TotalLINK

产品手册

上海朝识智能科技有限公司

2020 年 8 月

文档控制

■ 主要内容

本文整合 TotalLINK 系统调用各种 API 的方式，请根据需求设计模型。

■ 更改记录

日期	版本	作者	备注
2019-04	1.0	Dina	初始发布
2020-04-02	1.1	Liz	增加SOAP集成接口
2020-08-31	2.0	Jozey	增加物流追踪、发票识别、人脸识别、钉钉日志调用内容
2020-12-13	3.0	Jozey	新增钉钉请假调用实例

■ 支持版本

非特殊说明的功能，默认前后版本都支持

仅支持T20版本及以后版本的功能点

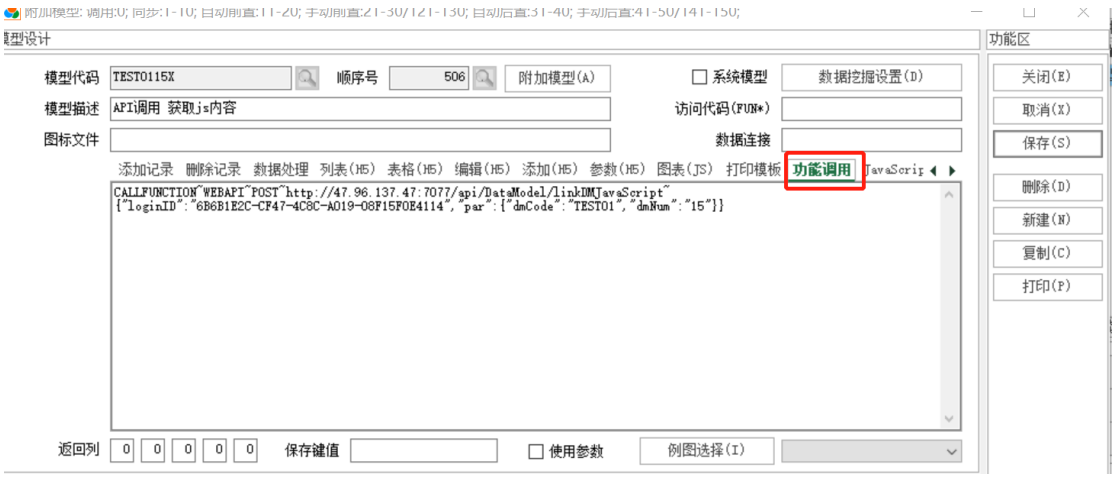
API 调用配置

目　录

1 API 调用设置

TotalLINK 在“功能调用”页签中，通过 CALLFUNCTION 关键字和对应的脚本格式可实现各种接口的调用。

如：各类业务系统、快递物流信息、新闻头条、货币汇率、基金财务数据等：

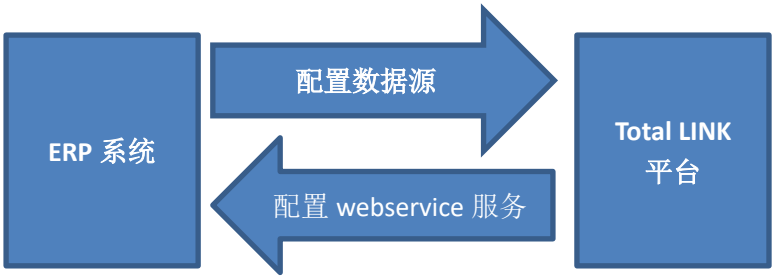


2 业务系统 API 调用

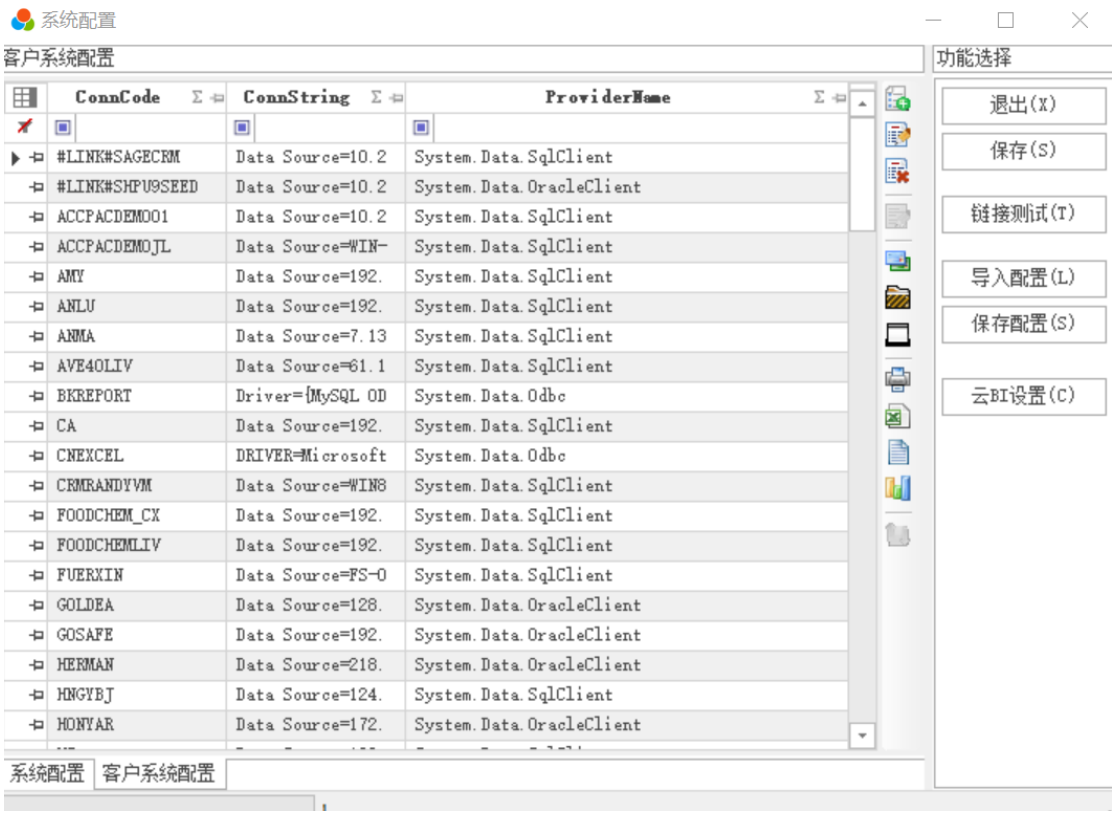
业务系统会有对应开放的接口调用方式，通过在 TotalLINK 平台上做对应的调用参数设置，实现两个系统的数据对接。

具体应用见下面两个实例。

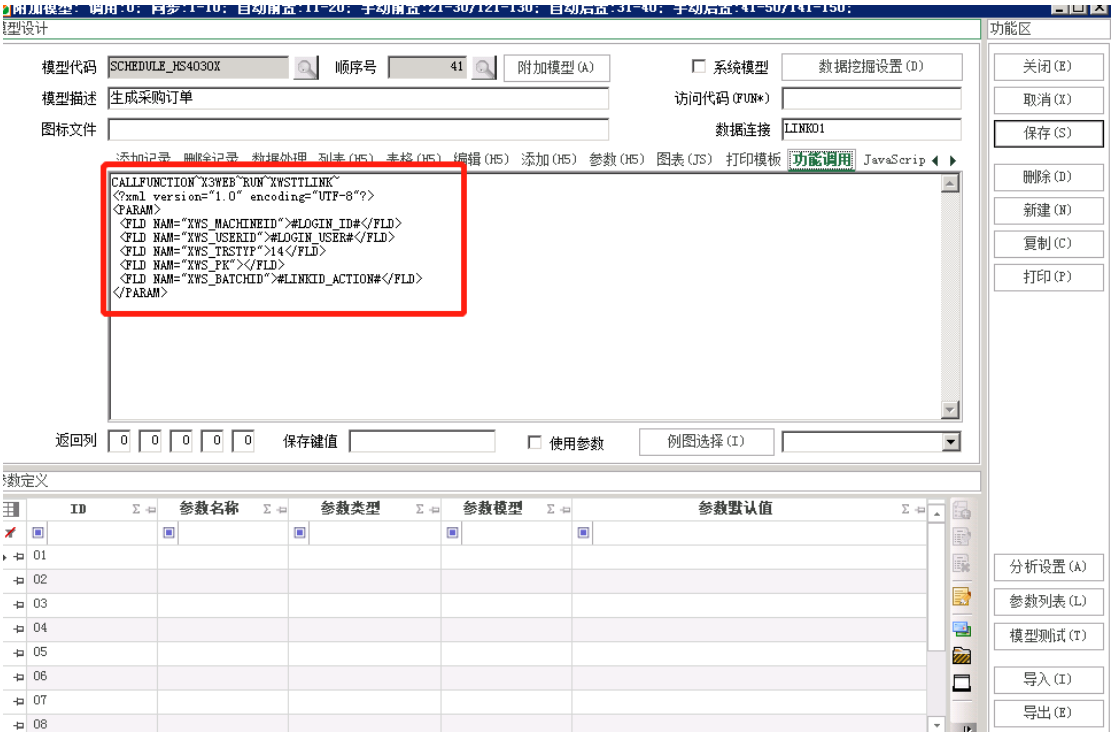
2.1 SAGEX3 ERP 系统



■ 通过配置数据源信息，可实现数据从 ERP 到 TotalLINK 平台的调用



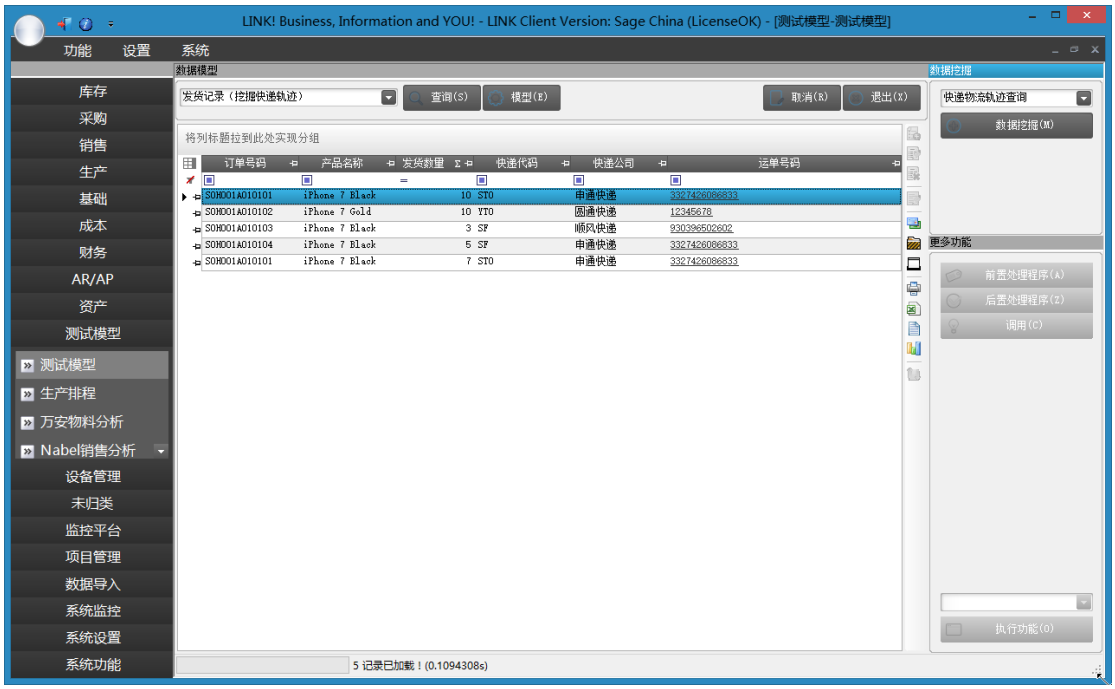
通过配置 webservice 服务，可将 TotalLINK 处理后的数据传回到 ERP：



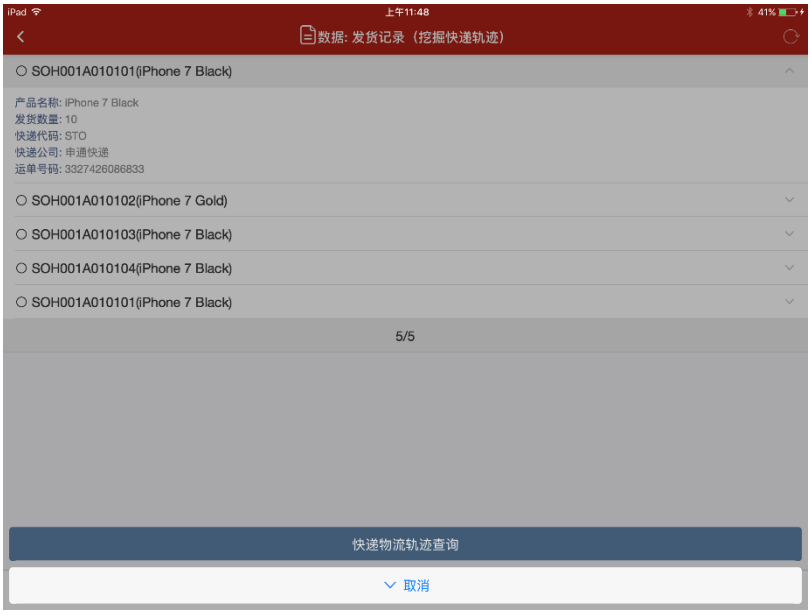
ERP 系统中配置 webservice 传递的参数，以导入模板的方式将 TotalLINK 处理的结果传至 ERP 系统，生成对应的单据。

2.2 物流信息追踪

2. 2. 1物流信息与业务系统的集成

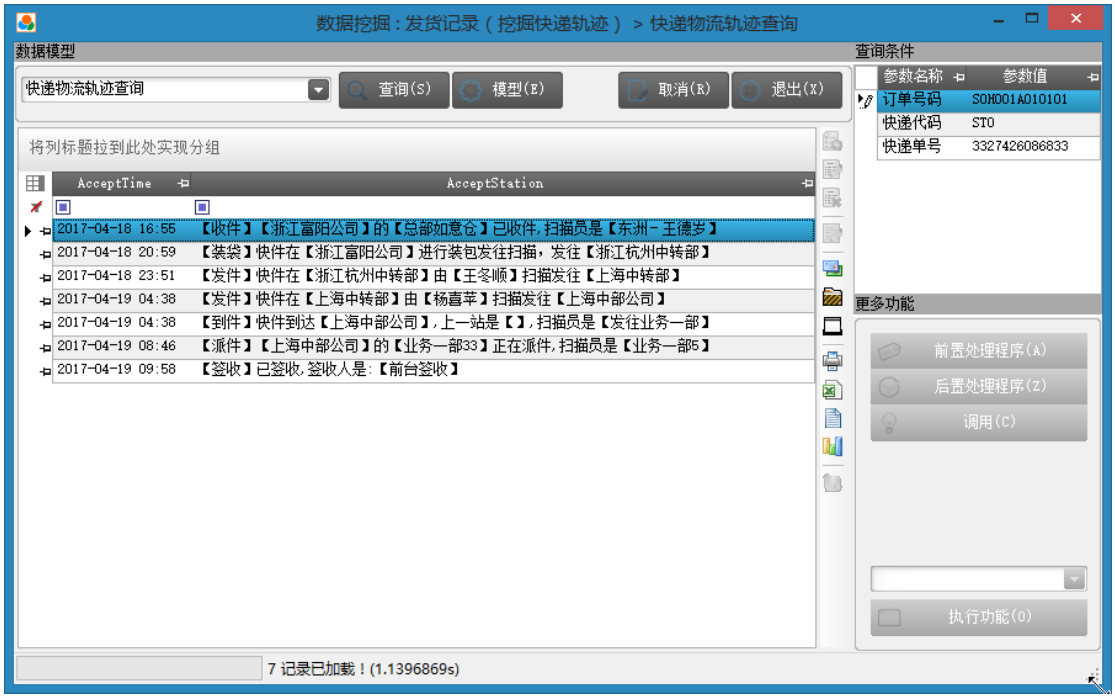


- TotalLINK 系统整合了各种业务系统和各类物流公司的信息接口
- 如上图所示，在 ERP 等业务系统中记录了订单的发货信息，通过关联模型即可查询到物流公司的具体物流追踪信息



