## 数据生产计划管理

系统需要实现通过数据生产计划，进行数据生产工作管理。

### 创建数据生产计划

数据生产计划可以分为年计划、月度计划，周计划 、日计划。

界面维护如图：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 计划管理 | 添加工单 | |  |  |  |
| 信息运营中心--数据生产计划 | | | | | |
| 计划名称 | XXXX年XX月数据生产计划 | | | | |
| 类型 | 月度 |  |  |  |  |
| 周期 | 2016年8月1日 | 到 | 2016年8月31日 |  |  |
| 制定人员 | XXX |  |  |  |  |
| 制定时间 | 2016年7月25日 |  |  |  |  |
| 月度执行目标 | | | | | |
| 净增目标 | | | | | |
| 目标类型 | 计划值 | 到达值 | 任务派发值 | 实际处理值 | 在途值 |
| 净增商家数据： | 50,000 | 30,000 | 40,000 | 0 | 0 |
| 净增深度信息： | 5,000 | 4,000 | 5,000 | 0 | 0 |
| 净增POI信息： | 15,000 | 5,000 | 10,000 | 0 | 0 |
| 维护目标 | | | | | |
| 目标类型 | 计划值 | | 任务派发值 | 实际处理值 | 在途值 |
| 核实商家数据： | 50,000 | | 40,000 | 0 | 0 |
| 核实深度信息： | 5,000 | | 5,000 | 0 | 0 |
| 核实POI信息： | 15,000 | | 10,000 | 0 | 0 |
| 备注： |  |  |  |  |  |
|  |  |  |  |  |  |
| 新增/删除/保存 | | | | | |

数据表结构：

1. 数据库表名为：T\_DATA\_PLAN (中文注释为数据生产计划比表)
2. 数据库字段为：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **字段类型** | **长度** | **描述** |
| **DPLID** | STRING | 40 | T\_DATA\_PLAN唯一编号 |
| **DPLNAME** | STRING | 40 | 计划名称 |
| **DPLTYPE** | STRING | 40 | 计划类型 |
| **DPLSTARTDT** | DATE |  | 计划开始时间 |
| **DPLENDDT** | DATE |  | 计划结束时间 |
| **DPLPLANNER** | STRING |  | 计划制定人员 |
| **DPLPLANTIME** | DATE |  | 计划制定时间 |
| **INCPLCUST** | INTEGER |  | 商家净增计划量 |
| **INCPFCUST** | INTEGER |  | 商家净增计划到达量，计划完成量 |
| **INCPLCUSTINFO** | INTEGER |  | 深度净增计划量 |
| **INCPFCUSTINFO** | INTEGER |  | 深度净增计划到达量 |
| **INCPLCUSTPOI** | INTEGER |  | 净增POI计划量 |
| **INCPFCUSTPOI** | INTEGER |  | 净增POI计划到达量 |
| **FIXPLCUST** | INTEGER |  | 维护商家计划量 |
| **FIXPFCUSTINFO** | INTEGER |  | 维护深度计划量 |
| **FIXPLCUSTPOI** | INTEGER |  | 维护POI计划量 |
| **CREATEDT** | DATE |  | 创建时间 |
| **FIXDT** | DATE |  | 修改订时间 |
| **FIX** |  |  |  |
| **DPLPLANNER** | STRING |  | 计划修订人 |

1. 计算字段为：

任务派发值：通过计算工单实际下发数据量计算获得。

实际处理值：通过计算工单实际下发数据量，采编完成实际工作统计获得。处理值表示数据生产所有流程都已经完成。

在途值：通过计算工单实际下发数据量，采编完成实际工作统计获得。在途表示数据生产仍然在流程中，没有最终完成。

### 计划工单维护

将已经制作工单添加计划中。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 计划工单维护 | | | | | | | | | | | |
| 待装入计划工单 | | | | | | | | | | | |
| 序号 | 工单编号 | 工单名称 | 自定义 工单名称 | 所属机构 | 流程类型 | 工单状态 | 创建人 | 待处理数据量 | 新增数据量 | 修改数据量 | 操作 |
| 1 | ‘1040000000381 | 高危行业维护 |  | 广州区域中心 | 基础采集工单 | 新建 | XXX | 10000 | 5000 | 5000 | 添加 |
| 已经添加计划工单 | | | | | | | | | | | |
| 序号 | 工单编号 | 工单名称 | 自定义 工单名称 | 所属机构 | 流程类型 | 工单状态 | 创建人 | 待处理数据量 | 新增数据量 | 修改数据量 | 操作 |
| 1 | ‘1040000000381 | 高危行业维护 |  | 广州区域中心 | 基础采集工单 | 新建 | XXX | 10000 | 5000 | 5000 | 添加/删除 |

