

订货日期

失效日期

币种

总数量

预订状态

单据状态

显示

允许为空

允许修改

允许导出

不允许复制①

编码: cDefaultValue

默认值

正常

设置公式

如配置销售组织、销售区域显示效果为下拉

返回

编辑

复制

提交

撤回

审批

下推

全局联查

▼

打印

删除

变更历史查询

<

>

销售组织信息

销售组织

销售区域

销售部门

销售渠道类型

代销信息

代销公司

代销级别

预订信息

编码

订货日期

失效日期

业务员

币种

总数量

价税合计

预订状态

单据状态

备注

销售组织(refer)

参照框(refer) ▼

多语标题

销售组织

数据形态①

显示/收集 ▼

是否必传② ☒

布局占比③

一列 二列 三列 整行

展现方式④

按弹窗显示 按下拉显示

参照配置

常用配置

点击参照配置，可配置下拉时显示字段

参照配置

公共领域

本领域

基本档案

销售组织 (职能)

参照配置

数据过滤

* 参照显示

组织名称 name

参照可用字段

主键 id

参照带入

销售组织id salesOrg

显示名称

组织编码 code

组织名称 name

参照名称

销售组织 (职能)

参照编码

ucf-org-center.bd_salesorg

参照类型

树参照

点击添加一行

取消

确定

完成后效果如下图

000002

开立态

只看必填项

销售信息

销售组织

销售区域

销售部门

代销信息

未选择

代销公司

预订信息

02 销售事业部

编码

业务员

价税合计

订货日期

币种

预订状态

失效日期

总数量

销售渠道类型

销售区域参照配置

公共领域

本领域

销售区域定义la详情(参照)

参照配置

数据过滤

*参照显示

名称 name

参照可用字段

可带入的字段

主键 id

销售区域id salesarea

点击添加一行

显示名称

编码 code x

名称 name x

负责人 areamaster_name (areamas... x

参照名称

销售区域定义la详情(参照)

参照编码

ypdtrain.RefTable_bdf07cf334

参照类型

树参照

显示效果如下图

销售信息

销售组织

销售渠道类型

未选择

代销信息

代销公司

02 销售事业部

销售区域

代销级别

03 华北区域 jack

02 西部区域 jack

01 东部区域 nick

销售部门

订货日期

3.4.3 子表界面配置

- 1、配置参照：配置商品编码，参照显示“物料编码”
先配置参照为“多行带入”

价税合计

预订状态

单据状态

备注

客户信息

* 客户

开票客户

物料编码(refer)

允许为空

允许修改

允许导出

不允许复制

默认值

设置公式

特性配置

多行带入

默认展开级次

展开至二级

只选择末级

物料编码

物料名称

数量

单位

单价

单品折扣

物料编码_0

物料名称_0

数量_0

单位_0

单价_0

0.00

配置参照



参照带出字段“物料名称”，“物料单位”主键以及显示名称



- 2、配置物料编码、物料名称固定
- 3、配置数值字段
数量：配置合计
价税合计：配置合计
税额：配置合计
- 4、配置编辑属性
价税合计不可编辑，单位不可编辑

3.4.4 应用效果

点击新增时，录入数据后

000001 开立志 只看必填项

销售信息

* 销售组织 销售事业部

销售区域 华北区域

* 销售部门 销售事业部

销售渠道类型 代销

代销信息

代销公司

代销级别

预订信息

编码 000001

* 订货日期 2022-08-26

失效日期 2022-09-11

* 业务员 joydev

* 币种 人民币

总数量 10.00

价税合计 10,000.00

预订状态 正常

单据状态 开立志

备注

客户信息

* 客户 华北制药

开票客户

预订单明细

序号 物料编码 物料名称 数量 单位 单价 单品折扣 税率 税额 价税合计 计划发货日期

1 01000001 台式机 100.00 台 2,000.00 25.00% 10.00% 1,000.00 1,000,000.00 2022-08-28

取消

保存并新增

保存

保存后效果

< 返回 000001 开立志

编辑

复制

提交

下推

全局联查

打印

删除

变更历史查询

销售信息

销售组织 销售事业部

销售区域 华北区域

销售部门 销售事业部

销售渠道类型 代销

代销信息

代销公司 --

代销级别 --

预订信息

编码 000001

订货日期 2022-08-26

失效日期 2022-09-11

业务员 joydev

币种 人民币

总数量 10.00

价税合计 10,000.00

预订状态 正常

单据状态 开立志

备注 --

客户信息

客户 华北制药

开票客户 --

预订单明细

序号 物料编码 物料名称 数量 单位 单价 单品折扣 税率 税额 价税合计 计划发货日期

1 01000001 台式机 100.00 台 2,000.00 25.00% 10.00% 1,000.00 1,000,000.00 2022-08-28

合计 100.00 1,000.00 1,000,000.00

创建时间 2022-08-26 10:20:30

修改时间 2022-08-26 14:41:09

创建人 joydev

修改人 joydev

42

3.5 页面规则功能配置

常见界面交互，如根据某字段取值，配置哪些字段是否可编辑态或者默认值等，可通过此功能来实现。

3.5.1 业务场景

业务需求：

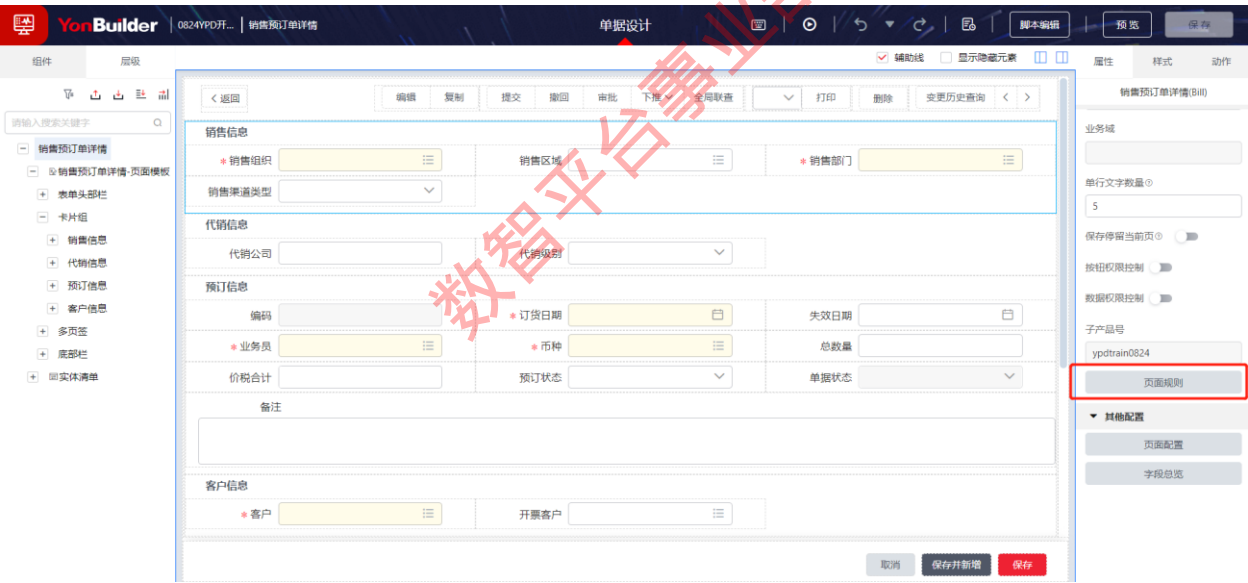
- 1、 当销售渠道类型为“代销”，则“代销公司”字段不能为空；

设计分析：

- 1、 简单业务规则可以通过页面设计器提供的页面规则功能实现页面字段监听联动效果；
- 2、 校验规则可支持配置在保存时简单校验规则，同时配置提示信息效果。

3.5.2 页面规则

选中卡片容器，下方“页面规则”