- 如图所示，是在移动设备上查询到的发货记录

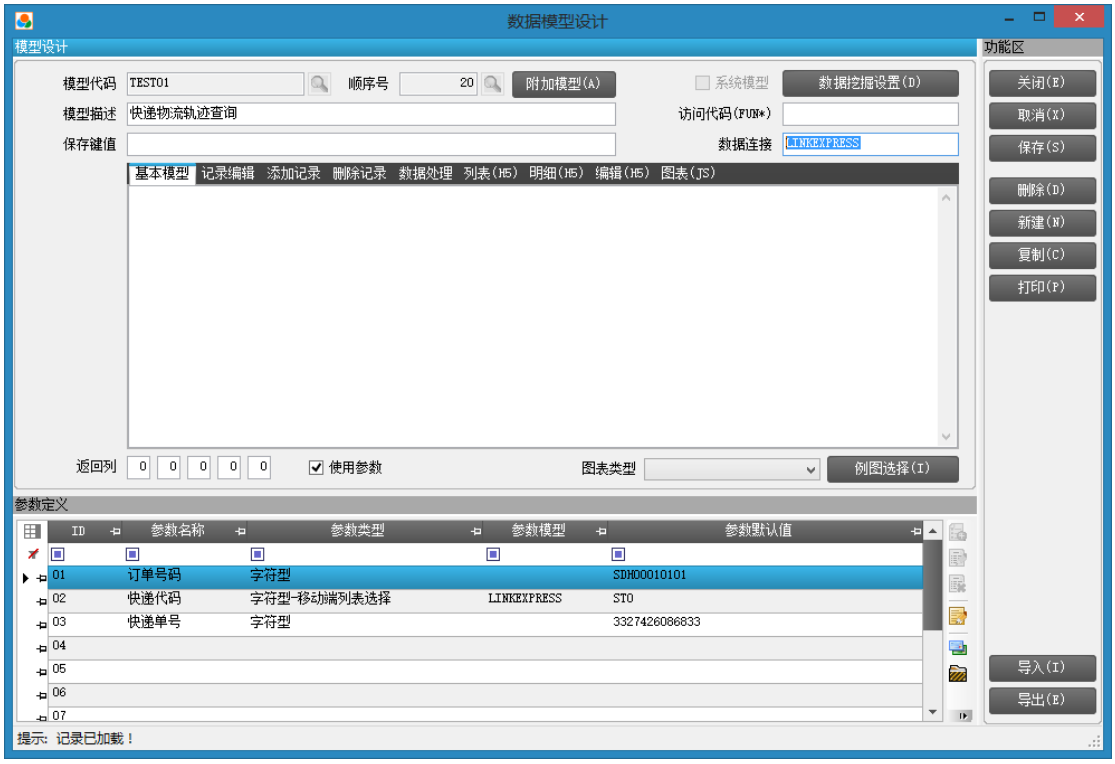
2. 2. 2物流追踪信息



- 如图所示，得到了物流追踪的具体信息

- 如图所示，是在移动设备上查询到的物流追踪信息

2. 2. 3物流追踪模型配置



- 如图所示，对于物流信息追踪的实现，需要在系统中配置一个特殊的数据模型
- 该模型需要使用一个特殊的数据链接代码：LINKEXPRESS
- 同时，该模型需要配置三个参数
  - 参数一
    - 订单号码/发货单号等业务单据的号码
    - 该参数是可选的，信息可以来自于 ERP 等业务系统
  - 参数二
    - 快递代码/物流公司代码
    - 为了实现不同物流公司的全面业务对接，必须使用统一的物流公司代码规范，目前已经支持一百多家物流公司的信息追踪
    - 具体的公司代码见下一节内容
  - 参数三
    - 快递单号/物流单号
    - 该信息与物流公司代码组合可以唯一确定物流的运单信息
- 如果需要和 ERP 等业务系统的单据进行关联查询，只需要将该模型作为数据挖掘模型即可

2.2.4物流公司代码对照表

2.2.4.1 国内物流公司

编码	名称
AJ	安捷快递
ANE	安能物流
AXD	安信达快递
BQXHM	北青小红帽
BFDF	百福东方
BTWL	百世快运
CCES	CCES 快递
CITY100	城市 100
COE	COE 东方快递
CSCY	长沙创一
CDSTKY	成都善途速运
DBL	德邦
DSWL	D 速物流
DTWL	大田物流
EMS	EMS
FAST	快捷速递
FEDEX	FEDEX 联邦(国内件)
FEDEX_GJ	FEDEX 联邦(国际件)
FKD	飞康达
GDEMS	广东邮政
GSD	共速达
GTO	国通快递
GTSD	高铁速递
HFWL	汇丰物流
HHTT	天天快递
HLWL	恒路物流
HOAU	天地华宇
hq568	华强物流
HTKY	百世快递
HXLWL	华夏龙物流
HYLSD	好来运快递
JGSD	京广速递
JIUYE	九曳供应链

JJKY	佳吉快运
JLDT	嘉里物流
JTKD	捷特快递
JXD	急先达
JYKD	晋越快递
JYM	加运美
JYWL	佳怡物流
KYWL	跨越物流
LB	龙邦快递
LHT	联昊通速递
MHKD	民航快递
MLWL	明亮物流
NEDA	能达速递
PADTF	平安达腾飞快递
QCKD	全晨快递
QFKD	全峰快递
QRT	全日通快递
RFD	如风达
SAD	赛澳递
SAWL	圣安物流
SBWL	盛邦物流
SDWL	上大物流
SF	顺丰快递
SFWL	盛丰物流
SHWL	盛辉物流
ST	速通物流
STO	申通快递
STWL	速腾快递
SURE	速尔快递
TSSTO	唐山申通
UAPEX	全一快递
UC	优速快递
WJWL	万家物流
WXWL	万象物流
XBWL	新邦物流
XFEX	信丰快递
XYT	希优特
XJ	新杰物流

YADEX	源安达快递
YCWL	远成物流
YD	韵达快递
YDH	义达国际物流
YFEX	越丰物流
YFHEX	原飞航物流
YFSD	亚风快递
YTKD	运通快递
YTO	圆通速递
YXKD	亿翔快递
YZPY	邮政平邮/小包
ZENY	增益快递
ZHQKD	汇强快递
ZJS	宅急送
ZTE	众通快递
ZTKY	中铁快运
ZTO	中通速递
ZTWL	中铁物流
ZYWL	中邮物流
AMAZON	亚马逊物流
SUBIDA	速必达物流
RFEX	瑞丰速递
QUICK	快客快递
CJKD	城际快递
CNPEX	CNPEX 中邮快递
HOTSCM	鸿桥供应链
HPTEX	海派通物流公司
AYCA	澳邮专线
PANEX	泛捷快递
PCA	PCA Express
UEQ	UEQ Express

2.2.4.2 国外物流公司

编码	名称
AAE	AAE 全球专递
ACS	ACS 雅仕快递
ADP	ADP Express Tracking

ANGUILAYOU	安圭拉邮政
AOMENYZ	澳门邮政
APAC	APAC
ARAMEX	Aramex
AT	奥地利邮政
AUSTRALIA	Australia Post Tracking
BEL	比利时邮政
BHT	BHT 快递
BILUYOUZHE	秘鲁邮政
BR	巴西邮政
BUDANYOUZH	不丹邮政
CA	加拿大邮政
D4PX	递四方速递
DHL	DHL
DHL_EN	DHL(英文版)
DHL_GLB	DHL 全球
DHLGM	DHL Global Mail
DK	丹麦邮政
DPD	DPD
DPEX	DPEX
EMSGJ	EMS 国际
ESHIPPER	EShipper
GJEYB	国际 e 邮宝
GJYZ	国际邮政包裹
GLS	GLS
IADLSQDYZ	安的列斯群岛邮政
IADLYYZ	澳大利亚邮政
IAEBNYYZ	阿尔巴尼亚邮政
IAEJLYYZ	阿尔及利亚邮政
IAFHYZ	阿富汗邮政
IAGLYZ	安哥拉邮政
IAGTYZ	阿根廷邮政
IAJYZ	埃及邮政
IALBYZ	阿鲁巴邮政
IALQDYZ	奥兰群岛邮政
IALYYZ	阿联酋邮政
IAMYZ	阿曼邮政
IASBJYZ	阿塞拜疆邮政

IASEBYYZ	埃塞俄比亚邮政
IASNYYZ	爱沙尼亚邮政
IASSDYZ	阿森松岛邮政
IBCWNYZ	博茨瓦纳邮政
IBDLGYZ	波多黎各邮政
IBDYZ	冰岛邮政
IBELSYZ	白俄罗斯邮政
IBHYZ	波黑邮政
IBJLYYZ	保加利亚邮政
IBJSTYZ	巴基斯坦邮政
IBLNYZ	黎巴嫩邮政
IBLSD	便利速递
IBLWYYZ	玻利维亚邮政
IBLYZ	巴林邮政
IBMDYZ	百慕达邮政
IBOLYZ	波兰邮政
IBTD	宝通达
IBYB	贝邮宝
ICKY	出口易
IDFWL	达方物流
IDGYZ	德国邮政
IE	爱尔兰邮政
IEGDEYZ	厄瓜多尔邮政
IELSYZ	俄罗斯邮政
IELTLYYZ	厄立特里亚邮政
IFTWL	飞特物流
IGDLPDEMS	瓜德罗普岛 EMS
IGDLPDYZ	瓜德罗普岛邮政
IGJESD	俄速递
IGLBYYZ	哥伦比亚邮政
IGLLYZ	格陵兰邮政
IGSDLJYZ	哥斯达黎加邮政
IHGYZ	韩国邮政
IHHWL	华翰物流
IHLY	互联易
IHSKSTYZ	哈萨克斯坦邮政
IHSYZ	黑山邮政
IJBBWYZ	津巴布韦邮政

IJEJSSTYZ	吉尔吉斯斯坦邮政
IJKYZ	捷克邮政
IJNYZ	加纳邮政
IJPZYZ	柬埔寨邮政
IKNDYYZ	克罗地亚邮政
IKNYYZ	肯尼亚邮政
IKTDWEMS	科特迪瓦 EMS
IKTDWYZ	科特迪瓦邮政
IKTEYZ	卡塔尔邮政
ILBYYZ	利比亚邮政
ILKKD	林克快递
ILMNYYZ	罗马尼亚邮政
ILSBYZ	卢森堡邮政
ILTWYYZ	拉脱维亚邮政
ILTWYZ	立陶宛邮政
ILZDSDYZ	列支敦士登邮政
IMEDFYZ	马尔代夫邮政
IMEDWYZ	摩尔多瓦邮政
IMETYZ	马耳他邮政
IMJLGEMS	孟加拉国 EMS
IMLGYZ	摩洛哥邮政
IMLQSYZ	毛里求斯邮政
IMLXYEMS	马来西亚 EMS
IMLXYYZ	马来西亚邮政
IMQDYZ	马其顿邮政
IMTNKEMS	马提尼克 EMS
IMTNKYZ	马提尼克邮政
IMXGYZ	墨西哥邮政
INFYZ	南非邮政
INRLYYZ	尼日利亚邮政
INWYZ	挪威邮政
IPTYYZ	葡萄牙邮政
IQQKD	全球快递
IQTWL	全通物流
ISDYZ	苏丹邮政
ISEWDYZ	萨尔瓦多邮政
ISEWYYZ	塞尔维亚邮政
ISLFKYZ	斯洛伐克邮政

ISLWNYYZ	斯洛文尼亚邮政
ISNJEYZ	塞内加尔邮政
ISPLSYZ	塞浦路斯邮政
ISTALBYZ	沙特阿拉伯邮政
ITEQYZ	土耳其邮政
ITGYZ	泰国邮政
ITLNDHDBGE	特立尼达和多巴哥 EMS
ITNSYZ	突尼斯邮政
ITSNYYZ	坦桑尼亚邮政
IWDMLYZ	危地马拉邮政
IWGDYZ	乌干达邮政
IWKLEMS	乌克兰 EMS
IWKLYZ	乌克兰邮政
IWLGYZ	乌拉圭邮政
IWLYZ	文莱邮政
IWZBKSTEMS	乌兹别克斯坦 EMS
IWZBKSTYZ	乌兹别克斯坦邮政
IXBYYZ	西班牙邮政
IXFLWL	小飞龙物流
IXGLDNYYZ	新喀里多尼亚邮政
IXJPEMS	新加坡 EMS
IXJPYZ	新加坡邮政
IXLYYZ	叙利亚邮政
IXLYZ	希腊邮政
IXPSJ	夏浦世纪
IXPWL	夏浦物流
IXXLYZ	新西兰邮政
IXYLYZ	匈牙利邮政
IYDLYZ	意大利邮政
IYDNXYYZ	印度尼西亚邮政
IYDYZ	印度邮政
IYGYZ	英国邮政
IYLYZ	伊朗邮政
IYMNYYZ	亚美尼亚邮政
IYMYZ	也门邮政
IYNYZ	越南邮政
IYSLYZ	以色列邮政
IYTG	易通关

IYWWL	燕文物流
IZBLTYZ	直布罗陀邮政
IZLYZ	智利邮政
JP	日本邮政
NL	荷兰邮政
ONTRAC	ONTRAC
QQYZ	全球邮政
RDSE	瑞典邮政
SWCH	瑞士邮政
TAIWANYZ	台湾邮政
TNT	TNT 快递
UPS	UPS
USPS	USPS 美国邮政
YAMA	日本大和运输(Yamato)
YODEL	YODEL
YUEDANYOUZ	约旦邮政

2.2.4.3 转运物流公司

编码	名称
ZY_AG	爱购转运
ZY_AOZ	爱欧洲
ZY_AUSE	澳世速递
ZY_AXO	AXO
ZY_AZY	澳转运
ZY_BDA	八达网
ZY_BEE	蜜蜂速递
ZY_BH	贝海速递
ZY_BL	百利快递
ZY_BM	斑马物流
ZY_BOZ	败欧洲
ZY_BT	百通物流
ZY_BYECO	贝易购
ZY_CM	策马转运
ZY_CTM	赤兔马转运
ZY_CUL	CUL 中美速递
ZY_DGHT	德国海淘之家
ZY_DYW	德运网

