单据服务

# 元数据定义

## BillField

对应单据模板中的field，也即MetaModel中field

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必须 | 含义 |
| code | String | Y | 属性编码 |
| name | String | Y | 属性名称 |
| type | String | Y | 属性类型 |
| showtype | String |  | 显示位置，适用于主表，值域为head,tail单据的头部或者尾部显示  前端没有支持 |
| showlist | boolean |  | 是否在单据的列表状态显示 |
| showcard | boolean |  | 是否在单据的卡片状态显示 |
| cardpos | int |  | 卡片状态该属性的显示位置 |
| listpos | int |  | 列表状态该属性的显示位置 |
| ispk | boolean |  | 是否为主键，当type=T\_PK时，须设置ispk为true |
| rangeset | String |  | 值域，适用于T\_CODE,T\_STAT两种类型,格式为：v1:lab1@v2:lab2:default@v3:lab3 |
| required | boolean |  | 是否必须，如为true，则文字前面有个\*标志，没有填值则无法提交  前端部分支持 |
| readOnly | boolean |  | 是否只读，卡片状态时不可编辑 |
| config | GenericVO |  | 值域配置，type=T\_REF时，须设置该值，可选属性包括：  {  DS\_CODE：数据集编码  param：对话框参数  {  show\_type：表格显示方式，tree树形显示；grid：表格显示  }  key：  } |
| isCate | boolean |  | 是否分类 |
| 以下字段BillField没有，配置只在前端起效 | | | |
| tip | String |  | 控件下方的提示 |
| editable | Boolean |  | 是否可编辑  其为true时，在列表状态不显示 |
|  |  |  |  |

### Q&A

1. **什么是卡片状态，列表状态？**

列表状态：以列表方式显示数据，如下图：



卡片状态：以卡片方式显示数据，如下图



1. **什么是listpos的显示位置？**

如“列表状态”图，商户名称为0，商户编号为1，电话号码为2，依次推类……

showlist为true时，listpos生效。实际上，前端按照单据模板showlist为true的顺序依次显示，与cardpos无关。

1. **什么是cardpos的显示位置？**

如“卡片状态”图，卡片为2列，商户名称的cardpos为0，商户编号的cardpos为1，电话号码的cardpos为2，依次推类……

showcard为true时，cardpos生效。

1. **什么是show\_type中的grid和tree显示方式？**

grid方式如下图：



tree方式如下图：



1. **rangset和config怎么用？**

type=T\_STAT时，设置rangset

type=T\_REF时，设置config

如果值的可选项比较多时，应采用T\_REF，设config的DS\_CODE为枚举数据集。

1. **isCate是什么意思？**

对于单据模板，可能需要不同显示方式。以报销单为例，不同报销类型的报销单内容是不同的，但流程基本一致。报销类型的isCate为true，对比普通报销与差旅报销的明细，所需字段是不一样的。因此普通报销的单据模板的获取路径为

/action/oa/pf/bs/model/gl\_ro\_bill/RANGXXXX/oa.pf.bs.model.find.action

普通报销明细：





差旅报销明细：





## RelationshipModel

描述主子表关系，slave中的rlt部分

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 含义 |
| bridge | **String** | 桥接表编码，是否通过rlt表进行桥接，适用于多对多的关系 |
| mainField | **String** | 主表的主键 |
| slaveField | **String** | 子表的外键 |
| rlt\_type | **int** | 连接类型，0：直连；1：桥接 |
| rlt\_method | **String** | 连接方式，取值有force,weak两种，force为默认值，将在删除主表记录时，直接删除子表相关记录，weak则不受影响，只删除主表数据，以及关联关系表记录 |

### Q&A

1. bridge如何实现？

???