点击添加，选择校验规则页签



1、配置销售渠道类型保存校验

页面规则

* 规则名称

销售渠道校验

* 规则编码

saleschannelCheck

* 触发时机

保存

动作1

取消

保存

前置条件

添加

逻辑符	提示方式	字段/公式	操作符	值类型	值	
	字段	销售渠道类型	等于	常量	代销	删
并且	字段	代销公司	等于	常量		删

执行操作

控制方式	提示方式	提示信息
禁止下一步	弹窗	销售渠道类型为“代销”，代销公司不能为空

配置完成后效果如下图

页面规则

交互规则

校验规则

添加

序号	规则名称	规则编码	页面状态	触发时机	状态
1	销售渠道校验	saleschannelCheck	--	保存	启用

2、提交校验失效日期应该小于当前日期

页面规则

* 规则名称

失效日期提校验

* 规则编码

expireValid

* 触发时机

提交

动作1

取消

保存

前置条件

添加

逻辑符	提示方式	字段/公式	操作符	值类型	值	
	字段	失效日期	小于等于	系统变量	当前日期	重置

执行操作

控制方式	提示方式	提示信息
禁止下一步	弹窗	预订单失效日期小于等于当前日期，无法提交审批，请检查

3.5.3 应用效果

1、列表页面点击新增，默认为直销，代销信息不可见

← 返回

销售信息

销售组织

销售部门

销售区域

销售渠道类型

直销

预订信息

编码

000003

业务员

总数量

备注

流程名称

订货日期

运输方式

公路运输

价税合计

流程版本

失效日期

币种

预订状态

单据状态

开立态

客户信息

客户

开票客户

预订详情

增行

删行

✖

✓	序号	物料编码	物料名称	单位	数量	含税单价	价税合计	税率	税额	单品折扣	计划	出
✓	1	--	--	--	--	--	--	--	--	--	2021	▼

取消

保存并新增

保存

切换销售渠道类型为“代销”

← 返回

销售信息

销售组织

销售部门

销售区域

销售渠道类型

代销

代销信息

代销公司

代销级别

预订信息

编码

000003

业务员

总数量

备注

订货日期

运输方式

公路运输

价税合计

失效日期

币种

预订状态

如果代销公司为空，则保存弹出提示

销售渠道类型

代销

代销信息

代销公司

预订信息

⚠

销售渠道类型为“代销”，代销公司不能为空

取消

确定

2、列表选择直销数据，点击编辑

默认方案 销售组织 编码 销售区域 2021-12-01~2021-12-06 新增 提交 下推 导入 导出 删除 打印 下载

序号	编码	销售组织	销售区域	客户	订货日期	失效日期	销售渠道类型	业务员	销售部门	开票客户
1	000002	--	西部区域	联想创业	2021-12-03	2021-12-25	直销	mary	创新中心	联想创业
2	000001	--	苏杭区域	联想实业集团	2021-12-01	2021-12-03	代销	jack	创新中心	联想创业
小计										
合计										

效果如下图

← 返回

销售信息

* 销售组织

销售渠道类型

直销

* 销售部门

创新中心

销售区域

西部区域

预订信息

编码

000002

* 订货日期

2021-12-03

* 失效日期

2021-12-25

业务员

mary

运输方式

公路运输

* 币种

人民币

总数量

价税合计

预订状态

备注

流程名称

流程版本

单据状态

开立志

列表选择代销数据，点击编辑

默认方案

销售组织

编码

销售区域

2021-12-01~2021-12-06

新增

提交

下推

导入

导出

删除

打印

下载

序号	编码	销售组织	销售区域	客户	订货日期	失效日期	销售渠道类型	业务员	销售部门	开展客户
1	000002	--	西部区域	联想创业	2021-12-03	2021-12-25	直销	mary	创新中心	联想创业
2	000001	--	苏杭区域	联想实业集团	2021-12-01	2021-12-03	代销	jack	创新中心	
小计										
合计										

界面效果如下图

← 返回

销售信息

* 销售组织

--

销售渠道类型

代销

* 销售部门

创新中心

销售区域

苏杭区域

代销信息

代销公司

成果中实业

代销级别

一级

预订信息

编码

000001

* 订货日期

2021-12-01

* 失效日期

2021-12-03

业务员

jack

运输方式

铁路运输

* 币种

人民币

总数量

1,000.0000

价税合计

268,766,600.5500

预订状态

--

备注

3、浏览态数据查看
代销数据浏览

← 返回

编辑 复制 提交 审批 下推 打印 删除 < >

销售信息	销售组织 --	销售部门 创新中心	销售区域 苏杭区域
销售渠道类型	代销		
代销信息	代销公司 成果中实业	代销级别 一级	
预订信息	编码 000001	订货日期 2021-12-01	失效日期 2021-12-03
	业务员 jack	运输方式 铁路运输	币种 人民币
	总数量 1,000.0000	价税合计 268,766,600.5500	预订状态 --
	备注 --		
流程名称 --	流程版本 --	单据状态 开立志	

直销数据浏览

← 返回

编辑 复制 提交 审批 下推 打印 删除 < >

销售信息

销售组织 -- 销售部门 创新中心 销售区域 西部区域

销售渠道类型 直销

预订信息

编码 000002 订货日期 2021-12-03 失效日期 2021-12-25

业务员 mary 运输方式 公路运输 币种 人民币

总数量 -- 价税合计 -- 预订状态 --

备注 --

流程名称 -- 流程版本 -- 单据状态 开立态

客户信息

客户 联想创业 开票客户 联想创业

4、配置审批流点击提交

< 返回 000001 开立态

编辑 复制 提交 下推 全局联查 打印 删除 变更历史查询 < >

销售信息

销售组织 销售事业部 销售区域 华北区域 销售部门 销售事业部

销售渠道类型 代销

代销信息

代销公司 ABC 代销级别 --

预订信息

编码 000001 订货日期 2022-08-01 失效日期 2022-08-26

业务员 joydev 币种 人民币 总数量 10.00

价税合计 10,000.00 预订状态 正常 单据状态 开立态

备注 --

客户信息

客户 华北制药 开票客户 --

预订单明细

修改 增行 删行

序号	物料编码	物料名称	数量	单位	单价	单品折扣	税率	税额	价税合计	计划发货日期	操作
1	01000001	台式机	100.00	台	2,000.00	25.00%	10.00%	1,000.00	1,000,000.00	2022-08-28	
合计			100.00					1,000.00	1,000,000.00		

创建时间 2022-08-26 10:20:30 修改时间 2022-08-26 14:53:34 创建人 joydev 修改人 joydev

点击提交效果

< 返回 000001 开立态

编辑 复制 提交 下推 全局联查 打印 删除 变更历史查询 < >

销售信息

销售组织 销售事业部 销售区域 华北区域 销售部门 销售事业部

销售渠道类型 代销

代销信息

代销公司 ABC 代销级别 --

预订信息

编码 000001 订货日期 2022-08-01 失效日期 2022-08-26

业务员 joydev 币种 人民币 总数量 10.00

价税合计 10,000.00 预订状态 正常 单据状态 开立态

备注 --

客户信息

客户 华北制药 开票客户 --

预订单明细

修改 增行 删行

序号	物料编码	物料名称	数量	单位	单价	单品折扣	税率	税额	价税合计	计划发货日期	操作
1	01000001	台式机	100.00	台	2,000.00	25.00%	10.00%	1,000.00	1,000,000.00	2022-08-28	
合计			100.00					1,000.00	1,000,000.00		

创建时间 2022-08-26 10:20:30 修改时间 2022-08-26 14:53:34 创建人 joydev 修改人 joydev

3.6 参照业务功能配置

在业务场景中，参照经常会遇到级联参照，比如组织选择后，自动过滤下属部门；或者在选择参照数据时需要用到固定条件过滤。下面我们将通过两个小案例实现参照过滤功能

3.6.1 业务场景

- 1、销售部门需要根据销售组织过滤
- 2、子表物料参照显示编码，录入后自动带出物料名称、单位字段；

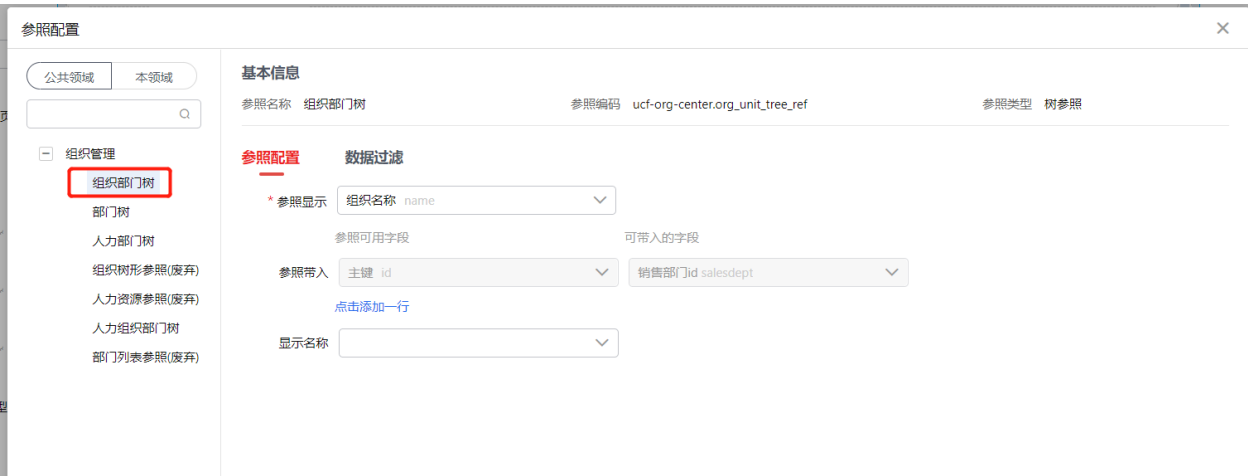
3.6.2 参照过滤

- 1、配置参照过滤

1) 打开卡片的页面设计器，选中“销售部门”参照，选择右侧“参照配置”功能；



按需求选择可使用参照，如下图选择参照为部门组织树



配置参照过滤，配置上级组织 ID 为销售组织

公共领域

本领域

Q

组织管理

组织部门树

部门树

人力部门树

组织树形参照(废弃)