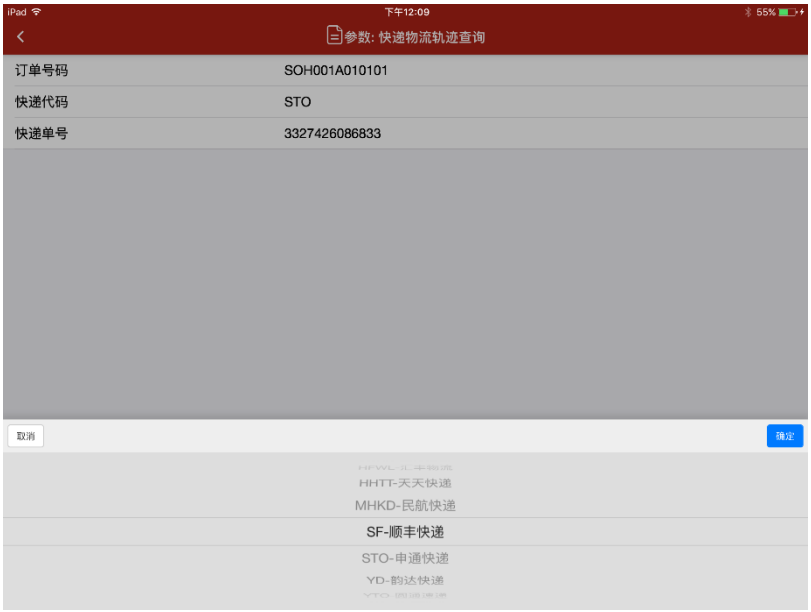
ZY_EFS	EFS POST
ZY_ESONG	宜送转运
ZY_ETD	ETD
ZY_FD	飞碟快递
ZY_FG	飞鸽快递
ZY_FLSD	风雷速递
ZY_FX	风行快递
ZY_FXSD	风行速递
ZY_FY	飞洋快递
ZY_HC	皓晨快递
ZY_HCYD	皓晨优递
ZY_HDB	海带宝
ZY_HFMZ	汇丰美中速递
ZY_HJSD	豪杰速递
ZY_HTAO	360hitao 转运
ZY_HTCUN	海淘村
ZY_HTKE	365 海淘客
ZY_HTONG	华通快运
ZY_HXKD	海星桥快递
ZY_HXSY	华兴速运
ZY_HYSD	海悦速递
ZY_IHERB	LogisticsY
ZY_JA	君安快递
ZY_JD	时代转运
ZY_JDKD	骏达快递
ZY_JDZY	骏达转运
ZY_JH	久禾快递
ZY_JHT	金海淘
ZY_LBZY	联邦转运 FedRoad
ZY_LPZ	领跑者快递
ZY_LX	龙象快递
ZY_LZWL	量子物流
ZY_MBZY	明邦转运
ZY_MGZY	美国转运
ZY_MJ	美嘉快递
ZY_MST	美速通
ZY_MXZY	美西转运
ZY_MZ	168 美中快递

ZY_OEJ	欧 e 捷
ZY_OZF	欧洲疯
ZY_OZGO	欧洲 GO
ZY_QMT	全美通
ZY_QQEX	QQ-EX
ZY_RDGJ	润东国际快线
ZY_RT	瑞天快递
ZY_RTSD	瑞天速递
ZY_SCS	SCS 国际物流
ZY_SDKD	速达快递
ZY_SFZY	四方转运
ZY_SOHO	SOHO 苏豪国际
ZY_SONIC	Sonic-Ex 速递
ZY_ST	上腾快递
ZY_TCM	通诚美中快递
ZY_TJ	天际快递
ZY_TM	天马转运
ZY_TN	滕牛快递
ZY_TPAK	TrakPak
ZY_TPY	太平洋快递
ZY_TSZ	唐三藏转运
ZY_TTHT	天天海淘
ZY_TWC	TWC 转运世界
ZY_TX	同心快递
ZY_TY	天翼快递
ZY_TZH	同舟快递
ZY_UCS	UCS 合众快递
ZY_WDCS	文达国际 DCS
ZY_XC	星辰快递
ZY_XDKD	迅达快递
ZY_XDSY	信达速运
ZY_XF	先锋快递
ZY_XGX	新干线快递
ZY_XIYJ	西邮寄
ZY_XJ	信捷转运
ZY_YGKD	优购快递
ZY_YJSD	友家速递(UCS)
ZY_YPW	云畔网

ZY_YQ	云骑快递
ZY_YQWL	一柒物流
ZY_YSSD	优晟速递
ZY_YSW	易送网
ZY_YTUSA	运淘美国
ZY_ZCSD	至诚速递

2.2.5物流公司选择

- 如图所示，可以通过列表选择上节列表中的国内、国外、转运物流公司
- 可以根据本公司的需要，配置可以选择的物流公司



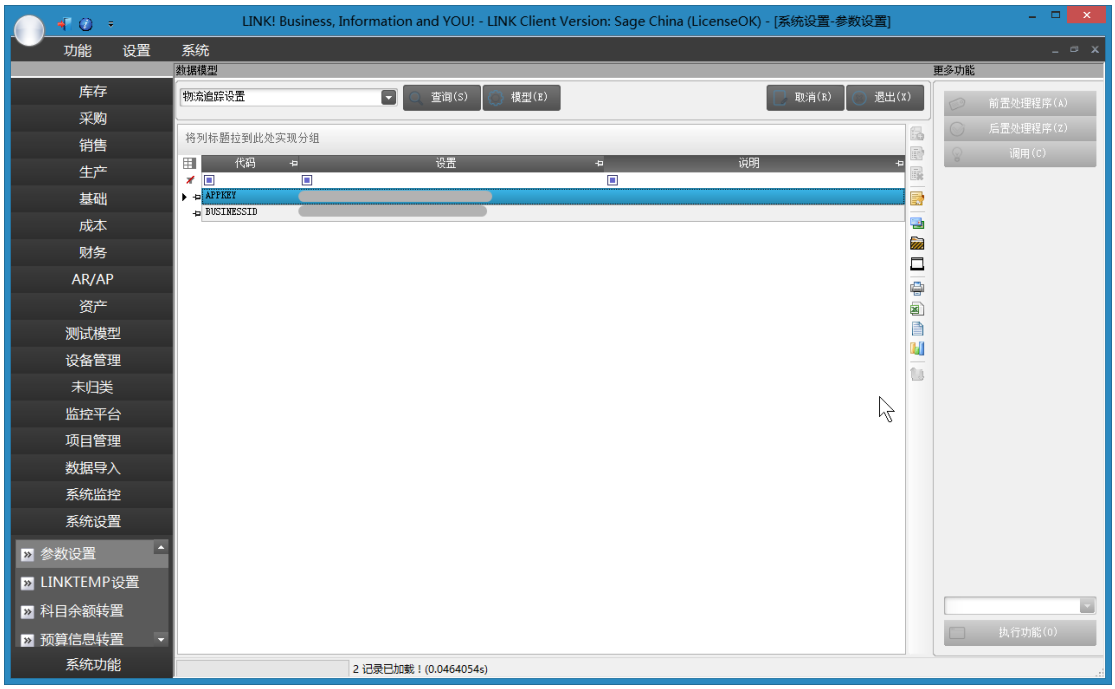
- 在移动端的列表选择功能如图所示

2. 2. 6物流追踪功能申请

TotalLink 系统采用合作伙伴“快递鸟”物流信息追踪的接口方案，具体接口功能申请请参考如下网站。

http://www.kdniao.com

2. 2. 7物流追踪参数设置



- 如图所示，实现物流信息追踪功能，需要在系统中设置两个参数
- 参数一：APPKEY
  - 从上述合作伙伴处申请到的 APPKEY
- 参数二：BUSINESSID
  - 从上述合作伙伴处申请到的 BUSINESSID

3 Web API 调用

通过 Web API 提供的调用方式，在 TotalLINK 设置对应的调用脚本，可实现信息调用，同时可将调用的信息保存到指定的表中。

3.1 调用新闻头条信息

相关的 Web API 会有特定的开放式调用方式，例如下方:

接口地址：http://v.juhe.cn/toutiao/index

返回格式：json

请求方式：get/post

请求示例：http://v.juhe.cn/toutiao/index?type=top&key=APPKEY

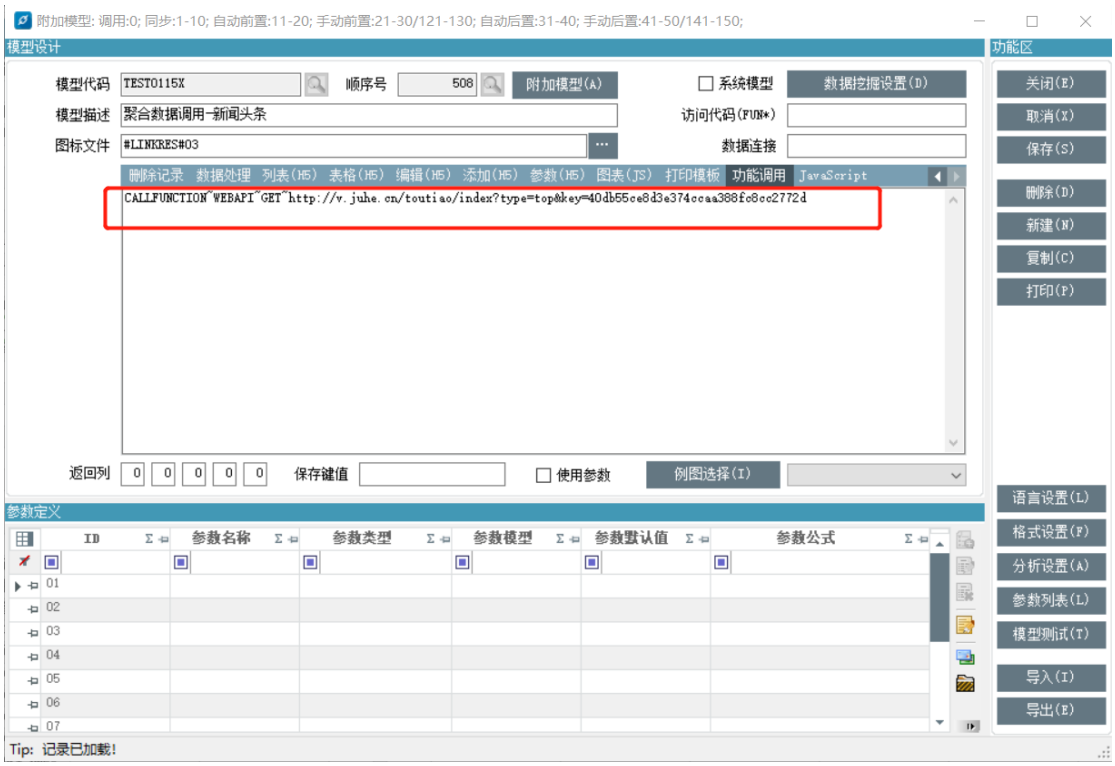
接口备注：返回头条，社会，国内，娱乐，体育，军事，科技，财经，时尚等新闻信息

API测试工具

请求参数说明：

名称	必填	类型	说明
key	是	string	应用APPKEY
type	否	string	类型,,top(头条，默认),shehui(社会),guonei(国内),guoji(国际),yule(娱乐),tiyu(体育)junshi(军事),keji(科技),caijing(财经),shishang(时尚)

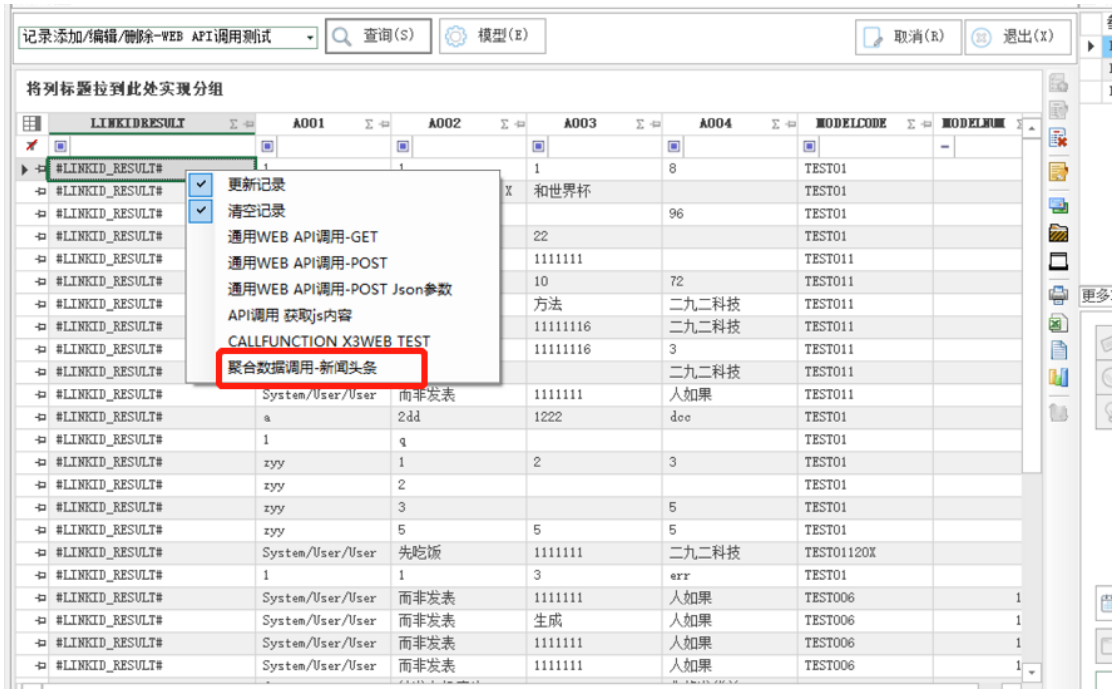
- 添加一个 508 模型，在“功能调用”模块写入以下内容：




```
CALLFUNCTION~WEBAPI~GET~http://v.juhe.cn/toutiao/index?type=top&key=40db55ce8d3e374ccaa388fc8cc2772d
```