#### 工单处理相关表

工单表：

|  |  |  |
| --- | --- | --- |
| **表名称** | **字段** | **描述** |
| **T\_ORDER** | ORDER\_ID | 工单ID |
| **T\_ORDER** | NAME | 工单名 |
| **T\_ORDER** | ALIASNAME | 自定义工单名 |
| **T\_ORDER** | DEPTID | 所属机构 |
| **T\_ORDER** | TYPE\_ID | 流程类型 |
| **T\_ORDER** | STATUS | 工单状态 |
| **T\_ORDER** | DEAD\_LINE | 处理期限 |
| **T\_ORDER** | INFO\_TYPE | 信息类型 |
| **T\_ORDER** | GENERATE\_TIME | 生成时间 |
| **T\_ORDER** | CREATOR\_ID | 创建人 |
| **T\_ORDER** | REMARK | 备注 |
| **T\_ORDER** | ENABLE\_CALLOUT | 启用外呼 |
| **T\_ORDER** | PUB\_COUNT | 发布数量 |
| **T\_ORDER** | SYS\_NAME | 业务系统 |
| **T\_ORDER** | MUST\_CALLOUT | 必须外呼 |

所属机构：

|  |  |  |
| --- | --- | --- |
| **表名称** | **BAS\_DEPARTMENT** | |
| **字段名称** | 类型 | 描述 |
| **DEPTID** | STRING | 机构ID |
| **PARENTID** | STRING | 上级ID |
| **DEPTCODE** | STRING | 机构编码 |
| **DEPTNAME** | STRING | 机构名称 |
| **PARTITION\_CODE** | STRING | 部门默认地市代码 |
| **TYPE\_ID** | STRING | 层级 |
| **DEFAULT\_CITY\_NO** | STRING | 默认地市区号 |

所属机构数据：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **所属机构编号** | **所属机构上级编号** | **编码** | **所属机构名称** | **上级编码** | **层级** | **地市区号** |
| **1** | 0 | GDHB | 广东省号百运营中心 | GZ | 1 | 020 |
| **10** | 4 | FS | 佛山分公司 | FS | 3 | 0757 |
| **11** | 4 | HY | 河源分公司 | HY | 3 | 0762 |
| **12** | 4 | HZ | 惠州分公司 | HZ | 3 | 0752 |
| **13** | 4 | JM | 江门分公司 | JM | 3 | 0750 |
| **14** | 4 | JY | 揭阳分公司 | JY | 3 | 0663 |
| **15** | 4 | MM | 茂名分公司 | MM | 3 | 0668 |
| **16** | 4 | QY | 清远分公司 | QY | 3 | 0763 |
| **17** | 4 | SG | 韶关分公司 | SG | 3 | 0751 |
| **18** | 4 | ST | 汕头分公司 | ST | 3 | 0754 |
| **19** | 4 | SW | 汕尾分公司 | SW | 3 | 0660 |
| **2** | 1 | GZAC | 广州区域中心 | GZ | 2 | 020 |
| **20** | 4 | YF | 云浮分公司 | YF | 3 | 0766 |
| **21** | 4 | YJ | 阳江分公司 | YJ | 3 | 0662 |
| **22** | 4 | ZH | 珠海分公司 | ZH | 3 | 0756 |
| **23** | 4 | ZJ | 湛江分公司 | ZJ | 3 | 0759 |
| **24** | 4 | ZQ | 肇庆分公司 | ZQ | 3 | 0758 |
| **25** | 4 | ZS | 中山分公司 | ZS | 3 | 0760 |
| **3** | 1 | SZAC | 深圳区域中心 | SZ | 2 | 0755 |
| **4** | 1 | MZAC | 梅州运营中心 | MZ | 2 | 0753 |
| **5** | 2 | GZ | 广州分公司 | GZ | 3 | 020 |
| **6** | 3 | SZ | 深圳分公司 | SZ | 3 | 0755 |
| **7** | 4 | MZ | 梅州分公司 | MZ | 3 | 0753 |
| **8** | 4 | CZ | 潮州分公司 | CZ | 3 | 0768 |
| **9** | 4 | DG | 东莞分公司 | DG | 3 | 0769 |