人力资源参照(废弃)

人力组织部门树

部门列表参照(废弃)

基本信息

参照名称 组织部门树

参照编码 ucf-org-center.org_unit_tree_ref

参照类型 树参照

参照配置

数据过滤

添加

逻辑符	字段	操作符	值类型	值
	上级组织ID	等于	字段	销售组织id

点击确定保存数据，页面设计器保存，参照类别配置完毕

3.6.3 应用效果

预览打开界面，销售组织选择后，点击“销售部门”

销售信息

销售组织

销售渠道类型

预订信息

编码

业务员

总数量

备注

流程名称

客户信息

销售组织

sales 数智销售中心

取消

确定

销售区域

失效日期

币种

预订状态

单据状态

“销售部门”参照弹出后，数据仅为“销售组织”的部门信息。

000002 开立志 只看必填项

销售信息

销售组织

销售渠道类型

代销信息

代销公司

预订信息

编码

业务员

价税合计

备注

客户信息

客户

销售部门

未选择

0101 销售一部

0102 人力资源部

0103 销售二部

1001 财务部

取消

确定

销售部门

失效日期

总数量

单据状态

4 函数式在线脚本

基于应用构建平台的脚本引擎，支持用户完成自定义逻辑的脚本编写；类型分为前端脚本和后端脚本，无论前端还是后端脚本都是以 JS 为语言规范，其中操作元数据是以平台提供的 YonQL 对数据进行操作，YonQL 支持 SQL95 标准规范与绝大部分 SQL 编写方式。

函数式在线脚本分为前端函数、API 函数。

- 前端脚本：可用于控件状态控制、控件赋值、简单数据计算、openAPI 调用；对于原有功能的扩展，也可以在页面初始化中绑定；
- 后端函数、API 函数：可进行前端函数调用、openAPI 调用；

4.1 前端函数

前端函数一般做一些设置默认值，或者校验。对于原有功能的扩展，请在页面初始化中绑定。触发前事件，一般可以通过使用 beforeXXX 的方式实现

4.1.1 业务场景

1、关键字段默认值赋值

新增时预订状态默认为“正常”，订货日期默认为“当前日期”；失效日期默认为“订货日期”+3 天；

2、保存前校验

失效日期>订货日期

子表物料编码不能重复；

3、金额计算-创建通用函数

子表：

数量*单价*单品折扣*税率=税额；

数量*单价*单品折扣+税额=含税总价

主表：

主表含税总价=子表含税总价合计；

主表数量=子表数量合计

4、编辑监听

当销售渠道类型为“代销”时，则代销信息页签可见，“代销公司”字段可编辑；

当销售渠道类型为“直销”时，则代销信息页签不可见，“代销公司”字段取值清空；

4.1.2 卡片页面初始化

在页面设计器，卡片页面上，可以注册前端函数，添加前端事件监听。

4.1.2.1 新建函数

页面设计器上方点击“脚本编辑”



打开界面如下图



4.1.2.2 设置默认值

业务需求：

新增时预订状态默认为“正常”，订货日期默认为“当前日期”；失效日期默认为“订货日期”+3 天；脚本实现：判断是否新增状态，如果为新增态则设置日期为当前日期，需要对日期做格式化

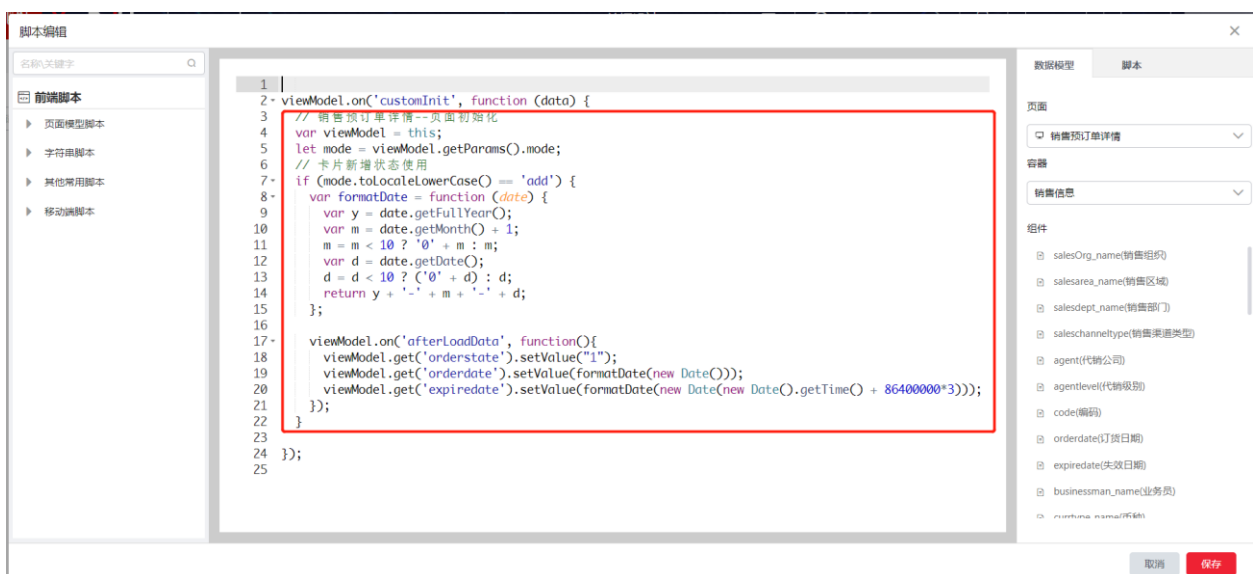

```

viewModel.on('customInit', function (data) {
    // 销售预订单详情--页面初始化
    var viewModel = this;
    let mode = viewModel.getParams().mode;
    // 卡片新增状态使用
    if (mode.toLocaleLowerCase() == 'add') {
        var formatDate = function (date) {
            var y = date.getFullYear();
            var m = date.getMonth() + 1;
            m = m < 10 ? '0' + m : m;
            var d = date.getDate();
            d = d < 10 ? ('0' + d) : d;
            return y + '-' + m + '-' + d;
        };

        viewModel.on('afterLoadData', function(){
            viewModel.get('orderstate').setValue("1");
            viewModel.get('orderdate').setValue(formatDate(new Date()));
            viewModel.get('expiredate').setValue(formatDate(new Date().getTime() + 86400000*3));
        });
    }
});