关键字：CALLFUNCTION ~LINKAPI （或者 WEBAPI）

- 点击“聚合数据调用-新闻头条”，调用新闻头条的 API，调用结果保存在 LINKAPIRESULT 表中。

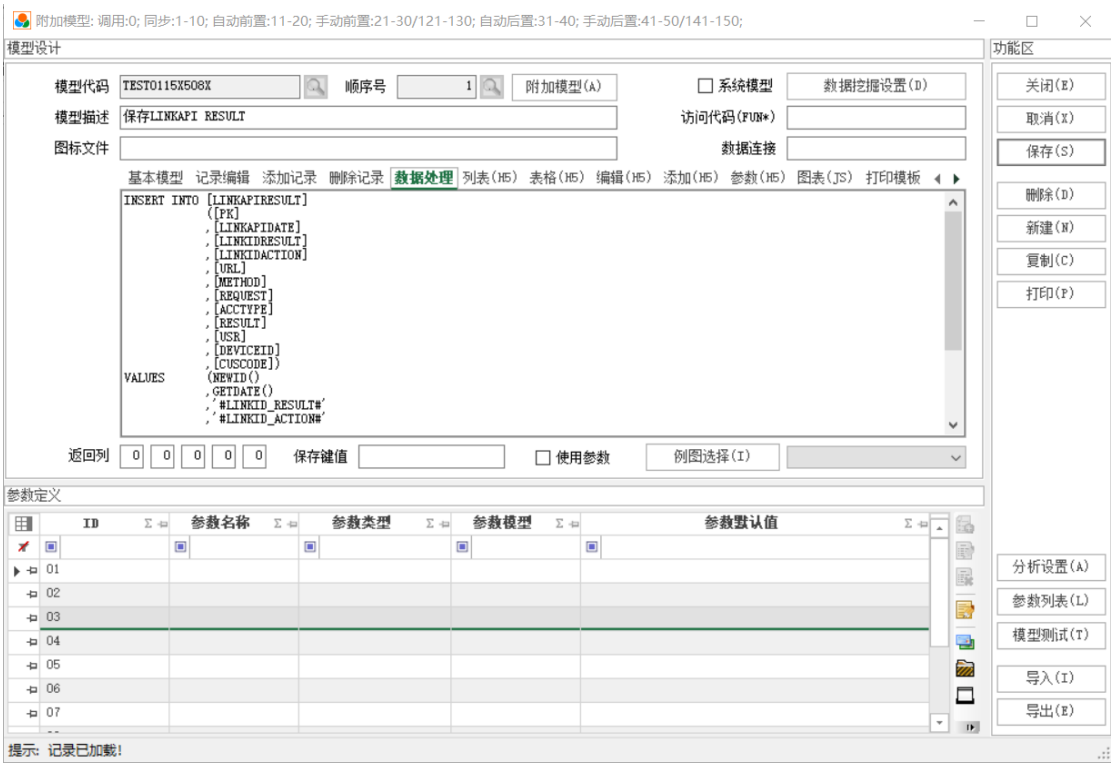


保存方式：触发保存功能，使用 508 附加模型的附加模型-同步模型 1 作为执行 API 保存的

模型。（TotalLINK 支持附加模型后添加一个同步模型处理数据。）

- 508 附加模型里添加一个同步模型：

同步模型将获得的数据插入到相对应的表中：




```
INSERT INTO [LINKAPIRESULT]
           ([PK]
           ,[LINKAPIDATE]
           ,[LINKIDRESULT]
           ,[LINKIDACTION]
           ,[URL]
           ,[METHOD]
           ,[REQUEST]
           ,[ACCTYPE]
           ,[RESULT]
           ,[USR]
           ,[DEVICEID]
           ,[CUSCODE])
          VALUES      (NEWID()
           ,GETDATE()
           ,'#LINKID_RESULT#'
           ,'#LINKID_ACTION#'
           ,'#LINKAPI_URL#'
           ,'#LINKAPI_METHOD#'
```

```
                ,'#LINKAPI_REQUEST#'
                ,'#LINKAPI_ACCTYPE#'
                ,'#LINKAPI_RESULT#'
                ,'#LOGIN_USER#'
                ,'#LOGIN_ID#'
                ,'#LINKCUSCODE#')
```

其中参数列表，0-6，含义如下

sParams(0) = sLinkIDResult

sParams(1) = sLinkIdAction

sParams(2) = sUrl

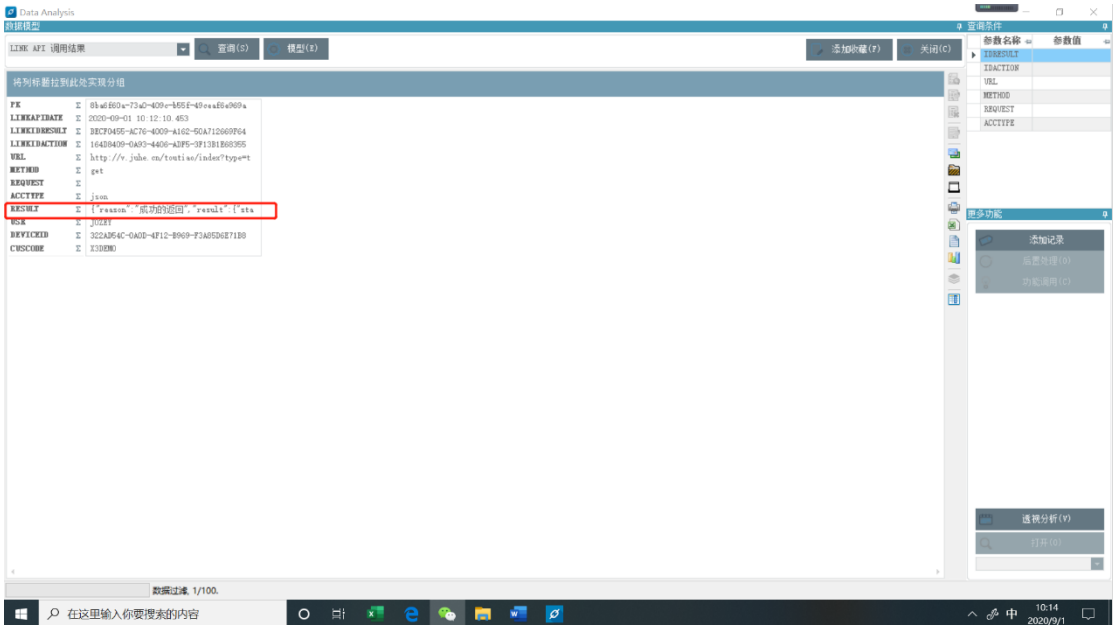
sParams(3) = sMethod

sParams(4) = requestJson

sParams(5) = acceptDataType

sParams(6) = result

查看保存的调用结果：



注：

- 通过模型的方式保存调用的结果会相对比较灵活，通过在保存模型中的不同设置，可以实现保存到不同业务系统，不同的表中。

- LINKID_ACTION：

  替换脚本中的 #LINKID_ACTION# 为统一的 GUID 号码，每次执行附加模型的动作

  sSQL = sSQL.Replace("#LINKID_ACTION#", Me.mLinkIdAction)

- LINKID_RESULT：

替换脚本中的 #LINKID_RESULT# 为统一的 GUID 号码，每次执行附加模型的动作

sSQL = sSQL.Replace("#LINKID_RESULT#", Me.mLinkIDResult)

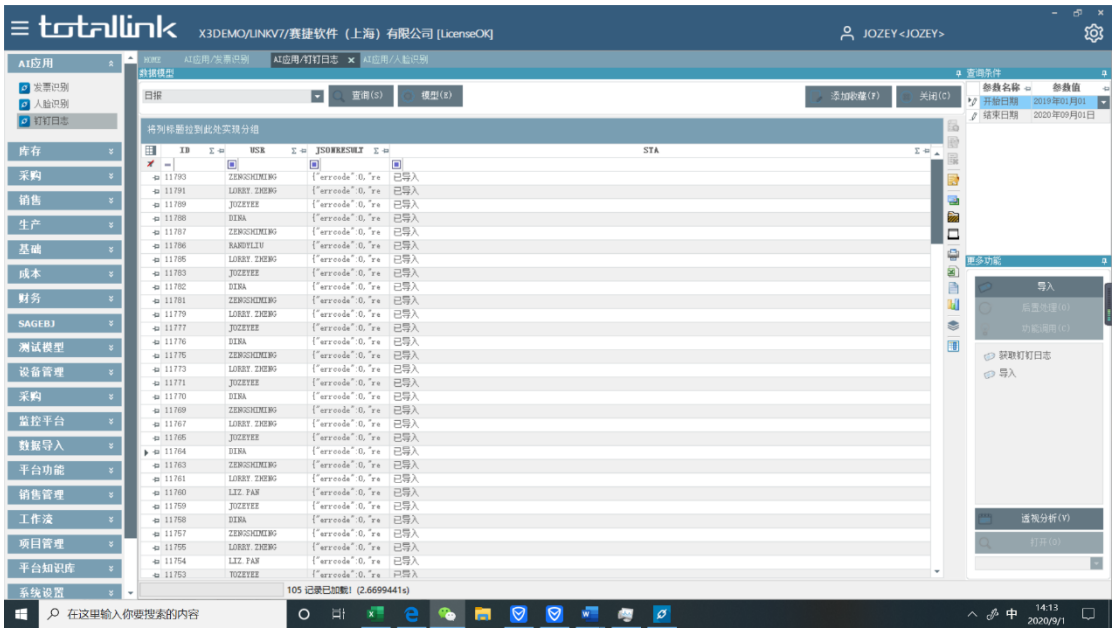
3.2 调用钉钉、微信的接口

TotalLINK 可以调用钉钉或微信的各种接口，比如获取签到记录、日志记录等。

下面介绍 TotalLINK 调用钉钉接口获取钉钉日志、钉钉请假的实例。(通过"#LINKDINGTALKTOKEN”, "#LINKWEIXINTOKEN#"引用 TOKEN 调用)

3. 2. 1钉钉日志

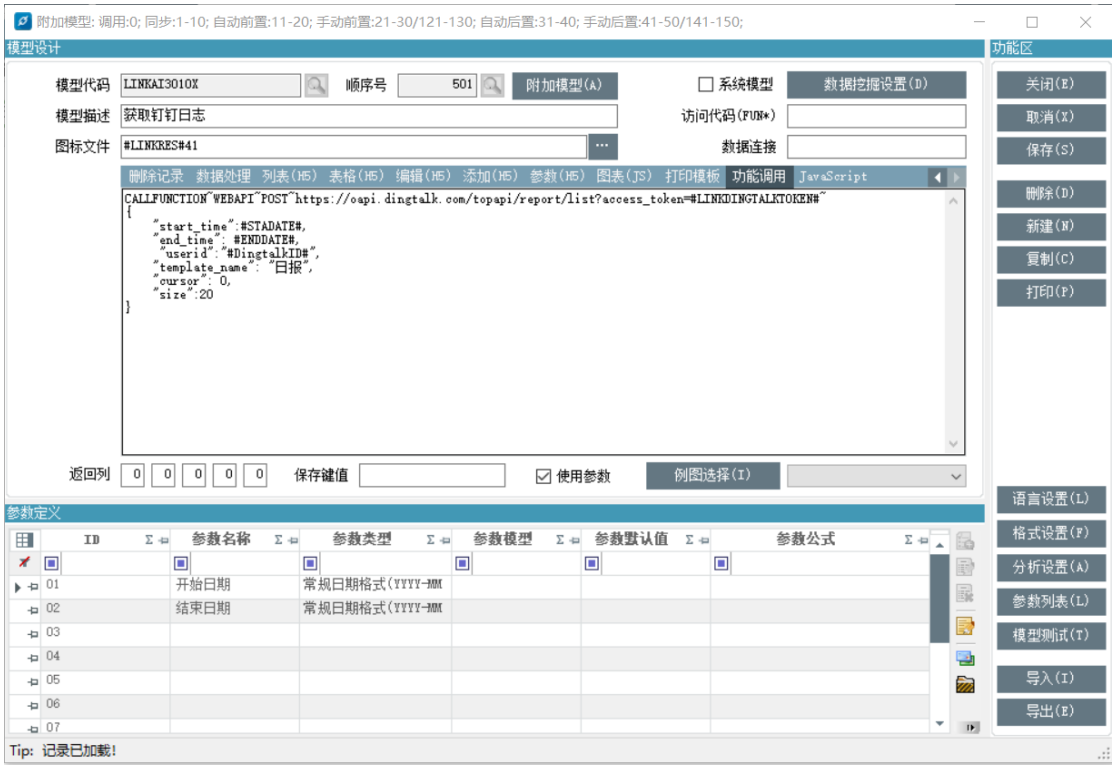
- 主模型查询结果如下：



参考代码：

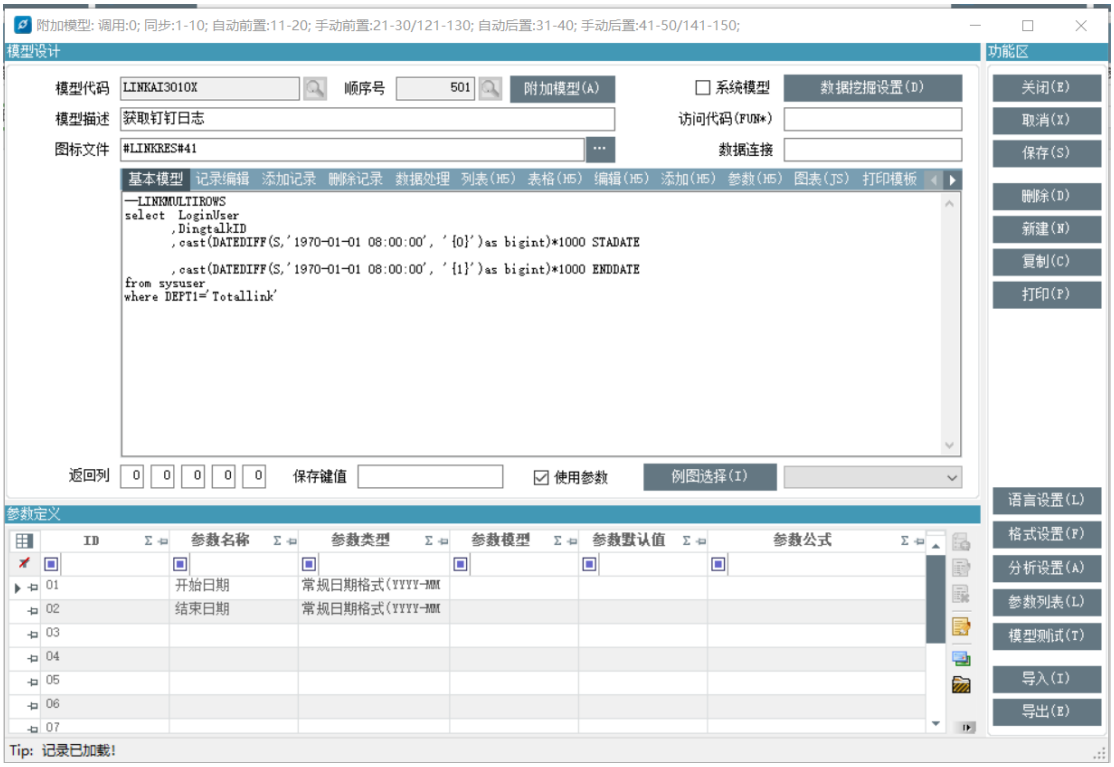
```
select
ID  ,
USR,
A008 JSONRESULT,
CASE WHEN A009='IMP'THEN N'已导入' else N'未导入'
end as STA
from LINKTEMP  WHERE KEYWORDS='Dingtalk'
order by ID DESC;
```

- 添加一个 501 附加模型，在功能调用中写入以下内容。用附加模型处理多行数据的形式获取日志信息。
- 下面的语句表示从钉钉接口获取日志信息，最大获取 20 条，起末时间来自 501 基本模型模块查询的数据：