工单类型关联工单“流程类型”

|  |  |  |
| --- | --- | --- |
| **表名称** | **T\_ORDER\_TYPE** | |
| **字段名称** | 类型 | 描述 |
| **ORDER\_TYPE\_ID** | STRING | 类型编号 |
| **ORDER\_TYPE\_CODE** | STRING | 工单类型代码 |
| **ORDER\_TYPE\_NAME** | STRING | 工单类型名称 |
| **ORDER\_TYPE\_REMARK** | STRING | 工单类型说明 |
| **AUTO\_PUBLISH** | STRING | 是否自动发布,1自动,0人工 |
| **TARG\_SYS** | STRING | 目标系统(10) |
| **NEW\_ORDER\_FREQ** | STRING | 新建工单的频度 |

所有字段相关相关说明，放在SYS\_FIELD\_NAME。

字段值字典放在SYS\_STATUS。通过表名和字段可以查询到。

## 工单管理

### 创建工单

创建工单界面：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 新增工单 | | | | | | | |
| 工单名称： |  | 自定义工单名称： |  | 必须外呼： |  |  |  |
| 流程类型： |  | 工单状态： |  | 启用外呼： |  | 信息类型： |  |
| 所属机构： |  | 处理期限： |  | 创建人： |  | 生成时间： |  |
| 备注： |  | | | | | | |
| 新增/修改/删除 | | | | | | | |

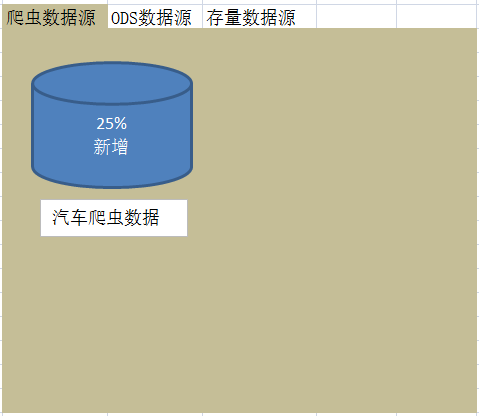
### 添加数据源

#### 工单数据源关系表

工单数据源关系表：

|  |  |  |
| --- | --- | --- |
| 表名称 | T\_ORDER\_DATA\_REL | |
| 字段 | **类型** | **描述** |
| ORDER\_ID | STRING | 工单ID |
| DATA\_ID | STRING | 数据源编号 |
| TRANCOUNT | NUMBER | 转化数据数 |
| ALLCOUNT | NUMBER | 数据总量 |
| CREATE\_DT | DATE | 默认时间 SYSDATE |
| CREATE\_USER | STRING | 建立关联人 |
| TRAN\_DT | DATE | 转化完成时间 |
| FLG | NUMBER | 0 待处理 ，1表示转化完成 |

#### 选择数据源



数据源表：

|  |  |  |
| --- | --- | --- |
| 表名称 | T\_IN\_DATA\_INFO | |
| 字段 | **类型** | **描述** |
| DATA\_ID | STRING | 数据源编号 |
| U32 | STRING | 防篡改标识 |
| TYPE | NUMBER | 数据来源类型 |
| OPERTYPE | NUMBER | 操作类型 |
| IN\_USER | STRING | 数据提供人 |
| CREATE\_DT | DATE | 转化完成时间 |
| USERFLG | NUMBER | 可用状态 |
| Info | STRING | 备注信息 |