```

将函数填写到刚上一步骤创建的函数体中，效果如下图



点击保存，页面设计器上也需要点击保存。

验证效果：打开列表单据，点击新增，自动添加订货日期和失效日期。

000002 开立志 只看必填项

销售信息

* 销售组织

销售区域

* 销售部门

销售渠道类型

代销

代销信息

代销公司

代销级别

预订信息

编码

* 订货日期

失效日期

* 业务员

* 币种

总数量

价税合计

预订状态

单据状态

备注

客户信息

* 客户

开票客户

预订单明细

序号

物料编码

物料名称

数量

单位

单价

单品折扣

税率

税额

价税合计

计划发货日期

暂时没有数据哦~

取消

保存并新增

保存

4.1.2.3 保存前校验

打开在上一章节中创建的页面初始化前端函数，添加如下标注的代码注册到页面初始化前端函数
我们可直接在页面设计器右上角“脚本编辑”按钮打开

YonBuilder 0824YPD开... | 销售预订单详情 单据设计

组件

层级

销售预订单详情

销售预订单详情-页面模板

表单头部栏

卡片组

多页签

底部栏

显示体满单

销售信息

* 销售组织

销售区域

* 销售部门

销售渠道类型

代销信息

代销公司

代销级别

预订信息

编码

* 订货日期

失效日期

* 业务员

* 币种

总数量

价税合计

预订状态

单据状态

备注

客户信息

* 客户

开票客户

取消

保存并新增

保存

脚本编辑

预览

保存

销售预订单详情

页面初始化

编辑脚本

此时出现函数界面如下图

53



添加如下黄色部分代码

```
viewModel.on('customInit', function (data) {
    // 销售预订单详情--页面初始化
    var viewModel = this;
    let mode = viewModel.getParams().mode;
    // 卡片新增状态使用
    if (mode.toLocaleLowerCase() === 'add') {
        var formatDate = function (date) {
            var y = date.getFullYear();
            var m = date.getMonth() + 1;
            m = m < 10 ? '0' + m : m;
            var d = date.getDate();
            d = d < 10 ? ('0' + d) : d;
            return y + '-' + m + '-' + d;
        };

        viewModel.on('afterLoadData', function() {
            viewModel.get('orderstate').setValue("1");
            viewModel.get('orderdate').setValue(formatDate(new Date()));
            viewModel.get('expiredate').setValue(formatDate(new Date(new Date().getTime() + 86400000*3)));
        });
    }
});
```

// 定义函数：判断物料 ID 是否相同

```
const checkList = (params) => {
```

```
    let temp = {};
```

```
    for (let i in params) {
```

```
        let key = params[i].materialId;    //判断依据:物料 ID
```

```
        if (temp[key]) {  
            return false;  
        } else {  
            temp[key] = params[i];  
        }  
    }  
    return true;  
}  
  
// 保存前校验  
viewModel.on('beforeSave', function(data) {  
    var expiredate = viewModel.get('expiredate').getValue();  
    var orderdate = viewModel.get('orderdate').getValue();  
    //判断 dateA 是否大于 dateB，为 true，失效日期应该大于订货日期  
    const isAfterDate = (expiredate, orderdate) => expiredate > orderdate;  
    if(!isAfterDate(expiredate,orderdate)){  
        cb.utils.alert('失效日期必须大于订货日期', 'error');  
        return false;  
    }  
    // 判断子表物料编码是否重复  
    if (!checkList(viewModel.getAllData().preorderdetailList)) {  
        cb.utils.alert('该订单子表物料编码有重复，请检查!', 'error');  
        return false;  
    }  
})  
});
```

编辑后效果保存

名称关键字

前端函数

页面模型脚本

字符串脚本

字符串处理

字符串验证

伪随机码生成

其他常用脚本

调试工具

服务调用

页面加载js文件

调用后端api函数

其他

移动端脚本

音频管理

数据缓存

设备信息

图像接口

屏幕方向

网络请求

二维码接口

文件管理

照相/摄像

```
1 function (event) {
2   var viewModel = this;
3   let mode = viewModel.getParams().mode;
4   // 卡片新增状态使用
5   if (mode.toLocaleLowerCase() == 'add') {
6     var formatDate = function (date) {
7       var y = date.getFullYear();
8       var m = date.getMonth() + 1;
9       m = m < 10 ? '0' + m : m;
10      var d = date.getDate();
11      d = d < 10 ? ('0' + d) : d;
12      return y + '-' + m + '-' + d;
13    };
14
15    viewModel.on('afterLoadData', function(){
16      viewModel.get('orderstate').setValue("1");
17      viewModel.get('orderdate').setValue(formatDate(new Date()));
18      viewModel.get('expiredate').setValue(formatDate(new Date(new Date().getTime() + 86400000*3)));
19    });
20  }
21
22  // 定义函数：判断物料ID是否相同
23  const checkList = (params) => {
24    let temp = {};
25    for (let i in params) {
26      let key = params[i].materialId; //判断依据:物料ID
27      if (temp[key]) {
28        return false;
29      } else {
30        temp[key] = params[i];
31      }
32    }
33    return true;
34  }
35  // 保存前校验
36  viewModel.on('beforeSave', function(data) {
37    var expiredate = viewModel.get('expiredate').getValue();
38    var orderdate = viewModel.get('orderdate').getValue();
39    //判断dateA是否大于dateB。为true，失效日期应该大于订货日期
40    const isAfterDate = (expiredate, orderdate) => expiredate > orderdate;
41    if (!isAfterDate(expiredate, orderdate)){
42      cb.utils.alert('失效日期必须大于订货日期', 'error');
43      return false;
44    }
45    // 判断子表物料编码是否重复
46    if (!checkList(viewModel.getAllData().preorderdetailList)) {
47      cb.utils.alert('该订单子表物料编码有重复，请检查!', 'error');
48      return false;
49    }
50  })
}
```

重新打开页面或者刷新，录入非法数据后，点击保存应用效果如下：保存后提示

返回

按钮

销售信息

* 销售组织

02 创新销售中心

* 销售部门

sa02_销售二部

销售区域

销售渠道类型

直销

1 失效日期必须大于订货日期 X

预定信息

编码

SP2220105000001

* 订货日期

2022-01-05

* 失效日期

2022-01-01

业务员

运输方式

公路运输

* 币种

人民币

总数量

价税总计

预定状态

正常

备注

流程名称

流程版本

单据状态

开立志

客户信息

* 客户

龙城地产

开票客户

龙城地产

销售预订单明细

增行

删行

序号	物料编码	物料名称	单位	数量	含税单价	单品折扣	税率	税额	价税合计	
1	01000002	显示器	台	--	--	--	--	--	--	
2	01000002	显示器	台	--	--	--	--	--	--	
小计				0.00				0.0000	0.0000	

← 返回

按钮

销售信息

* 销售组织02 创新销售中心

* 销售部门sa02 销售二部

销售区域

销售渠道类型直销

该订单子表物料编码有重复, 请检查!

预定信息

编码SP2220105000001

* 订货日期2022-01-05

* 失效日期2022-01-08

业务员

运输方式公路运输

* 币种人民币

总数量

价税总计

预定状态正常

备注

流程名称

流程版本

单据状态开立志

客户信息

* 客户龙城地产

开票客户龙城地产

销售预订单明细

增行

删行

	序号	物料编码	物料名称	单位	数量	含税单价	单品折扣	税率	税额	价税合计	
<input type="checkbox"/>	1	01000002	显示器	台	--	--	--	--	--	--	
<input checked="" type="checkbox"/>	2	01000002	显示器	台	--	--	--	--	--	--	
小计					0.00			0.0000	0.0000		

4.1.3 金额计算

值改变后事件

1、 子表：

- 数量*单价*单品折扣*税率=税额；
- 数量*单价*单品折扣+税额=含税总价
- 主表含税总价=子表含税总价合计；
- 主表数量=子表数量合计

在刚才的函数里面，添加如下代码：

```
// 获取子表表格模型
let detailsModel = viewModel.get('preorderdetailList');
detailsModel.on('afterCellValueChange', function(data) {
    let rowIndex = data.rowIndex;
    //数量*单价*单品折扣*税率=税额; 数量*单价*单品折扣+税额=含税总价
    let num = detailsModel.getCellValue(rowIndex, 'num');
    let price = detailsModel.getCellValue(rowIndex, 'price');
    let discount = detailsModel.getCellValue(rowIndex, 'discount');
    let faxrate = detailsModel.getCellValue(rowIndex, 'faxrate');

    if(['num', 'price', 'discount', 'faxrate'].includes(data.cellName)) {
        let fax = num*price*(100-discount)*faxrate/10000;
        let money = num*price*(100-discount)/100 + fax;
        if(!isNaN(fax)) { detailsModel.setCellValue(rowIndex, 'fax', fax); }
        if(!isNaN(money)) { detailsModel.setCellValue(rowIndex, 'money', money); }
    }

    let detailsData = detailsModel.getData();
    // 主表合计价税总计
    if(['num', 'price', 'discount', 'faxrate'].includes(data.cellName)) {
        let total = 0;
        detailsData.forEach(item => {
            if(item.money){ total += item.money }
        });
        viewModel.get('total').setValue(total);
    }
    // 主表总数量
    if (data.cellName === 'num') {
        let totalnum = 0;
        detailsData.forEach(item => {
            if(item.num){totalnum += item.num}
        });
        viewModel.get('totalnum').setValue(totalnum);
    }
})
```