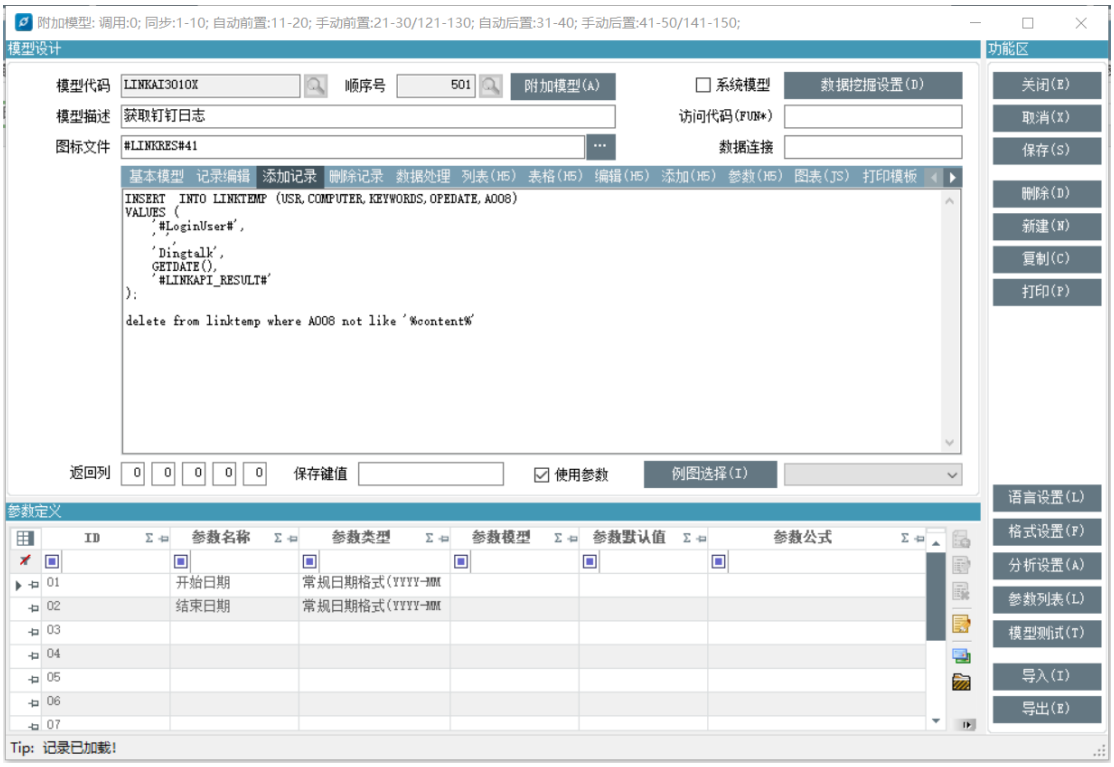



```
CALLFUNCTION~WEBAPI~POST~https://oapi.dingtalk.com/topapi/report/list?access_toke
n=#LINKDINGTALKTOKEN#~
{
    "start_time":#STADATE#,
    "end_time": #ENDDATE#,
    "userid":"#DingtalkID#",
    "template_name": "日报",
    "cursor": 0,
    "size":20
}
```

- 下面是 501 基本模型 模块的代码：



- 添加记录模块写 INSERT 语句，将获取到的钉钉信息插入到中间表中

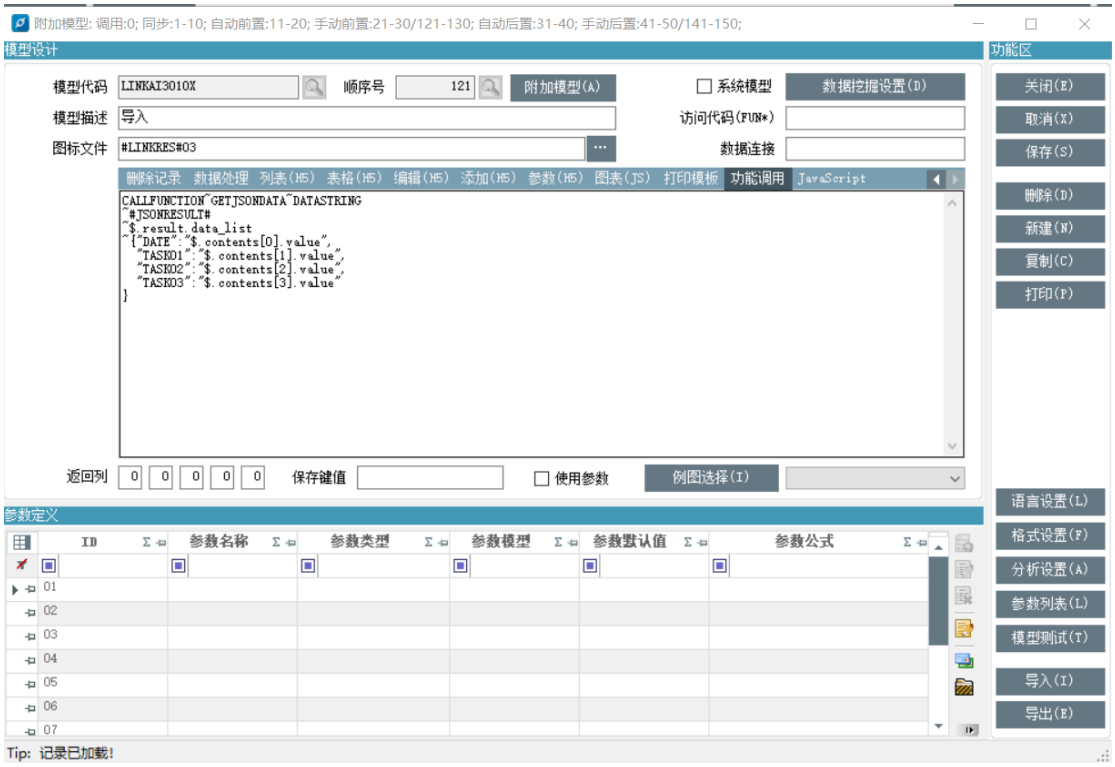


参考代码：

```
INSERT    INTO LINKTEMP (USR,COMPUTER,KEYWORDS,OPEDATE,A008)
VALUES (
```

```
        '#LoginUser#',
        '',
        'Dingtalk',
        GETDATE(),
        '#LINKAPI_RESULT#'
);
delete from linktemp where A008 not like '%content%' and KEYWORDS='Dingtalk'//清除空白日志信息
```

- 添加一个 121 模型，获取钉钉日志详细数据
- 功能调用代码，解析主模型查询的 JSONRESULT 字段



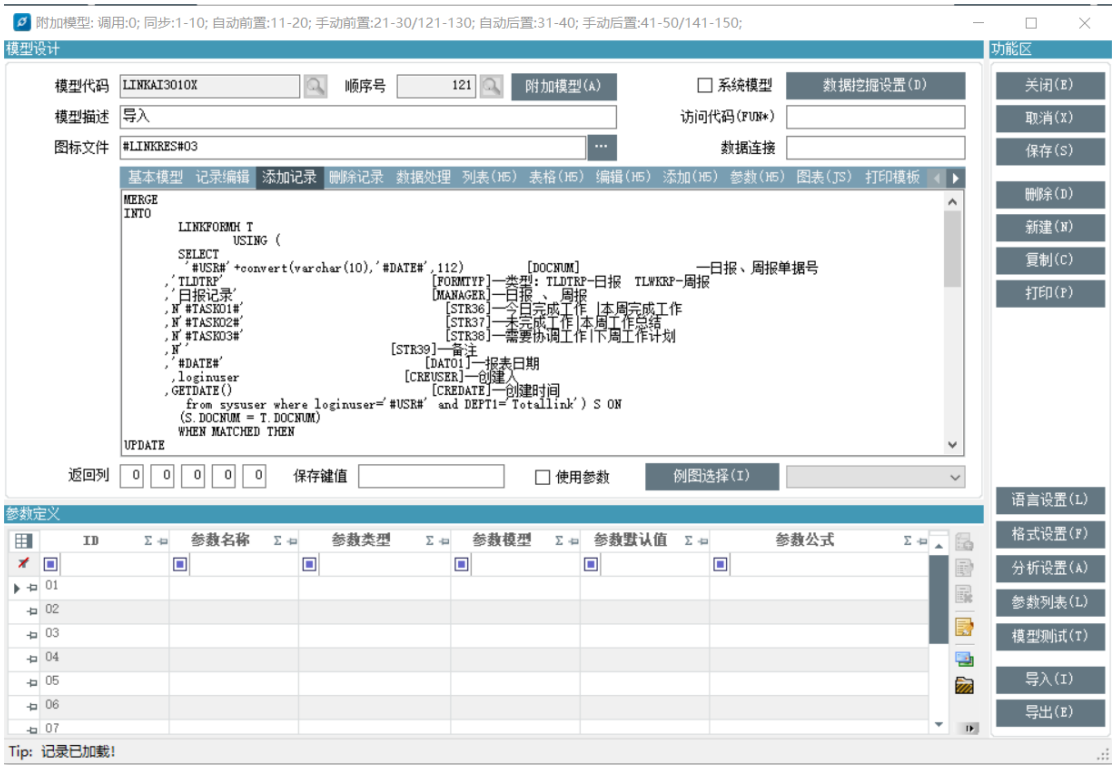
钉钉日志接口参考地址：https://ding-doc.dingtalk.com/doc#/serverapi2/yknhmg

参考代码：

```
CALLFUNCTION~GETJSONDATA~DATASTRING
~#JSONRESULT#
~$.result.data_list
~{"DATE":"$.contents[0].value",
   "TASK01":"$.contents[1].value",
   "TASK02":"$.contents[2].value",
   "TASK03":"$.contents[3].value"
```

```
}
```

- 将数据插入到钉钉日志表中：



参考代码：

```
MERGE
INTO
    LINKFORMH T
        USING (
    SELECT
     '#USR#'+convert(varchar(10),'#DATE#',112)         [DOCNUM]                    -
-日报、周报单据号
      ,'TLDTRP'                                  [FORMTYP]--类型：TLDTRP-日报  TLWKRP-
周报
      ,'日报记录'                                 [MANAGER]--日报 、 周报
      ,N'#TASK01#'                                  [STR36]--今日完成工作 |本周完成工作
      ,N'#TASK02#'                                  [STR37]--未完成工作|本周工作总结
      ,N'#TASK03#'                                  [STR38]--需要协调工作|下周工作计划
      ,N''                                 [STR39]--备注
      ,'#DATE#'                                [DAT01]--报表日期
       ,loginuser                            [CREUSER]--创建人
      ,GETDATE()                                [CREDATE]--创建时间
     from sysuser where loginuser='#USR#' and DEPT1='Totallink') S ON
    (S.DOCNUM = T.DOCNUM)
    WHEN MATCHED THEN
UPDATE
SET
       [STR36]=S.[STR36]                    --今日完成工作 |本周完成工作
      ,[STR37]=S.[STR37]                    --未完成工作|本周工作总结
      ,[STR38]=S.[STR38]                    --需要协调工作|下周工作计划
      ,[STR39]=S.[STR39]                    --备注
```

```
      ,[DAT01]=S.[DAT01]
    WHEN NOT MATCHED THEN
INSERT
    (  [LINKROWID]
      ,[DOCNUM]                 --日报、周报单据号
      ,[FORMTYP]                --类型：TLDTRP-日报  TLWKRP-周报
      ,[MANAGER]                --日报 、 周报
      ,[STR36]                  --今日完成工作 |本周完成工作
      ,[STR37]                  --未完成工作|本周工作总结
      ,[STR38]                  --需要协调工作|下周工作计划
      ,[STR39]                  --备注
      ,[DAT01]                  --报表日期
      ,[STR08]                  --报表状态
      ,[CREUSER]                --创建人
      ,[CREDATE]                --创建时间
       )
VALUES (
       NEWID()
      ,S.[DOCNUM]                 --日报、周报单据号
      ,S.[FORMTYP]                --类型：TLDTRP-日报  TLWKRP-周报
      ,S.[MANAGER]                --日报 、 周报
      ,S.[STR36]                  --今日完成工作 |本周完成工作
      ,S.[STR37]                  --未完成工作|本周工作总结
      ,S.[STR38]                  --需要协调工作|下周工作计划
      ,S.[STR39]                  --备注
      ,S.[DAT01]                  --报表日期
       ,'SUBMIT'                 --报表状态
      ,S.[CREUSER]                --创建人
      ,S.[CREDATE]                --创建时间

);
update LINKTEMP set A009='IMP' where ID=#ID#
```

3. 2. 2钉钉请假

钉钉请假信息可通过调用钉钉工作流接口获取

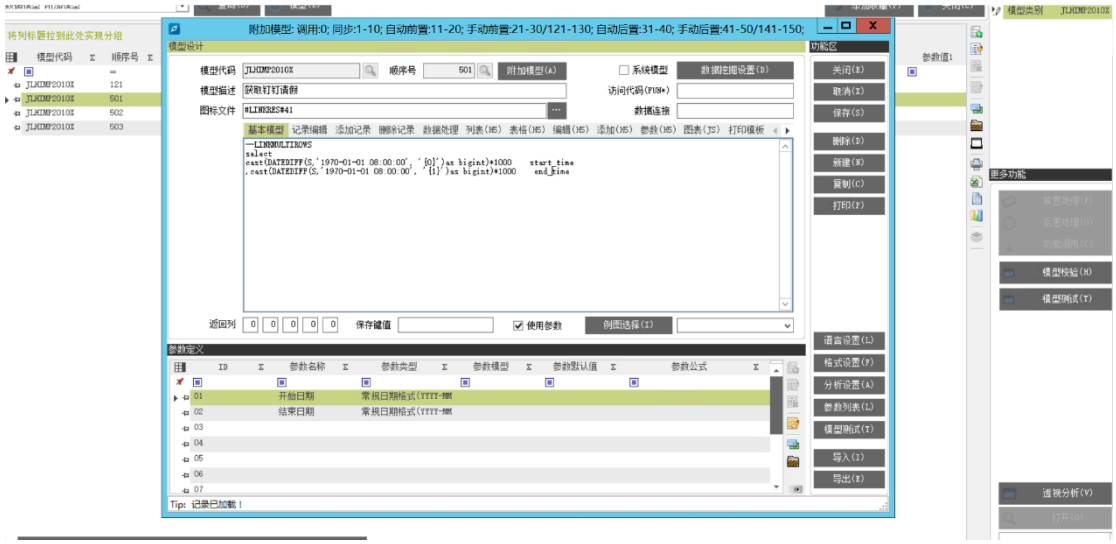
接口地址: https://ding-doc.dingtalk.com/doc#/serverapi2/hh8lx5

模型设计思路：

1. 获取请假列表 id 到中间表
2. 将请假列表解析成多行插入请假表中
3. 通过列表 ID 获取请假详情，插入到中间表中
4. 解析中间表中的请假详情到请假表中