目前主要用force连接方式

## MetaModel

单据模板中的main和slave部分

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 含义 |
| metaCode | **String** | **Y** | 实体编码，数据库表码 |
| metaName | **String** | **Y** | 实体名称，数据库表名 |
| isFileset | **Boolean** |  | 是否为附件 |
| layout | **String** | **Y** | 布局，可选值为:form;grid |
| fetchAction | **String** |  | 只适用于slave的元数据，用于自定义查询子集 |
| fields | **List<BillField>** | **Y** | 属性列表 |
| rlt | **RelationshipModel** |  | 只适用于slave meta，用于描述slave与main关系 |

### Q&A

1. **isFileset如何用？附件如何设置？**

例如：

{"metaCode":"RLT\_BIZ\_ATT","metaName":"附件","isfileset":true}

RltBizAttMeta定义了附件表

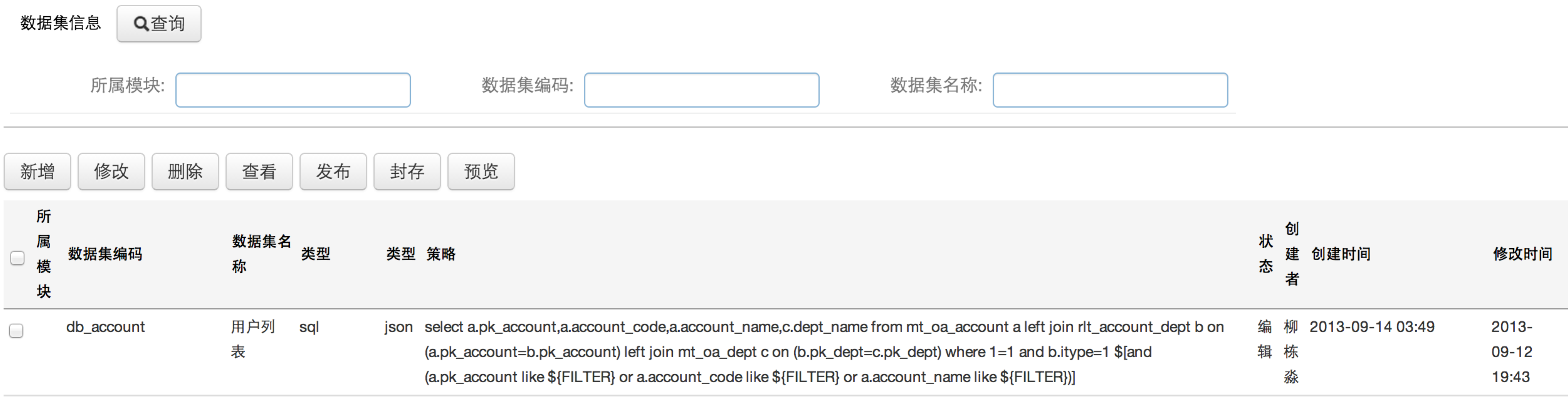
1. **layout的form和grid有何不同？**

貌似目前没有区别

grid



form



1. **fetchAction有什么用？**

子集不采用默认查询url，就在此设置查询请求的url。

## BillModel

单据模板模型

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必填 | 含义 |
| bill\_code | **String** | **Y** | 单据编码，字符下划线，系统内唯一，按照模块编码+业务对象编码的方式命名 |
| layout | **String** | **Y** | 子集的界面布局方式，list就是平铺，tab是多页签  目前只支持list |
| ext | **String** |  | 单据的数据操作扩展类，用于嵌入各种操作前后的校验和预处理 |
| main | **MetaModel** | **Y** | 单据的主表对象，是一个meta model |
| slave | **Map<String,MetaModel>** |  | 单据的子集列表，是一个meta model list |
| cateField | **BillField** |  | 只在class中存在，等于MetaModel中iscate=true的BillField |

### Q&A

1. **BillModel中的layout VS MetaModel中的layout？**

BillModel的layout是设置整体框架的布局方式，以列表还是多页显示；MetaModel的layout是设置表格数据的显示方式。

1. **ext什么意思？**

ext的值对应一个Service，该Service继承自ExtBillBizService，通过实现ExtBillBizService的接口，在单据操作前后插入一些操作，如在保存前校验数据有效性等。

## AttachmentMeta

附件元数据

# BillController

提供请求响应

## find

获取单据模板

请求url：/oa/pf/bs/model/{bill\_code}/oa.pf.bs.model.find.action

bill\_code：单据编码List，只用第一个

pk\_cate：分类编码，可为null，如果不为null，则文件路径为

/oa/pf/bs/model/{bill\_code}/{pk\_cate}/oa.pf.bs.model.find.action

返回：BillModel的GenericVO结构

public GenericVO find(@RequestParam("bill\_code") List<String> billCodes, @RequestParam(value = "pk\_cate", required = false)