应用效果：录入对应数据后，税额、价税合计、总数量、价税总计会自动计算并带入。

4.1.4 查询区扩展

业务需求：要求仅能查询到自己创建的业务数据，使用前端函数扩展功能。
打开列表页面设计器，点击“脚本编辑”



编写前端函数



示例代码如下

```
const { userId, userName } = cb.utils.getUser();
// 查询前
viewModel.on('beforeSearch', function(args) {
    let search = args.params.condition.commonVOs;
    search.push({
        itemName: 'creator',
        value1: userId
    })

    args.params.condition.commonVOs = search
});
```

保存函数，页面设计器点击保存
效果如下图

The screenshot displays a web application interface for managing sales orders. At the top, there are tabs for '销售预订单-页面模板-单据设计' and '销售预订单aa'. The browser address bar shows a URL with parameters like 'domainKey=mdtrain&busiObj=salespreorder&tplid=1757257548384699416&designPreview=true&ss...'. Below the address bar, there are input fields for '编码', '销售组织', and '销售区域', along with a search button. A table of sales orders is visible, with columns for '序号', '销售信息', '失效日期', '订货日期', '业务员', '销售部门', '客户', '开票客户', '运输方式', and '币种'. The table contains four rows of data. Below the table, there are tabs for 'Elements', 'Console', 'Network', 'Sources', 'Memory', 'Performance', 'Application', 'Lighthouse', 'Recorder', 'Performance insights', and 'Overrides'. The 'Network' tab is selected, showing a list of requests. The first request is highlighted, and its 'Request Payload' is expanded, showing a JSON object with fields like 'page', 'billnum', 'busiObj', 'condition', 'commonVOs', 'filtersId', and 'solutionId'. The 'condition' field is expanded, showing a list of items with 'schemeName' and 'value1'.

序号	销售信息	失效日期	订货日期	业务员	销售部门	客户	开票客户	运输方式	币种
1	销售事业部 重庆	2023-09-30	2023-09-04	James	销售一部	红星中心	--	陆运	人民币
2	销售事业部 重庆	2023-07-30	2023-06-21	James	销售一部	红星中心	红星中心	陆运	人民币
3	销售事业部 重庆	2023-06-30	2023-06-13	James	销售一部	红星中心	--	陆运	人民币
4	销售事业部 重庆	2023-06-30	2023-06-21	James	销售一部	红星中心	红星中心	陆运	人民币

```
{  "page": {    "pageSize": 20,    "pageIndex": 1,    "totalCount": 1,    "billnum": "salespreorderList",    "bClick": true,    "bEmptyWithoutFilterTree": true,    "billnum": "salespreorderList",    "busiObj": "salespreorder"  },  "condition": {    "commonVOs": [      {        "schemeName": "默认方案",        "value1": "默认方案",        "isDefault": true      },      {        "schemeName": "默认方案",        "value1": "默认方案",        "isDefault": true      }    ]  },  "commonVOs": [    {      "schemeName": "默认方案",      "value1": "默认方案",      "isDefault": true    },    {      "schemeName": "默认方案",      "value1": "默认方案",      "isDefault": true    }  ],  "filtersId": "1757257548384699419",  "solutionId": "1757257548384699420"}
```

4.1.5 前端调试

在页面打开后打开 F12 下方 Console 找到对应位置，可以在对应的代码上做调试

← 返回 按钮

销售渠道类型 直销

预定信息

编码 000001 * 订贷日期 2021-12-29 * 失效日期 2021-12-31

业务员 joy 运输方式 公路运输 * 币种 人民币

总数量 7,000.0000 价税总计 20,404,950.0000 预定状态 正常

备注

流程名称 流程版本 单据状态 开立志

客户信息

* 客户 龙城地产 开票客户 龙城地产

销售预订单明细 增行 删行 ↕

序号	物料编码	物料名称	单位	数量	含税单价	单品折扣	税率	税额	价税合计
1	01000004	机床	台	2,000.00	10,000.0000	0%	2%	400,000.0000	20,400,000.0000
2	01000003	二级管	个	5,000.00	1,000	1%	0%	0.0000	4,950.0000
3	01000004	机床	台	--	--	--	--	--	--
小计				7,000.00				400,000.0000	20,404,950.0000

取消 保存并新增 保存

元素 控制台 来源 网络 性能 内存 应用 安全 Lighthouse PageSpeed Overrides EditThisCookie Adblock Plus Redux Components

网页 文件系统 替换 内容脚本 代码段

getHpExtCodeRe...source=BILL_BTN

```
46 }
47
48 // 定义函数: 判断物料ID是否相同
49 const checkList = (params) => {
50   let temp = {};
51   for (let i in params) {
52     let key = params[i].materialId; // 判断依据: 物料ID
53     if (temp[key]) {
54       return false;
55     } else {
56       temp[key] = params[i];
57     }
58   }
59   return true;
60 }
61 // 保存前校验
62 viewModel.on('beforeSave', function(data) {
63   var expiredate = viewModel.get('expiredate').getValue();
64   var orderdate = viewModel.get('orderdate').getValue();
65   // 判断dateA是否大于dateB, 为true, 失效日期应该大于订贷日期
66   const isAfterDate = (expiredate, orderdate) => expiredate > orderdate;
67   if (!isAfterDate(expiredate, orderdate)) {
68     cb.utils.alert('失效日期必须大于订贷日期', 'error');
69     return false;
70   }
71   // 判断子表物料编码是否重复
72   if (!checkList(viewModel.DgetAllData().preorderdetailList)) {
73     cb.utils.alert('该订单子表物料编码有重复, 请检查!', 'error');
74     return false;
75   }
76 }
77 // 在子表表格模型
78
```

第 51 行, 第 1 列 覆盖率: 不适用

监视

- 断点
 - getHpExtCodeRegister?domainKey=developplat...
 - for (let i in params) {
 - getHpExtCodeRegister?domainKey=developplat...
 - if (!checkList(viewModel.getAllData().pr...
- 作用域
 - 未暂停
- 调用堆栈
 - 未暂停
 - XHR/提取断点
 - DOM 断点
 - 全局监听器
 - 事件监听器断点
 - CSP 违规断点

录入数据后

销售渠道类型: 直销

预定信息

编码: 000001 * 订货日期: 2021-12-29 * 失效日期: 2021-12-31

业务员: joy 运输方式: 公路运输 * 币种: 人民币

总数量: 7,000.0000 价税总计: 20,404,950.0000 预定状态: 正常

备注:

流程名称: 流程版本: 单据状态: 开立志

客户信息

* 客户: 龙域地产 开票客户: 龙域地产

销售预订单明细

序号	物料编码	物料名称	单位	数量	含税单价	单品折扣	税率	税额	价税合计
1	01000004	机床	台	2,000.00	10,000.0000	0%	2%	400,000.0000	20,400,000.0000
2	01000003	二极管	个	5,000.00	1,000.0000	1%	0%	0.0000	4,950.0000
3	01000004	机床	台	---	---	---	---	---	---
小计				7,000.00				400,000.0000	20,404,950.0000

getHpaExtCodeRe_source=BILL_BTN

```

46 }
47
48 // 定义函数: 判断物料ID是否相同
49 const checkList = (params) => {
50   let temp = {};
51   for (let i in params) {
52     let key = params[i].materialId; // 判断依据: 物料ID
53     if (temp[key]) {
54       return false;
55     } else {
56       temp[key] = params[i];
57     }
58   }
59   return true;
60 }
61 // 保存前校验
62 viewModel.on('beforeSave', function(data) {
63   var expiredate = viewModel.get('expiredate').getValue();
64   var orderdate = viewModel.get('orderdate').getValue();
65   // 判断dateA是否大于dateB, 为true, 失效日期应该大于订货日期
66   const isAfterDate = (expiredate, orderdate) => expiredate > orderdate;
67   if (isAfterDate(expiredate, orderdate)) {
68     cb.utils.alert('失效日期必须大于订货日期', 'error');
69     return false;
70   }
71   // 判断子表物料编码是否重复
72   if (!checkList(viewModel.D.getAllData().preorderdetailList)) {
73     cb.utils.alert('该订单子表物料编码有重复, 请检查!', 'error');
74     return false;
75   }
76 }
77 // 获取子表表格模型
78

```

调试程序已暂停

监视

断点

getHpaExtCodeRegister?domainKey=developpl...

getHpaExtCodeRegister?domainKey=developpl... if (!checkList(viewModel.getAllData())...