获取请假列表信息到中间表

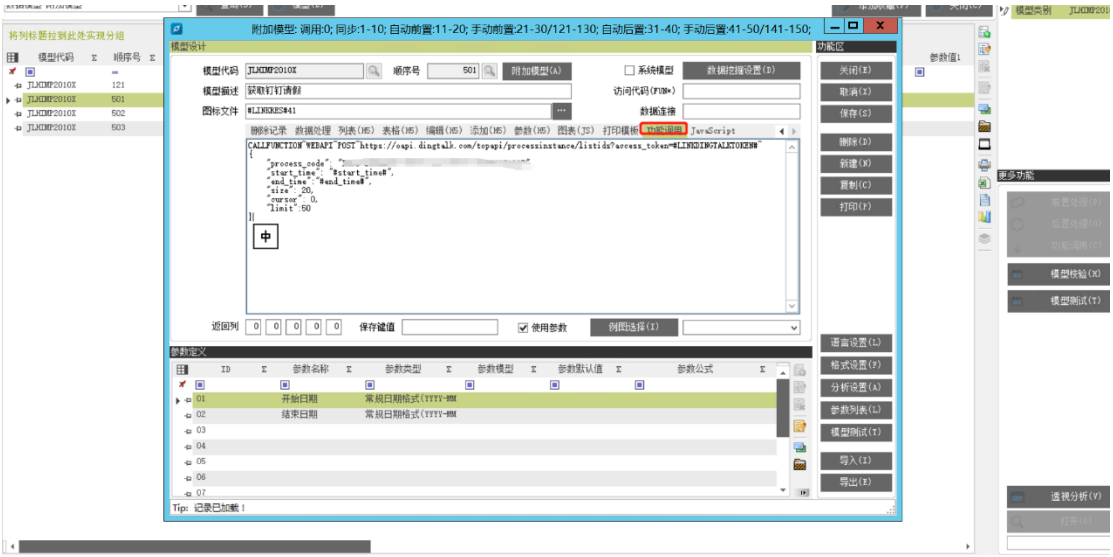
- 新增附加 501，“基本模型”模块将时间转换为 13 位时间戳



参考代码：

```
--LINKMULTIROWS
select
cast(DATEDIFF(S,'1970-01-01 08:00:00', '{0}')as bigint)*1000    start_time
,cast(DATEDIFF(S,'1970-01-01 08:00:00', '{1}')as bigint)*1000    end_time
```

- 在“功能调用”模块写入以下代码，获取请假审批列表详情



参考代码：

```
CALLFUNCTION~WEBAPI~POST~https://oapi.dingtalk.com/topapi/processinstance/listids
?access_token=#LINKDINGTALKTOKEN#~
{
    "process_code": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "start_time": "#start_time#",
    "end_time":"#end_time#",
    "size": 20,
    "cursor": 0,
    "limit":50
}
```

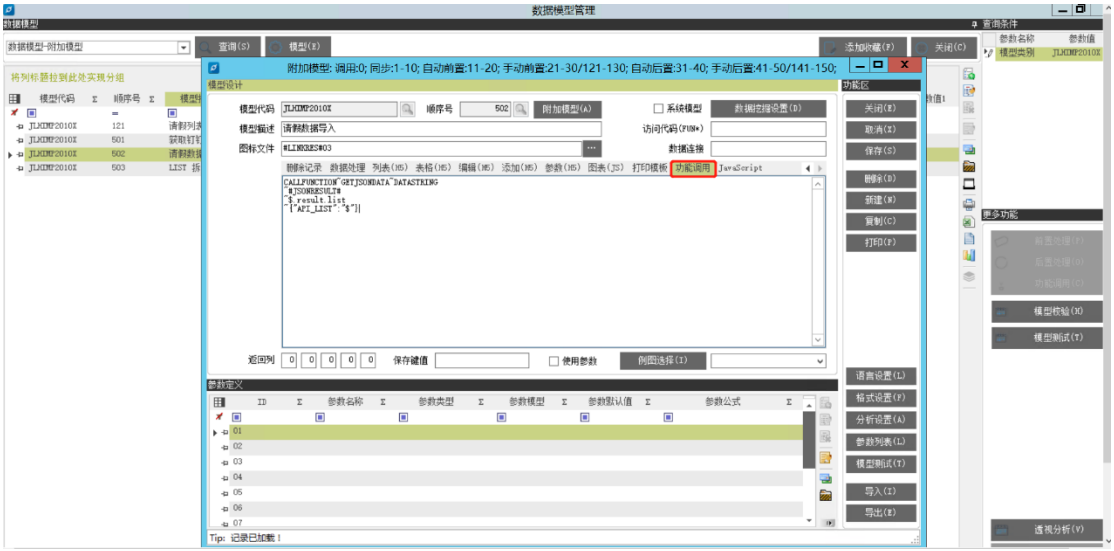
- “添加记录”模块将数据插入数据库表中

参考代码：

```
INSERT  INTO LINKTEMP (USR,COMPUTER,KEYWORDS,OPEDATE,A008)
VALUES (
    '',
    ' ',
    'Dingtalk-QJ',
    GETDATE(),
    '#LINKAPI_RESULT#'
);
```

将请假列表解析到多行插入请假表中

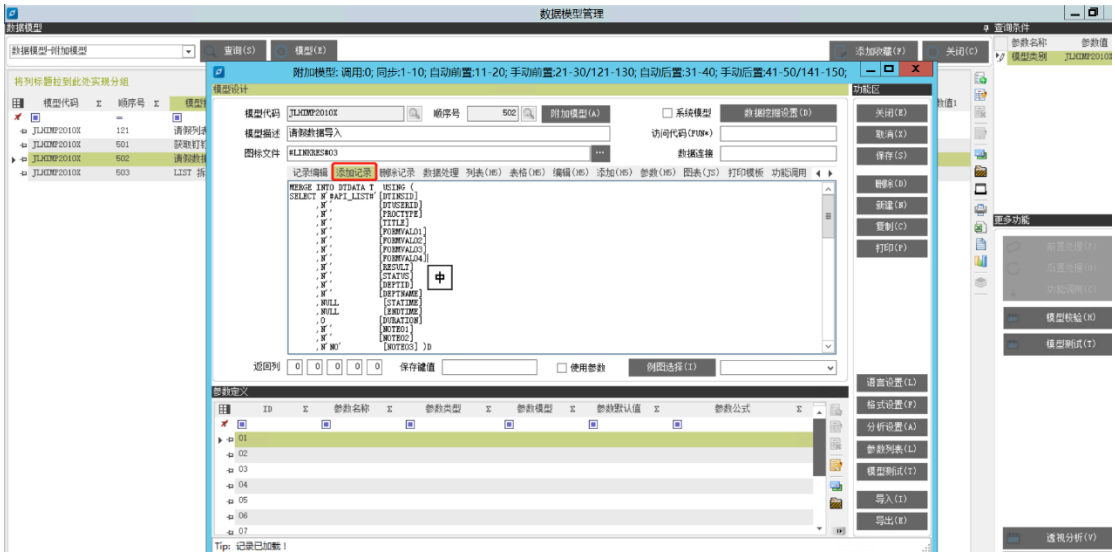
- 新建附加 502，解析请假审批列表，功能调用模块加入以下代码



参考代码：

```
CALLFUNCTION~GETJSONDATA~DATASTRING
~#JSONRESULT#
~$.result.list
~{"API_LIST":"$"}
```

- “添加记录”模块加入以下代码



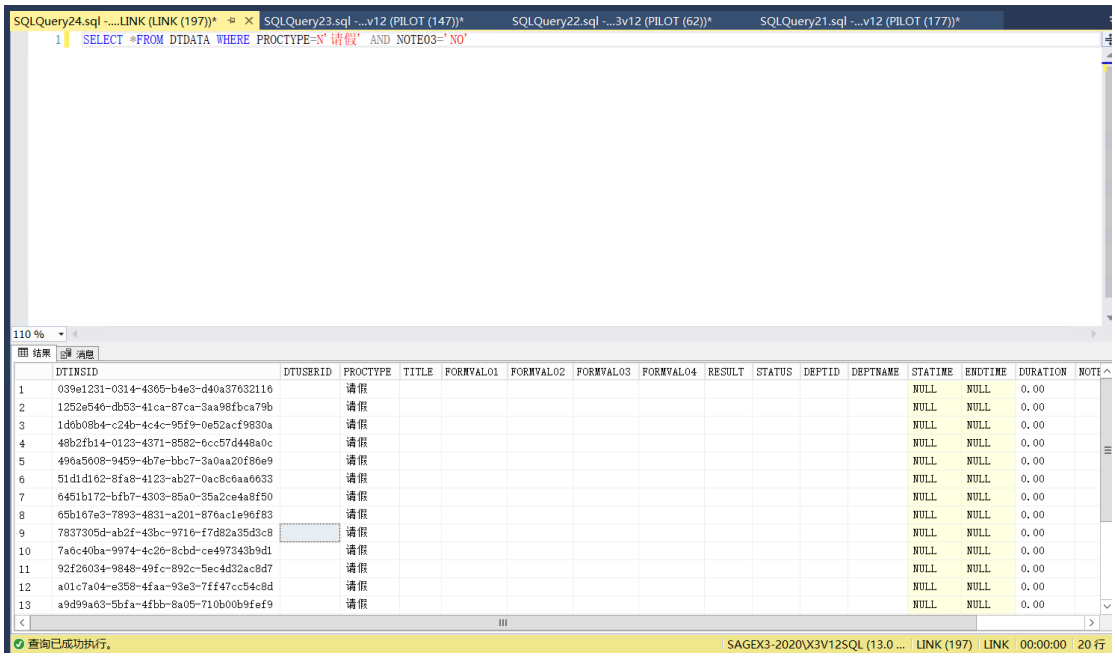

```
MERGE INTO
DTDATA T
  USING (

SELECT N'#API_LIST#'[DTINSID]
      ,N''           [DTUSERID]
      ,N''           [PROCTYPE]
      ,N''           [TITLE]
      ,N''           [FORMVAL01]
      ,N''           [FORMVAL02]
      ,N''           [FORMVAL03]
      ,N''           [FORMVAL04]
      ,N''           [RESULT]
      ,N''           [STATUS]
      ,N''           [DEPTID]
      ,N''           [DEPTNAME]
      ,NULL           [STATIME]
      ,NULL           [ENDTIME]
      ,0             [DURATION]
      ,N''           [NOTE01]
      ,N''           [NOTE02]
      ,N'NO'          [NOTE03]
 )D
ON
 (D.DTINSID = T.DTINSID)

 WHEN MATCHED THEN

UPDATE
SET
 [DTINSID]=D.[DTINSID]
WHEN NOT MATCHED THEN

INSERT
(
[DTINSID]
,[DTUSERID]
,[PROCTYPE]
,[TITLE]
,[FORMVAL01]
,[FORMVAL02]
,[FORMVAL03]
,[FORMVAL04]
,[RESULT]
```

```
,[STATUS]
,[DEPTID]
,[DEPTNAME]
,[STATIME]
,[ENDTIME]
,[DURATION]
,[NOTE01]
,[NOTE02]
,[NOTE03]
)

VALUES
(
D.[DTINSID]
,D.[DTUSERID]
,N'请假'
,D.[TITLE]
,D.[FORMVAL01]
,D.[FORMVAL02]
,D.[FORMVAL03]
,D.[FORMVAL04]
,D.[RESULT]
,D.[STATUS]
,D.[DEPTID]
,D.[DEPTNAME]
,D.[STATIME]
,D.[ENDTIME]
,D.[DURATION]
,D.[NOTE01]
,D.[NOTE02]
,D.[NOTE03]
);
UPDATE LINKTEMP  SET A009='IMP' WHERE KEYWORDS='Dingtalk-QJ' AND ID='#ID#'
```