## save

保存单据数据

请求url：/oa/pf/bs/data/oa.pf.bs.data.save.action

bill\_code：单据编码

data：以键值对包含主子表的内容

client：用户信息

返回：新生成的主键pk

public String save(@RequestParam("bill\_code") String bill\_code, @RequestParam("data") GenericVO data, ClientEnv client)

## fetch

searchURL，查询单据数据

请求url：/oa/pf/bs/data/oa.pf.bs.data.fetch.action

bill\_code：单据编码

param：查询条件，键值对

client：用户信息

public List<GenericVO> fetch(@RequestParam("bill\_code") String bill\_code, @RequestParam("param") GenericVO param, ClientEnv client)

## deleteSlaveData

删除子表数据

请求url：/oa/pf/bs/data/oa.pf.bs.data.deleteSlave.action

bill\_code：单据编码

pk\_cate：分类

slave\_meta\_code：子表编码

pk\_value\_main：主键值

pk\_list：

client：用户信息

public String deleteSlaveData(

@RequestParam("bill\_code") String bill\_code,

@RequestParam("pk\_cate") String pk\_cate,

@RequestParam("slave\_meta\_code") String slave\_meta\_code,

@RequestParam("pk\_value\_main") String pk\_value\_main,

@RequestParam("pk\_list") List<String> pk\_list, ClientEnv env)

## saveSlaveData

保存子表数据

请求url：/oa/pf/bs/data/oa.pf.bs.data. saveSlave.action

bill\_code：单据编码

pk\_cate：分类

data：子表数据

slave\_meta\_code：子表的MetaCode

client：用户信息

public String saveSlaveData(@RequestParam("bill\_code") String bill\_code,

@RequestParam(value = "pk\_cate", required = false) String pk\_cate,

@RequestParam("data") GenericVO slaveData,

@RequestParam("slave\_meta\_code") String slaveMetaCode, ClientEnv env)

## fetchCache

获取mongodb中保存的BillModel

bill\_code：单据编码

public List<GenericVO> fetchCache(@RequestParam("bill\_code") String bill\_code)

## cleanCache

清除mongodb中保存的BillModel

bill\_code：单据编码

public String cleanCache(@RequestParam("bill\_code") List<String> bill\_code\_list)

## cleanAllCache

清除mongodb中保存的mt\_oa\_bs\_model，mt\_oa\_dataset，mt\_oa\_pod集合

返回全部集合名称

public GenericVO cleanAllCache()

## findBillData

单击修改按钮时，根据主键值返回以BillModel格式封装的GenericVO键值对

请求url：/oa/pf/bs/data/{bill\_code}/oa.pf.bs.data.find.action

bill\_code：单据编码

pk\_cate：分类

pk\_value\_main：主键值

client：用户信息

public GenericVO findBillData(@RequestParam("bill\_code") String bill\_code,

@RequestParam(value = "pk\_cate", required = false) String pk\_cate,

@RequestParam("pk\_value\_main") String pk\_value\_main, ClientEnv client)

# BillBizService

/\*\*

\* 根据billcode去获取bill对应的结构(billmodel)

\*首先从MonogoDB中获取billModel，如果没有，则去该路径下找action/oa/pf/bs/model/${bill\_code}/${pk\_cate}/oa.pf.bs.model.find.action，然后保存到MongoDB中

\* @param billCode

\* 单据编码

\* @param catePK

\* 动态单据中,主表所属分类

\* @throws BuzException

\*/

public GenericVO find(String billCode, String catePK) throws BuzException;

/\*\*

\* 从MongoDB中获取bill model，仅供内部delete使用

\*

\* @param billCode

\* @param pk\_cate

\* @param client：没用

\* @return

\* @throws BuzException

\*/

public GenericVO findByVO(String billCode, String pk\_cate, ClientEnv client) throws BuzException;

/\*\*

\* 保存,修改bill中添加的数据

\*

\* @param model

\* @param billdata

\* @param env

\* TODO

\* @throws BuzException

\*/

public String saveBillData(BillModel model, GenericVO billdata, ClientEnv env) throws BuzException;