作用域

代码块

i: undefined

本地

this: undefined

params: Array(3)

0: {materialname: '机床', num: 2000, di...

1: {unit_name: '个', unit: 257165952274...

2: {hasDefaultInit: true, _id: 'rowId_...

length: 3

[[Prototype]]: Array(0)

temp: {}

闭包

全局

调用堆栈

checkList getHpaExtCodeRe...ce=BILL_BTN:51

(匿名) getHpaExtCodeRe...ce=BILL_BTN:72

(匿名) cb.models.js:512

(匿名) cb.models.js:505

特殊说明: 页面设计器中调用的函数, 可以通过在函数中添加 **debugger**, 方便调试, 完成后如下图

```

1 function (event) {
2   var viewModel = this;
3   debugger
4   viewModel.get('taxno').setValue("");
5 }
6

```

此时可以断点到指定行中做调试操作。

4.2 后端函数

4.2.1 业务场景

业务需求：销售预订单保存时校验客户+订货日期销售订单唯一

4.2.2 后端函数说明

在后端函数中，核心知识点如下，如何获取传入的业务数据，如何调用持久层提供的 API。

1、 函数扩展点说明

扩展点名称	扩展点 code
保存前	beforeActionSave
保存后	afterActionSave
列表查询前	beforeListQuery
列表查询后	afterListQuery
删除前	beforeDelete
删除后	afterDelete
导入数据处理扩展点	handleImportData
提交前	beforeSubmit
提交后	afterSubmit
数据导入保存前扩展点	beforeExportQuery
数据导出后	afterExportData
查询详情前	beforeDetailQuery
查询详情后	afterDetailQuery
	beforeRefDataPermission
	SETTLE_AFTER
	afterPrintDataQuery
	beforeRefDataQuery

扩展点名称	扩展点 code
	afterRefDataQuery
	beforeTreeDataQuery
	afterTreeDataQuery
	initDefaultValue

2、 获取数据方法：

在后端函数中，从参数 `param` 中通过如下方法可获取当前业务数据

//在动作前可以获取当前业务数据

`let data=param.billDO;`//获取单条数据

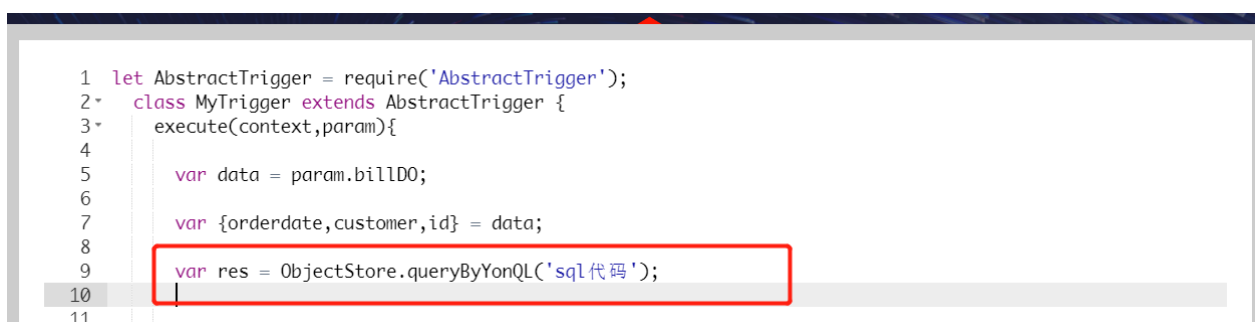
`let data=param.billDOs;`//获取多条数据

3、 持久层 API 获取

点击左侧函数，可以快速插入函数提供功能。比如插入条件查询 API：`ObjectStore.queryByYonQL`



插入后修改效果如下图



4、YonQL 查询模型

YonQL 查询都是针对元数据模型的查询，此时我们可以通过在右侧，选择需要查询的数据模型为当前节点的数据模型。

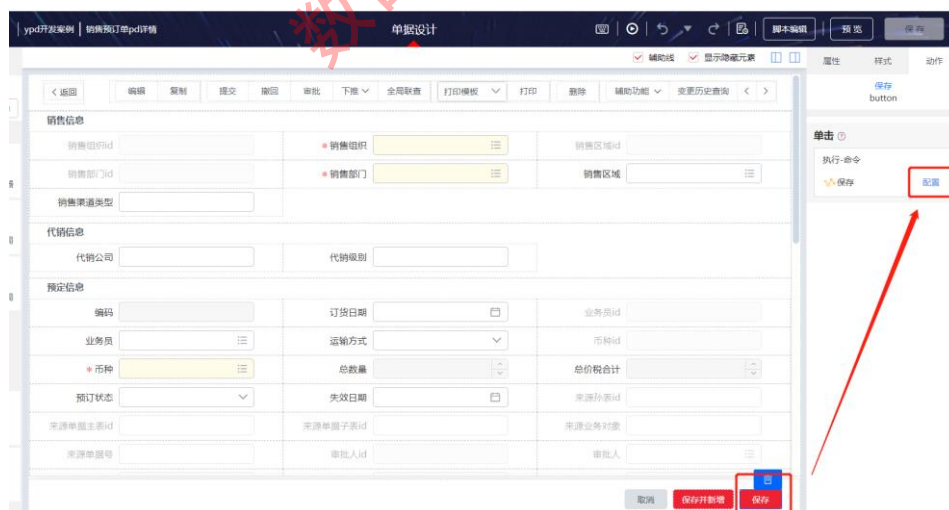
此时，如比如当前需要查询当前的模型，在右侧“模型”处选择模型，选择完成后，点击 uri 可以获取 sql 中所需的当前模型 uri。如下图所示



4.2.3 后端函数开发

4.2.3.1 新建后端函数

找到保存按钮，右侧点击“配置”



点击新增函数



编写后端函数，如下图



代码示例

```
let AbstractTrigger = require('AbstractTrigger');
class MyTrigger extends AbstractTrigger {
  execute(context,param){

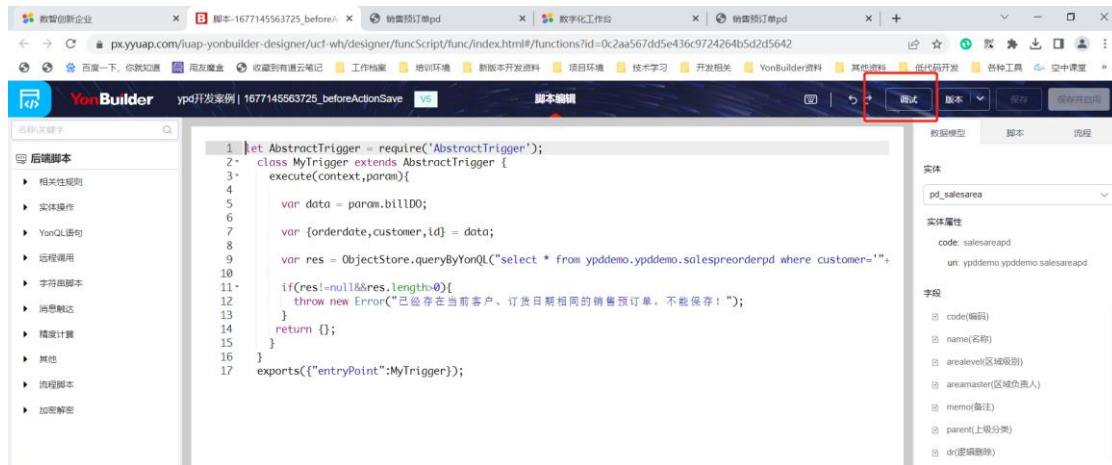
    var data = param.billDO;
    var {orderdate,customer,id} = data;

    var res = ObjectStore.queryByYonQL("select * from ypddemo.ypddemo.salespreorderpd where customer='"+customer+"' and orderdate='"+orderdate+"' and id!='"+id+"'");

    if(res!=null&&res.length>0){
      throw new Error("已经存在当前客户、订货日期相同的销售预订单，不能保存！");
    }
    return {};
  }
}
exports({"entryPoint":MyTrigger});
```