执行结果：

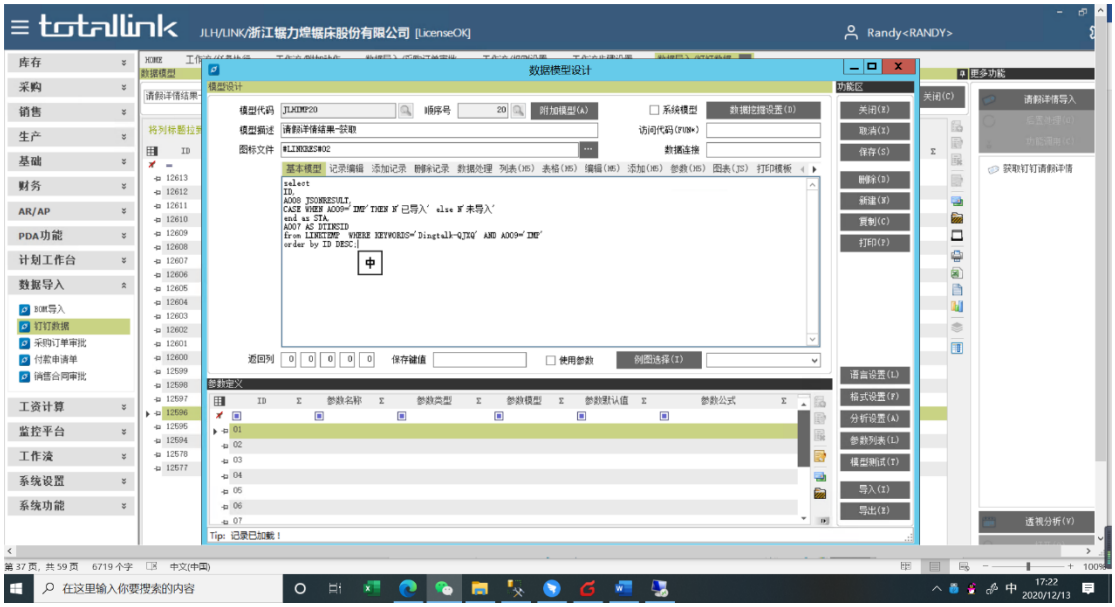
- 如下图，点击 501 按钮，获得多个请假审批 ID

- 点击 502 按钮，将列表 ID 解析并拆分到数据库表中



通过列表 ID 获取请假详情

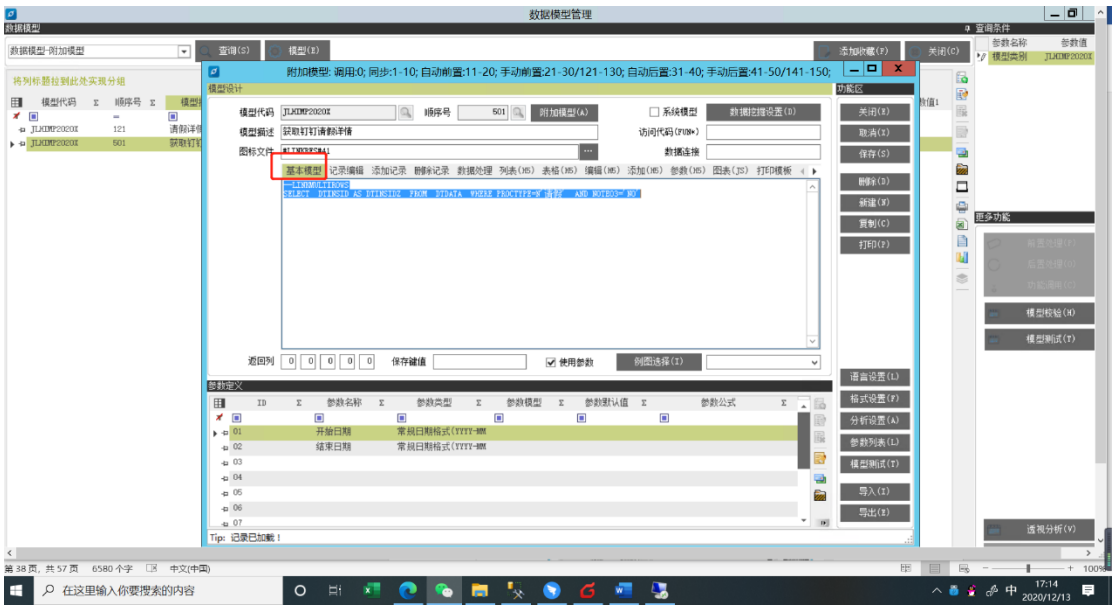
- 新建基本模型



参考代码：

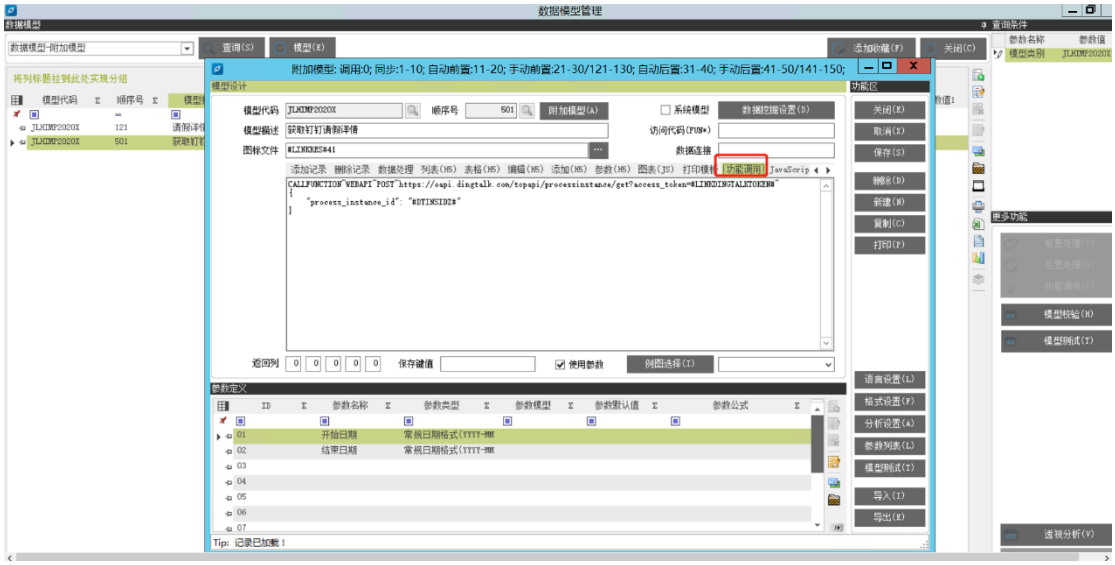
```
select
ID,
A008 JSONRESULT,
CASE WHEN A009='IMP'THEN N'已导入' else N'未导入'
end as STA,
A007 AS DTINSID
from LINKTEMP  WHERE KEYWORDS='Dingtalk-QJXQ'
order by ID DESC;
```

- 在基本模型背后添加附加 501 模型，传递请假列表 ID 获取请假详情
- 在附加 501 基本模型查出待处理的请假审批 ID




```
--LINKMULTIROWS
SELECT  DTINSID AS DTINSIDZ  FROM  DTDATA  WHERE PROCTYPE=N'请假'  AND NOTE03='NO'
```

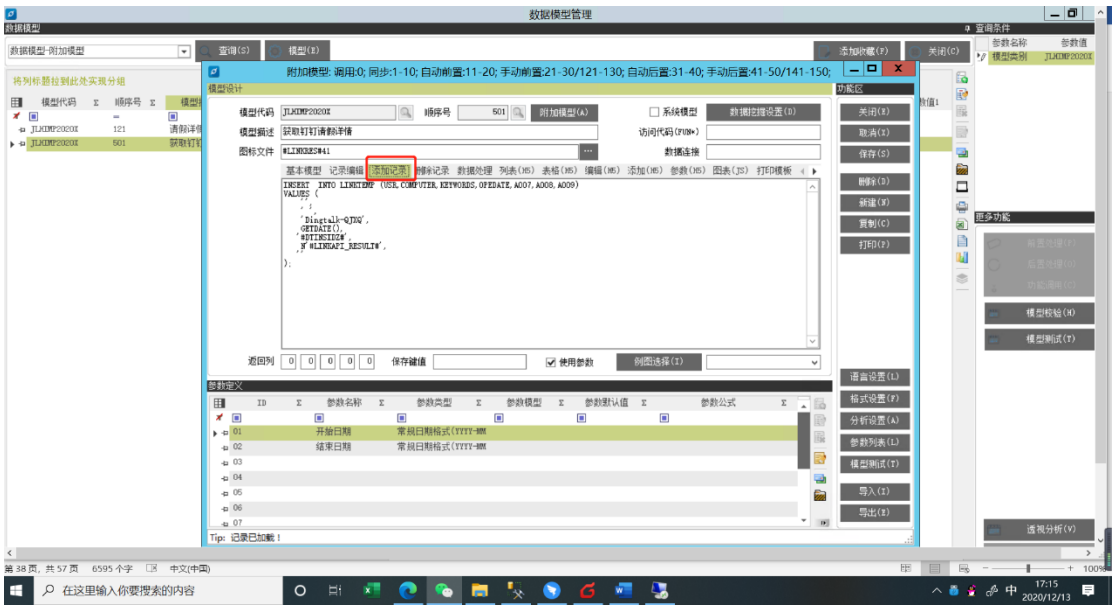
- 解析请假列表 ID



参考代码：

```
CALLFUNCTION~WEBAPI~POST~https://oapi.dingtalk.com/topapi/processinstance/get?acc
ess_token=#LINKDINGTALKTOKEN#~
{
    "process_instance_id": "#DTINSIDZ#"
}
```

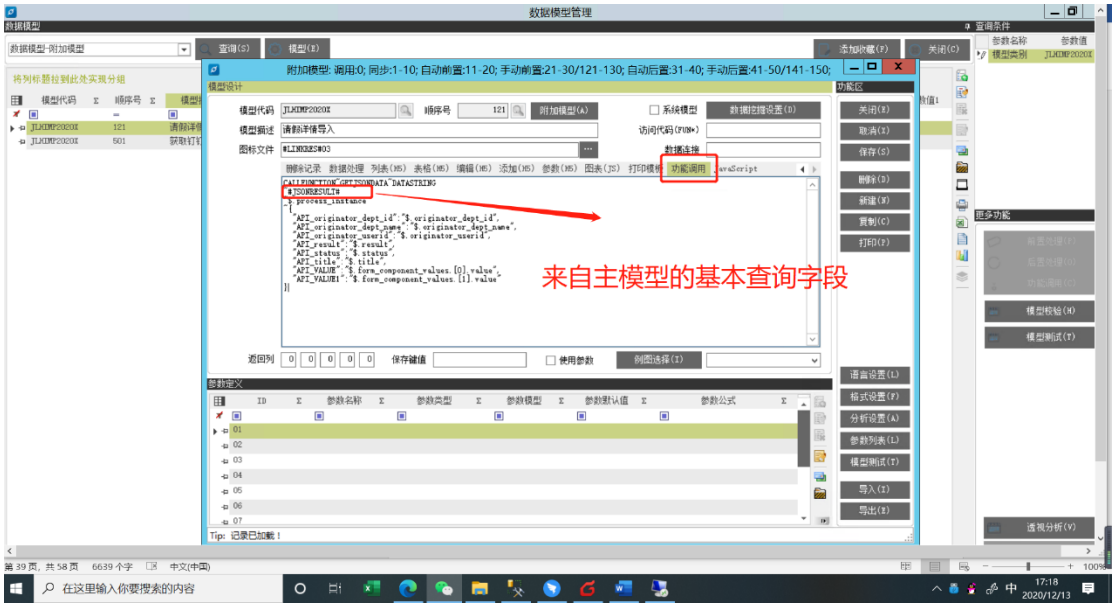
- 将请假详情插入到中间表中



```
INSERT  INTO LINKTEMP (USR,COMPUTER,KEYWORDS,OPEDATE,A007,A008,A009)
VALUES (
    '',
    ' ',
    'Dingtalk-QJXQ',
    GETDATE(),
   '#DTINSIDZ#',
    N'#LINKAPI_RESULT#',
   ''
);
```

解析中间表中的请假详情到请假表中

- 新建附加 121，解析中间表中的请假详情到最终表中的对应字段中

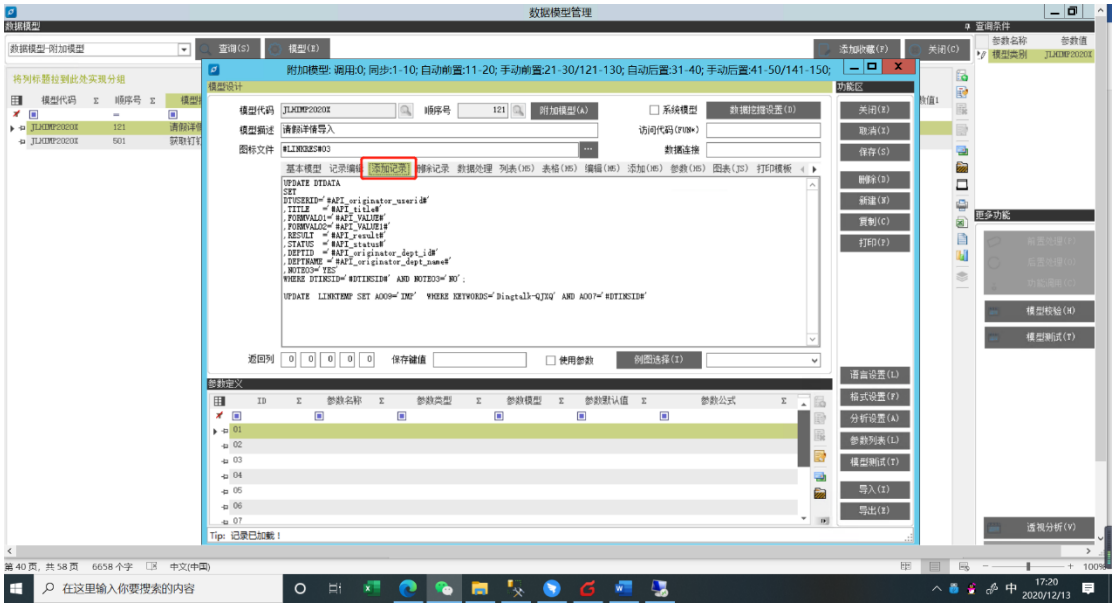


参考代码：

```
CALLFUNCTION~GETJSONDATA~DATASTRING
~#JSONRESULT#
~$.process_instance
~{
  "API_originator_dept_id":"$.originator_dept_id",
  "API_originator_dept_name":"$.originator_dept_name",
```

```
  "API_originator_userid":"$.originator_userid",
  "API_result":"$.result",
  "API_status":"$.status",
  "API_title":"$.title",
  "API_VALUE":"$.form_component_values.[0].value",
  "API_VALUE1":"$.form_component_values.[1].value"
}
```

- 附加 121“添加记录”模块将解析的数据解析到请假表对应字段

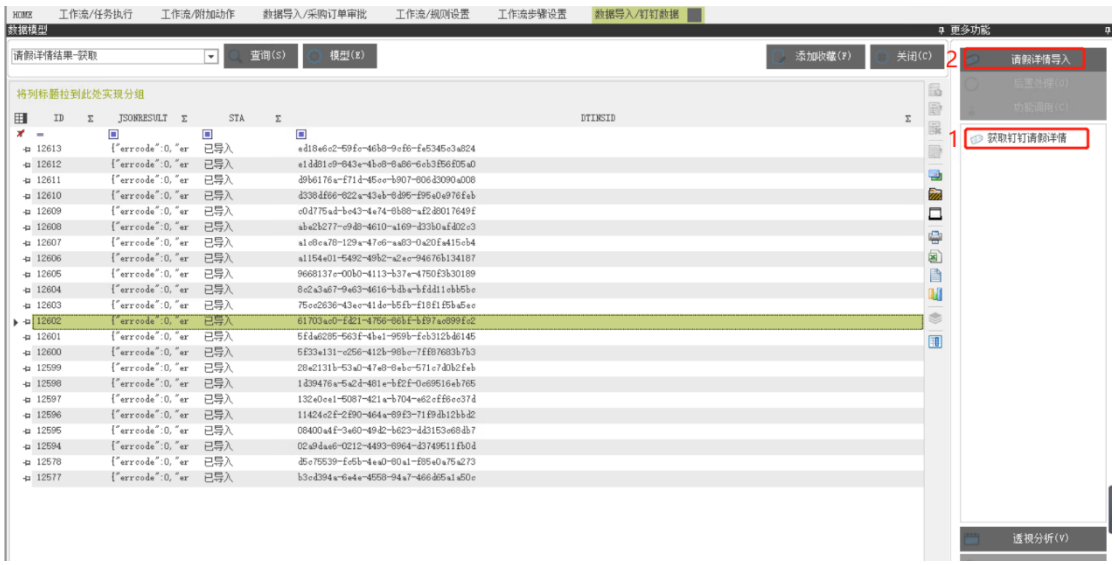


参考代码:

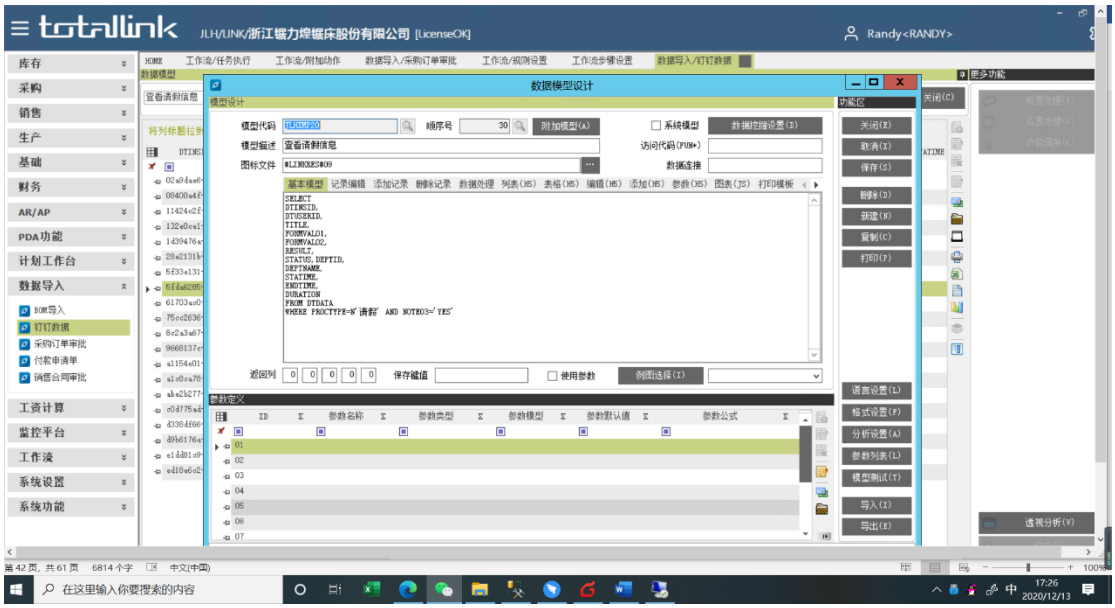
```
UPDATE DTDATA
SET
DTUSERID='#API_originator_userid#'
,TITLE    ='#API_title#'
,FORMVAL01='#API_VALUE#'
,FORMVAL02='#API_VALUE1#'
,RESULT   ='#API_result#'
,STATUS   ='#API_status#'
,DEPTID   ='#API_originator_dept_id#'
,DEPTNAME ='#API_originator_dept_name#'
,NOTE03='YES'
WHERE DTINSID='#DTINSID#' AND NOTE03='NO';

UPDATE      LINKTEMP   SET   A009='IMP'      WHERE   KEYWORDS='Dingtalk-QJXQ'   AND
A007='#DTINSID#'
```