**数据来源类型：**

1 表示爬虫数据源

2 表示ODS数据源

3 表示存量数据源

**防篡改标识：**

爬虫数据源 使用原始文件哈希KEY。

ODS数据源 采用当前哈希（ID+数据总条数）

存量数据源 采用当前哈希（ID+数据总条数）

所有哈希采用32位全大写。

**可用状态：**

1 表示可用

0 表示处理中

2 表示不可用

## 数据源制作

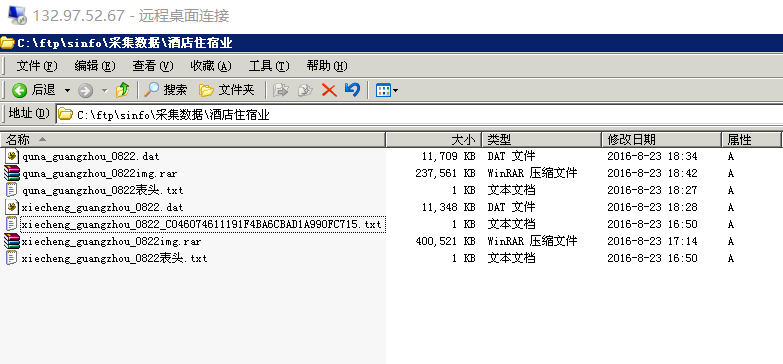
### 爬虫数据源制作

#### 数据读取解析

FTP数据在132.97.52.67上

账号:info

密码：a



##### 强调按照规范放置

所有文件都在SINFO根目录，并且文件存储方式为按照规范命名。

乱放无法通过程序读取，比如：采集数据\酒店住宿业

文件名称：

全英文不要出现中文。

例如：

quna\_guangzhou\_0822.dat

quna\_guangzhou\_0822HEAD.dat

quna\_guangzhou\_KEY0822.dat

数据文本文件名称采用 quna\_guangzhou\_0822+ ".dat"

数据文本结构文件名称采用 quna\_guangzhou\_0822+ " HEAD.dat"

数据文件校验KEY采用 quna\_guangzhou\_0822+ " KEY.dat"

注：必须按照规范方式，否则无法处理。

##### 解析文件数据结构

从根目录读取数据文件为以HEAD.dat结尾文件读取数据内容。

通过固定分割符号解析出数据字段。根据这些字段将数据存储到Mongo DB数据库。

Mongo DB数据库 集合名称 DATA\_SIN

|  |  |  |
| --- | --- | --- |
| 数据部分 | 实体 | 描述 |
| 数据来源信息 | DataHeadinfo | 存储数据来自那个文件，数据哈希值。 |
| 数据内容信息 | DataInfo | 解析实际数据每行JSON结构数据 |
| 数据统计信息 | DataLenInfo | 解析实际数据每行JSON结构数据，计算每个数据内容存储长度。 |
| 存储时间 | CreateDate | 解析入库时间 |

数据实体DataHeadinfo

|  |  |  |
| --- | --- | --- |
| 数据部分 | 实体 | 描述 |
| 数据文件名称 | DataFileName | 存储数据来自那个文件，数据哈希值。 |
| 数据文件校验KEY | DataKEY | 解析实际数据每行JSON结构数据 |
| 数据字段定义数组 | DataField[] | 存储数据字段名称 |

数据实体 DataField

|  |  |  |
| --- | --- | --- |
| 数据部分 | 实体 | 描述 |
| 字段名称 | FieldName | 系统自动生成字段名称。如：F0001 |
| 中文描述 | FieldCName | 结构给出字段名称。如：地市 |

注：因为JSON无法支持中文命名为属性，因此采用自动生成序号英文字段。

字段采用从F0001 扩展，最大支持到9999个字段。

数据实体DataInfo（采用对应转化字段生成实体数据JSON串）

采用JSON接口转化成数据对象数据。

数据统计信息 DataLenInfo（采用对应转化字段生成实体数据JSON串，存储存储空间占用情况）

采用JSON接口转化成数据对象数据。

##### 建立Mongo数据与Oracle数据库传统模型关联

###### 具体实现界面UI

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | 保存 |
| 爬虫数据 | 酒店爬虫数据0822 | 酒店 | 选择 |
|  | 客户信息 | |  |
|  | 商家名称 | 商家名称 |  |
|  | … | … |  |
|  | 电话信息 | |  |
|  | 电话 | 电话 |  |
|  | … | … |  |
|  | 行业深度 | |  |
|  | 停车场 | 停车场 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