4.2.3.2 函数调试

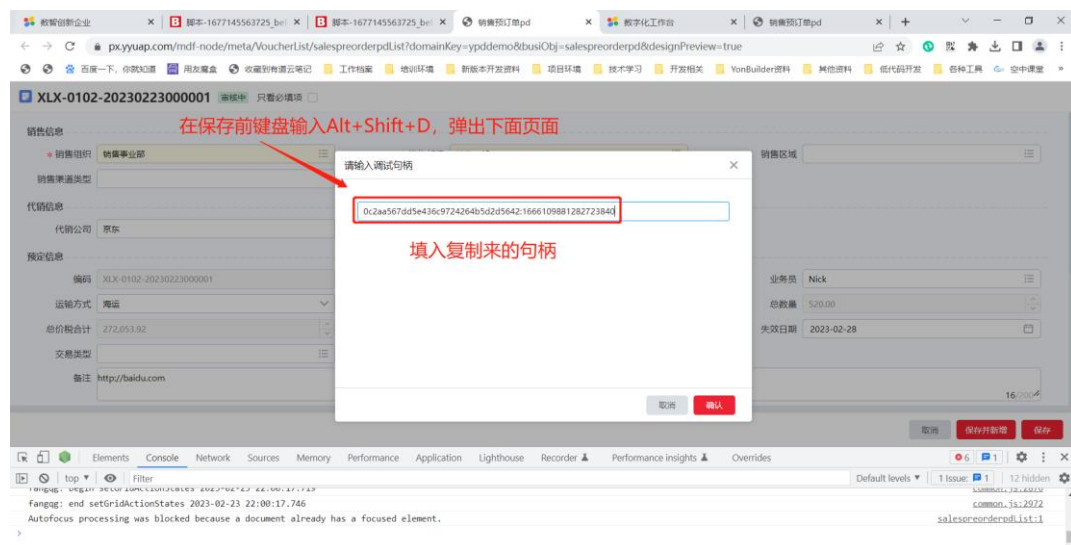
函数右上方“调试”按钮



复制调试句柄，下面为调试跳转后页面



打开页面后，在真正的函数调用前，输入按 SHIFT+ALT+D，弹出页面将上面复制的调试句柄填充进去，点击确定



点击保存

销售信息

销售组织 销售事业部 销售部门 销售一部 销售区域

销售渠道类型

代销信息

代销公司 京东 代销级别 一级

预定信息

编码 XLX-0102-20230223000001 订单日期 2023-02-09 业务员 Nick

运输方式 海运 币种 人民币 总数量 520.00

总价合计 272,053.92 预订状态 正常 失效日期 2023-02-28

交易类型 单据状态 审核中

备注 http://baidu.com

取消 保存并继续 保存

跳转到调试器



左侧点击行上，可以配置调试点；点击右侧上方调试按钮，可跳转到断点或者单步执行都可以



4.2.4 应用效果

px.yyuap.com/mdf-node/meta/VoucherList/salespreorderpdList?domainKey=ypddemo&busiObj=salespreorderpd&designPreview=true

百度一下, 你就知道 用友魔盒 收藏到有道云笔记 工作档案 培训环境 新版本开发资料 项目环境 技术学习 开发相关 YonBuilder资料 其他资料 低代码开发 各种工具 空中课堂

XLX-0102-20230223000001 审核中 只看必填项

销售信息

销售组织

销售事业部

销售渠道类型

代销信息

代销公司

京东

代销级别

一级

预定信息

编码

XLX-0102-20230223000001

运输方式

海运

总价税合计

272,053.92

交易类型

备注

http://baidu.com

订货日期

2023-02-23

币种

人民币

预订状态

正常

单据状态

审核中

业务员

Nick

总数量

520.00

失效日期

2023-02-28

销售区域

com.yonyou.iuap.ucf.common.exception.BusinessException: 已经存在当前客户、订货日期相同的销售预订单, 不能保存!

取消

保存并新增

保存

数智平台事业部

5 典型业务功能

业务需求

预订单“整单关闭”功能

- 1、 添加“辅助功能”父按钮，“整单关闭”子按钮，选中数据成功后可将预订状态变更为“已关闭”，同时更新子表“是否行关闭”为“是”；
- 2、 只有预订状态为“正常”状态的按钮可进行“整单关闭”操作；

预订单子表“行关闭”功能

- 1、 预订单子表添加“行关闭”按钮，可多选中子表行点击“行关闭”，提示信息“是否确认进行行关闭操作”，点击是，则刚更新子表“是否行关闭”为“是”；
- 2、 只有“是否行关闭”为“是”的单据才进行“行关闭”，判断是否存在需要进行行关闭的数据，同时返回提示信息。

5.1 按钮功能开发

以关闭功能为例，学习如何在线上通过前后端函数定义完成按钮功能。

5.1.1 业务需求

业务需求描述：

预订单“整单关闭”功能

- 1、 添加“辅助功能”父按钮，“整单关闭”子按钮，选中数据成功后可将预订状态变更为“已关闭”，同时更新子表“是否行关闭”为“是”；
- 2、 只有预订状态为“正常”状态的按钮可进行“整单关闭”操作；
- 3、 整单关闭后，子表状态业务“行关闭”状态

预订单子表“行关闭”功能

- 1、 预订单子表添加“行关闭”按钮，可多选中子表行点击“行关闭”，提示信息“是否确认进行行关闭操作”，点击是，则刚更新子表“是否行关闭”为“是”；
- 2、 只有“是否行关闭”为“是”的单据才进行“行关闭”，判断是否存在需要进行行关闭的数据，同时返回提示信息。

5.1.2 主表按钮开发

5.1.2.1 界面添加按钮

拖拽下拉按钮到按钮栏



调整父按钮名称



添加子按钮



修改子按钮名称为“整单关闭”



5.1.2.2 前端函数开发

在页面初始化函数里面添加如下代码：（注意自定义按钮编码替换）

```
// 整单关闭按钮
viewModel.get('button36bj').on('click', function() {
  let data = viewModel.getAllData();
  let orderstate = viewModel.get('orderstate').getValue();
  if (orderstate != '1') {
    cb.utils.alert('只有预订状态为“正常”,才可以整单关闭。', 'error');
  } else {
    cb.rest.invokeFunction("4b7134e393a8463ebf8f3939dedccb4e", {data: data}, function(err, res) {
      if (res.data === 'success') {
        cb.utils.alert('整单关闭操作成功!', 'success');
        viewModel.execute('refresh');
      }
    })
  }
})
```

5.1.2.1 API 函数开发

需求描述：主表预订状态更新为“已关闭”，子表所有行关闭字段更新为“是”；
在“函数”页签，“后端”对应分组上，点击新增“API”类型脚本，录入函数名称，点击确定

新增脚本

* 脚本名称

PreSaleCloseBill

* 类型

api脚本

* 目录

业务函数

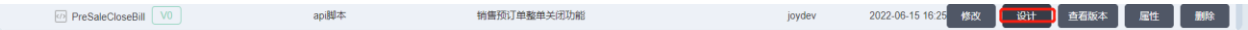
备注

销售预订单整单关闭功能

取消

确定

完成后效果如下图，点击“设计”



在函数页面设计器中，编写 API 函数

名称关键字

后端函数

实体操作

YonQL语句

远程调用

字符串脚本

消息触达

精度计算

其他

流程脚本

加密解密

```
1 let AbstractAPIHandler = require('AbstractAPIHandler');
2 class MyAPIHandler extends AbstractAPIHandler {
3   execute(request){
4     //更新主表
5     var data = request.data;
6     var object = {id:data.id, orderstate:"3"};
7     var res = ObjectStore.updateById("bipdev1221.bipdev1221.salespreorder",object,"dc3a297f");
8
9     var updateWrapper = new Wrapper();
10    updateWrapper.eq("salespreorder_id",data.id);
11    // 待更新字段内容
12    var toUpdate = {lineclose:"Y"};
13    var res = ObjectStore.update("bipdev1221.bipdev1221.preorderdetail",toUpdate,updateWrapper,"dc3a297f");
14    return {data: 'success'};
15  }
16 }
17 exports({"entryPoint":MyAPIHandler});
18
```

数据模型

函数

流程

模型

销售区域定义

模型属性

code: salesarea

uri: bipdev1221.bipdev1221.salesarea

字段

A code(编码)

A name(名称)

salesarea(区域级别)

A memo(备注)