基础信息读取字段从：

所有字段相关相关说明，放在SYS\_FIELD\_NAME。

字段值字典放在SYS\_STATUS。通过表名和字段可以查询到。

模型数据通过：

客户深度信息 T\_CONTENT

深度信息模板T\_CONTENT\_TYPE

深度信息模板明细 T\_TEMPLET\_METADATA\_REL

###### 数据库表结构

Mongo DB映射传统数据结构（T\_mongoToOraInfo）

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 描述 |
| ID | STRING | 编号 |
| DataKEY | STRING | 数据文件校验KEY |
| TYPE | STRING | 映射数据类型 |
| MFIELD | STRING | MongoDB数据属性 |
| OFIELD | STRING | Oracle数据字段 |
| FLG | NUMBER | 完成标识 |

DataKEY数据文件校验KEY标识那个数据文件。

TYPE映射数据类型

"T\_CUSTOMER" 表示映射T\_CUSTOMER表。

"T\_CUSTOMER\_TEL" 表示映射T\_CUSTOMER\_TEL表。

"T\_CONTENT" 表示映射T\_ CONTENT表。

对应T\_CONTENT映射按照数据平面模板映射，因此按照没有映射数据采用空串占位方式。

如:只有一个字段映射，其他字段采用空串处理。

##### 线程方式处理映射

需要JAVA服务端增加线程监控，启动映射转化将数据转化到Oracle数据。

#### 数据处理校验

从爬虫数据采集直接同步过来数据存在诸多问题。需要通过校验解决。

##### 号码拆分处理

原始电话信息采用多个电话信息采用固定分割符号分割，需要通过存储过程将其拆分处理出来。

参考爬虫电话号码拆分脚本。

##### 电话号码校验

剔除不符合规范电话号码信息。例如：号码过短、400、800号码等。

##### 电话结合信息补充

补充电话号码在信令中通话次数、114前台报号次数。

##### 匹配存量数据

解决商家名称、电话匹配、电话匹配、商家名称匹配。这个阶段分离完全新增、修改补充数据。

#### 数据打包处理

数据校验后的数据进行打包处理，及修改数据建立映射关系。

打包处理主要涉及将数据从非标准结构转化成传统标准数据。

建议映射主要将生产数据，将存量数据与爬虫数据建立映射关系。

注：映射关系通过存量商家ID 到 爬虫商家ID（新生成商家ID）。

### 现有数据逻辑

工单添加数据主要涉及：

客户信息 T\_CUSTOMER

客户电话信息T\_CUSTOMER\_TEL

客户深度信息 T\_CONTENT

客户深度信息关系 T\_PARTY\_CONTENT\_RELATIONSHIP。

表与表之间关系：

T\_CUSTOMER 与 T\_CUSTOMER\_TEL 通过

NODE\_CODE、CUSTOMER\_ID 关联实现。主要解决客户和客户电话一对多关系。由于业务上需求实际上一对一的。

T\_PARTY\_CONTENT\_RELATIONSHIP、T\_CUSTOMER、T\_CONTENT 通过NODE\_CODE、CUSTOMER\_ID、CONTENT\_ID、PARTY\_ID（ T\_PARTY\_CONTENT\_RELATIONSHIP 表中CUSTOMER\_ID）关联。解决客户与其所在深度信息的联系。

关于深度信息实现方法：

客户深度信息 T\_CONTENT

深度信息模板T\_CONTENT\_TYPE

深度信息模板明细 T\_TEMPLET\_METADATA\_REL

T\_CONTENT. TYPE\_ID 、T\_CONTENT\_TYPE.TYPE\_ID、T\_TEMPLET\_METADATA\_REL. TYPE\_ID进行关联起来。

T\_CONTENT存储实际数据，数据内容通过T\_CONTENT\_TYPE与T\_TEMPLET\_METADATA\_REL实现解析。

例如：

T\_CONTENT\_TYPE 模板类型为“基本微信运营”

|  |  |
| --- | --- |
| **TYPE\_ID** | **TYPE\_NAME** |
| **1000000002980** | 基本微信运营 |