具体代码内容如下：

```
let AbstractAPIHandler = require('AbstractAPIHandler');
class MyAPIHandler extends AbstractAPIHandler {
  execute(request){
    //更新主表
    var data = request.data;
    //子表待更新数据
    var detailData = data.preorderdetailList;
    var detailObject = detailData.map(item => {
      return {
        id: item.id,
        lineclose:'Y',
        _status:"Update"
      }
    });
    var object = {id:data.id, orderstate:"3", "preorderdetailList":detailObject};
    var res = ObjectStore.updateById("bipdev1221.bipdev1221.preorderdetail",object, "salespreorder");

    return {data: 'success'};
  }
}
exports({"entryPoint":MyAPIHandler});
```

5.1.3 子表行按钮开发

5.1.3.1 前端添加按钮

在子表按钮栏区域添加 “行关闭” 按钮，选择按钮合适区域，点击工具栏，右键添加按钮



添加完成后可修改按钮标题，效果如下图



5.1.3.2 配置按钮状态



5.1.3.3 前端函数开发

1、在原有的页面初始化函数中，增加行关闭按钮点击事件。点击“脚本编辑”，弹出前端函数界面。




```
var viewModel = this;
let alertPromise = new cb.promise();
// 获取子表表格模型
let detailsModel = viewModel.get('preorderdetailList');
cb.utils.confirm('确认进行行关闭操作?', function() {
    let selectData = detailsModel.getSelectedRows();
    if (selectData.length === 0) {
        cb.utils.alert('请先选择要操作的子表数据。', 'warning');
        return false;
    } else if (selectData.some(item => item.lineclose === 'Y')) {
        cb.utils.alert('请选择 是否行关闭 为否的数据。', 'warning');
        return false;
    }
    cb.rest.invokeFunction('987ab121f25d4eae872cb022bd223d56', {data: selectData}, function(err, res) {
        if (res.data === 'success') {
            cb.utils.alert('行关闭操作成功!', 'success');
            viewModel.execute('refresh');
        }
    })
    return alertPromise.resolve();
}, function() {
    return alertPromise.reject();
})
})
```

完成后，保存函数

5.1.3.4 API 函数定义

新建 API 函数实现“行关闭”功能，需要进行子表数据更新操作。

需求：在当前行数据上，将“是否行关闭”更新为“是”

实现方式：

1、API 函数可使用 `ObjectStore.updateBatch` 函数方法进行批量更新操作。

2、先更新后重新查询最新数据返回给前端刷新 Model；

注意要替换模型 uri，应该配置为当前预订单的模型 uri，配置完成后效果如下图



The screenshot displays the API function definition interface. The code editor shows the following JavaScript code:

```
1 let AbstractAPIHandler = require('AbstractAPIHandler');
2 class MyAPIHandler extends AbstractAPIHandler {
3   execute(request){
4     var dataArray = request.data;
5     var object = dataArray.map(item => {
6       id: item.id,
7       lineclose: 'Y'
8     });
9   }
10 }
11
12 var res = ObjectStore.updateBatch("bipdev1221.bipdev1221.preorderdetail", object, "dc3a297f");
13
14 return {data: "success"};
15 }
16 }
17 exports({ "entryPoint": MyAPIHandler });
```

On the right, the configuration panel shows the data model configuration. The model is named "销售区域定义" (Sales Area Definition). The URI is set to "bipdev1221.bipdev1221.salesarea". The configuration panel also lists the fields: code, name, salesarea, and memo.

源码示例如下：

```
let AbstractAPIHandler = require('AbstractAPIHandler');
class MyAPIHandler extends AbstractAPIHandler {
  execute(request){
    var dataArray = request.data;
    var object = dataArray.map(item => {
      return {
        id: item.id,
        lineclose:'Y'
      }
    });

    var res = ObjectStore.updateBatch("bipdev1221.bipdev1221.preorderdetail",object, "salespreorder");

    return {data:"success"};
  }
}
exports({"entryPoint":MyAPIHandler});
```

5.1.4 应用效果

未勾选子表数据，点击 行关闭 按钮页面展示如图：

销售信息

销售组织 库管中心

销售渠道类型 直销

销售区域 东部区域

销售部门 创新销售中心

销售区域 东部区域

预定信息

编码 000001

业务员 joy

总数量 7,000.0000

备注 --

流程名称 --

客户信息

客户 龙城地产

开票客户 龙城地产

订货日期 2021-12-29

运输方式 公路运输

价税总计 510,000.0000.0000

流程版本 --

失效日期 2021-12-31

币种 人民币

预定状态 正常

单据状态 开立态

销售预订单明细

行关闭

序号	物料编码	物料名称	单位	数量	含税单价	单品折扣	税率	税额	价税合计	出
1	01000004	机床	台	2,000.00	10,000.0000	0%	2%	400,000.0000	20,400,000.0000	
2	01000003	二极管	个	5,000.00	--	--	--	--	--	
小计				7,000.00				400,000.0000	20,400,000.0000	

创建时间 2021-12-30 10:47:21

修改时间 2021-12-31 16:33:10

创建人 joydev

修改人 joydev

点击行关闭，更新状态为“是”并自动刷新页面：

销售信息

销售组织 库管中心

销售渠道类型 直销

预定信息

编码 000001

业务员 joy

总数量 7,000.0000

备注 --

流程名称 --

客户信息

客户 龙城地产

销售预订单明细

销售部门 创新销售中心

行关闭操作成功!

订货日期 2021-12-29

运输方式 公路运输

价税总计 510,000,000.0000

流程版本 --

开票客户 龙城地产

加载中...

销售区域 东部区域

失效日期 2021-12-31

币种 人民币

预定状态 正常

单据状态 开立态

✓	序号	物料编码	物料名称	单位	数量	含税单价	单品折扣	税率	税额	价税合计	⌵⌴
✓	1	01000004	机床	台	2,000.00	10,000.0000	0%	2%	400,000.0000	20,400,000.0000	
✓	2	01000003	二极管	个	5,000.00	--	--	--	--	--	
小计					7,000.00				400,000.0000	20,400,000.0000	

创建时间 2021-12-30 10:47:21

修改时间 2021-12-31 16:33:10

创建人 joydev

修改人 joydev

数智平台事业部

6 附录

6.1 Model 模型基本概念

后续开发过程中使用所有的方法的前提，是先学会怎么找对模型

1、通常对某个模型进行操作时，需要先找到这个模型，然后调用模型上支持的方法和事件；

2、开发环境，会在开发者调试工具里开放一个把柄就是 **yya**，也就是页面最外层的 **viewModel**，通过 **yya** 这个把柄获取页面内的相关模型，然后可以使用相关的方法和事件。

具体以获取查询区模型为例：

```
> yya.getCache('FilterViewModel')
< ▼ 0 {__get_data: f, __set_data: f, __del_data: f, __cls_data: f, __data: f, ...} ⓘ
  ▶ __data: {listeners: Array(1), propertyNames: Array(17), events: {...}, cache: {...}, params: {...}, ...}
  ▶ __cls_data: f ()
  ▶ __data: f ()
  ▶ __del_data: f (p)
  ▶ __get_data: f (p)
  ▶ __set_data: f (p,l,f)
  ▼ __proto__: C
    ▶ init: f (C)
    ▶ initData: f ()
    ▶ modelType: "FilterViewModel"
    ▶ __proto__: C
```

视图业务模型我们称之为 **viewModel**，它是对 **model** 数据模型的组合和组装，实现对数据的联动和修改。

- **viewModel** 原生数据模型（最外层模型）
- **BaseModel** 数据模型基类
- **GridModel** 表格模型（对 **BaseModel** 的封装）
- **TreeModel** 树形模型（对 **BaseModel** 的封装）
- **FilterModel** 过滤器模型（对 **BaseModel** 的封装）
- **ReferModel** 参照模型（对 **BaseModel** 的封装）
- **ListModel** 列表模型（对 **BaseModel** 的封装）

各个页面样式都由各个 **Model** 模型组装

默认方案

ID

名字

文本内容

filterModel

新增

下推

导入

导出

删除

打印

下载

ID	名字	文本内容	回顾日期	回顾人	编码	流程名称	流程版本
0055001	王城	描述性内容	2021-04-26	王城	要去		
2277271571943680	吴松	描述内容	2021-05-12	吴松	说是	gridModel	

viewModel

请输入关键字

项目1

treeModel

项目档案GKG

编码

项目1

上级分类

名称

项目1

物料计量单位

千克

设计工程量

1.00000000

工程特征

序号	特征值名称	特征值	物料计量单位
1	工程特征1	1	千克
2	工程特征11	11	克
3	工程特征111	66	克

viewModel



用友BIP | iuap平台

数智平台事业部