T\_TEMPLET\_METADATA\_REL 存在模板“1000000002980”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEMPLET\_METADATA\_REL\_ID** | **TYPE\_ID** | **METADATA\_FIELD\_ID** | **FIELD\_SEQ** | **FIELD\_ALIAS\_NAME** | **VISIBLE** |
| **1220001849469** | 1000000002980 | 1220000846489 | 1 | 经营范围 | 1 |
| **1220001849470** | 1000000002980 | 1220000846490 | 2 | 微信公众号 | 1 |
| **1220001849510** | 1000000002980 | 1220000846509 | 3 | 微信认证 | 1 |
| **1220001849511** | 1000000002980 | 1220000846510 | 4 | 帐户主体 | 1 |
| **1220001849472** | 1000000002980 | 1220000846491 | 5 | 官网网址 | 1 |
| **1220001849473** | 1000000002980 | 1220000846492 | 6 | 网站备案 | 1 |
| **1220001849474** | 1000000002980 | 1220000846493 | 7 | alexa排名 | 1 |

T\_CONTENT 按照模板给定字段顺序解析实现查询、存储。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CONTENT\_ID** | **TYPE\_ID** | **INFO\_TYPE** | **CONTENT** | **TITLE** | **NODE\_CODE** |
| **1210900887478** | 1000000002980 | 1 | ※※※※http://www.kxsl.com/※粤ICP备12032835号-1※ | 基本微信运营 | 020 |
| **1210900887479** | 1000000002980 | 1 | ※※※※http://www.huazhu.com/Hanting※沪ICP备12041475※919 | 基本微信运营 | 020 |

注：所有信息都通过※分割出来的，分割没有值将为空串。

附加说明：  
T\_METADATA\_FIELD 模板字段表

|  |  |  |
| --- | --- | --- |
| **模板字段** | **类型** | **描述** |
| **METADATA\_FIELD\_ID** | STRING | 扩展信息属性字段表主键,由GET\_ID()生成 |
| **FIELD\_NAME** | STRING | 字段名称(弃用,改为模板关联表中的field\_alias\_name) |
| **FIELD\_TYPE** | STRING | 字段类型:1文本,2整数,3日期,4时间,5日期时间,6小数,7枚举(单选),8集合(多选) |
| **VALUE\_DEFINE** | STRING | 字段类型为枚举或集合类型时,定义可选项,以"|"分隔 |
| **VALUE\_DEFAULT** | STRING | 字段默认值,字段类型为枚举或集合类型时,必须为定义的可选项之一;其他类型应与定义的类型相匹配 |
| **DIST\_NAME** | STRING | 全局唯一名称 |
| **FIELD\_DESC** | STRING | 备注 |
| **ORIGIN\_ID** | STRING |  |
| **FOREIGNTABLE** | STRING | 外键值表(Field\_Type为外键类型时有用) |

### 新模型建设

#### 数据存储模型

#### 模型定义表

模型定义主要定义模型类型名称。

涉及字段定义如下：

编号

名称

英文名称（用于实例化）

结构类型

描述

创建人

创建日期

修订人

修订日期。

**名称**主要是说明模型用于做什么。

**结构类型**主要分类平面结构、立体多层结构。平面结构就是所有属性字段在一个层级上。立体多层结构就是所有属性可以无限扩展结构。这种结构通常理解为二叉树结构形式。在互联网定义其实还存在星型结构，但本文档目前仅仅支持二叉树结构。

#### 属性定义表

|  |  |  |
| --- | --- | --- |
| **属性字段** | **类型** | **描述** |
| **模板ID** | STRING | 用于关联模型 |
| **属性ID** | STRING | 相同模板唯一标识 |
| **属性父ID** | STRING | 上级唯一标识，顶级该标识默认为“0” |
| **属性名称** | STRING | 属性中文名称 |
| **属性英文名称** | STRING | 用于实例化 |
| **属性父名称** | STRING | 属性中文名称 |
| **属性父英文名称** | STRING | 用于实例化 |
| **属性类型** | STRING | 支持数字、文本、日期、枚举、数组。 |
| **属性默认值** | STRING | 数字为“0”文本为空串，日期默认为“2016-01-01 00：00：000”  枚举默认为枚举默认值  数组默认为为NULL。 |

数据样例：

{

"customerid":"xxxx",

" customername":"xxxx",

"customerTels":[

{"areacode":"020",

"telnum":"38637089"},

{"areacode":"020",

"telnum":"38637086"}

]

}

注：父节点通常将解析为对象类型。例如例子中电话将解析为CustomerTel的对象数组。