执行结果：

- 点击 501 按钮，传递列表 ID 到中间表，执行 121，获取界面上的详情并解析到最终表



- 新建模型，查看请假表中请假的信息



参考代码：

```
SELECT
DTINSID,
DTUSERID,
TITLE,
FORMVAL01,
FORMVAL02,
RESULT,
STATUS,DEPTID,
DEPTNAME,
STATIME,
ENDTIME,
DURATION
FROM DTDATA
WHERE PROCTYPE=N'请假' AND NOTE03='YES'
```

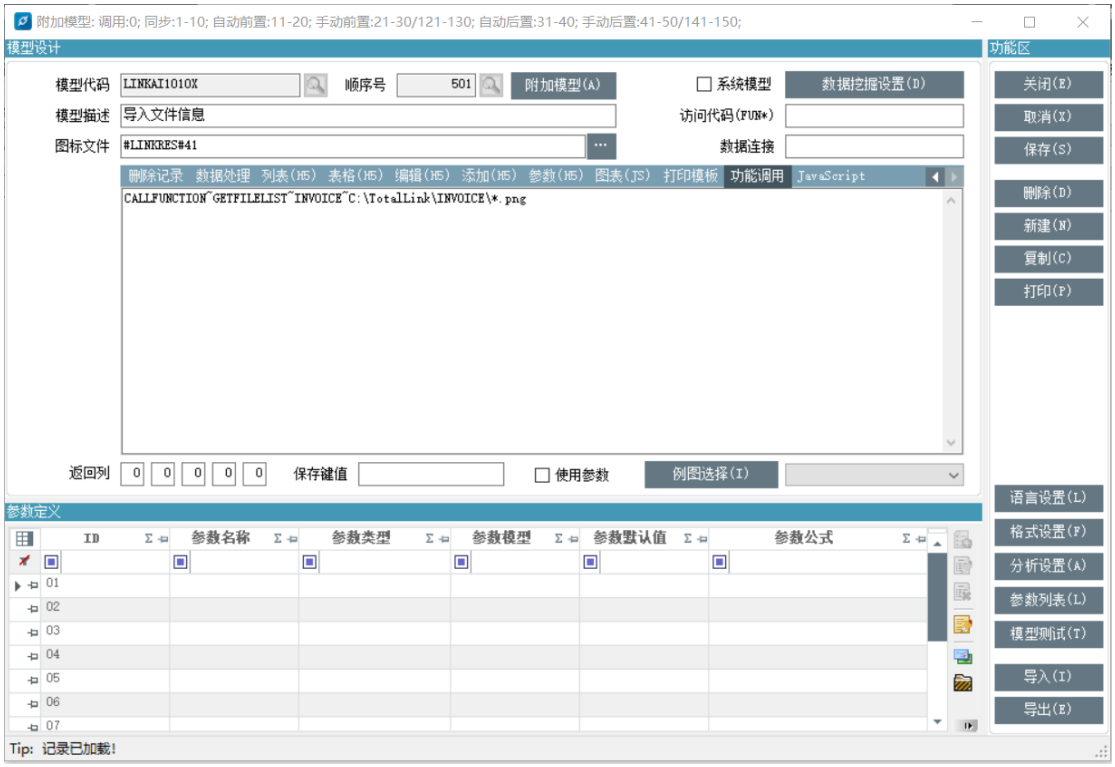
3.3 发票、火车票识别

TotalLINK 可以识别并获取发票、火车票等信息，下面介绍 TotalLINK 获取发票信息的实例，火车票方法以此类推。

3. 3. 1客户端应用

客户端发票识别的应用场景一般是用户把发票放到固定目录文件夹下，系统开启自动任务识别固定目录下的发票图片并导入系统表中

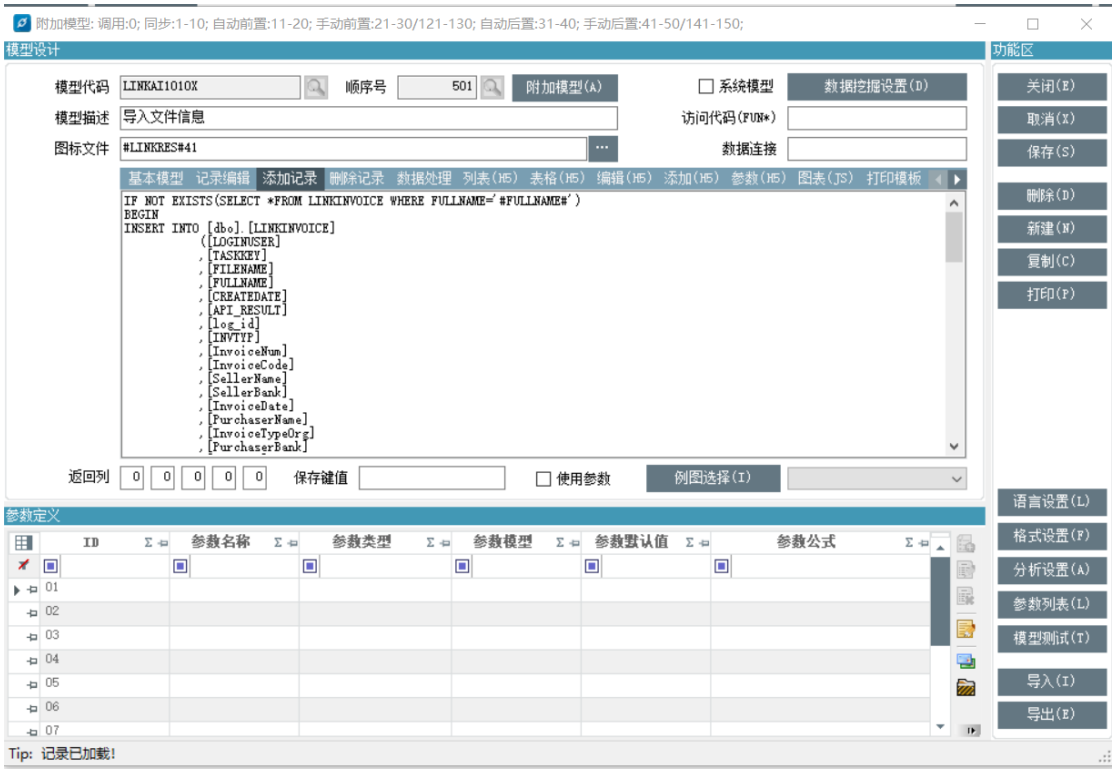
- 新建 501 附加模型，功能调用模块写入获取固定目录下所有发票图片代码



参考代码：

```
CALLFUNCTION~GETFILELIST~INVOICE~C:\TotalLink\INVOICE\*.png
```

- 在 501 添加记录模块把发票图片信息插入到发票表中

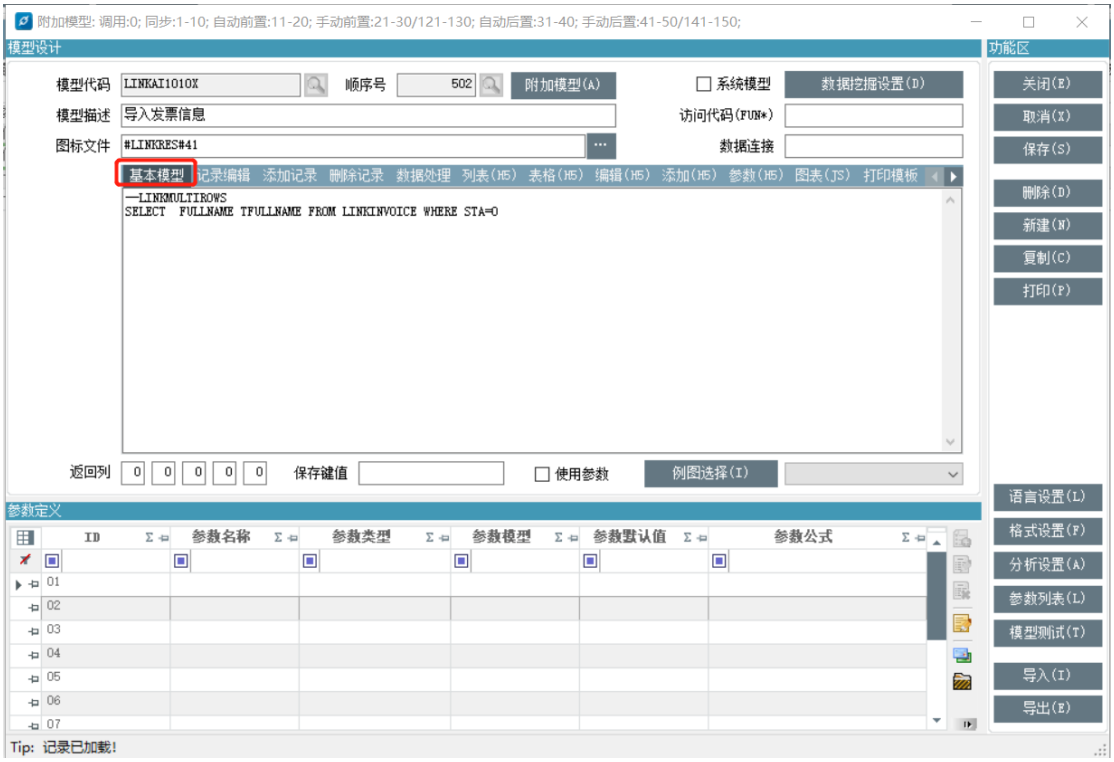


参考代码:

```
IF NOT EXISTS(SELECT *FROM LINKINVOICE WHERE FULLNAME='#FULLNAME#')
BEGIN
INSERT INTO [dbo].[LINKINVOICE]
           ([LOGINUSER]
           ,[TASKKEY]
           ,[FILENAME]
           ,[FULLNAME]
           ,[CREATEDATE]
           ,[API_RESULT]
           ,[log_id]
           ,[INVTYP]
           ,[InvoiceNum]
           ,[InvoiceCode]
           ,[SellerName]
           ,[SellerBank]
           ,[InvoiceDate]
           ,[PurchaserName]
           ,[InvoiceTypeOrg]
           ,[PurchaserBank]
           ,[Remarks]
           ,[SellerAddress]
           ,[PurchaserAddress]
           ,[Payee]
           ,[PurchaserRegiste]
           ,[FLAG01]
           ,[FLAG02]
           ,[FLAG03]
           ,[FLAG04]
           ,[FLAG05]
           ,[NOTE01]
           ,[NOTE02]
           ,[NOTE03]
           ,[NOTE04]
           ,[NOTE05]
           ,[TotalAmount]
```

```
            ,[TotalTax]
            ,[AmountInFiguers]
            ,[CheckCode]
            ,[PurchaserRegisterNum]
            ,[SellerRegisterNum]
            ,[Checker]
            ,[NoteDrawer]
            ,[STA])
      VALUES
            ('#LOGIN_USER#'
            ,'INVOICE'
            ,'#FILENAME#'
            ,'#FULLNAME#'
            ,getdate()
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,0
            ,0
            ,0
            ,0
            ,0
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,0
            ,0
            ,0
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,''
            ,0)
          END
```

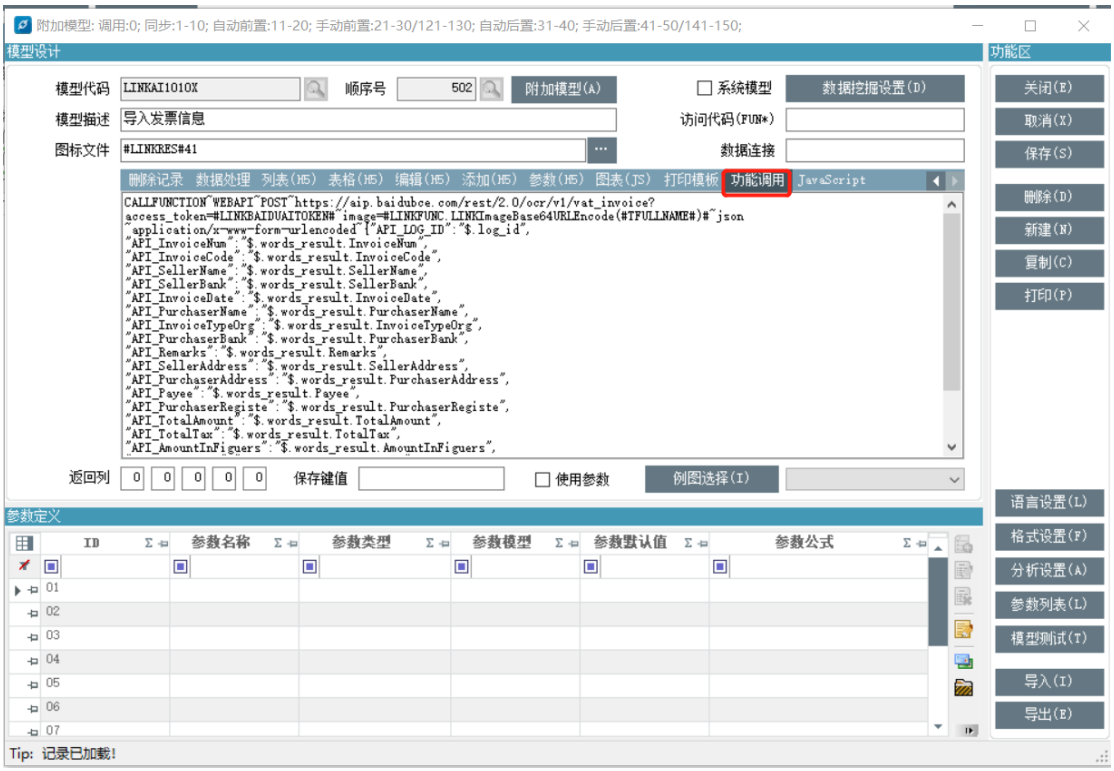
- 新建附加 502，利用附加模型处理多行数据方式获取发票图片详细信息，识别发票详细信息并导入到发票表中



参考代码：

```
--LINKMULTIROWS
SELECT  FULLNAME TFULLNAME FROM LINKINVOICE WHERE STA=0
```

- 功能调用模块



参考代码：

```
CALLFUNCTION~WEBAPI~POST~https://aip.baidubce.com/rest/2.0/ocr/v1/vat_invoice?acc
ess_token=#LINKBAIDUAITOKEN#~image=#LINKFUNC.LINKImageBase64URLEncode(#TFULLNAME#
```

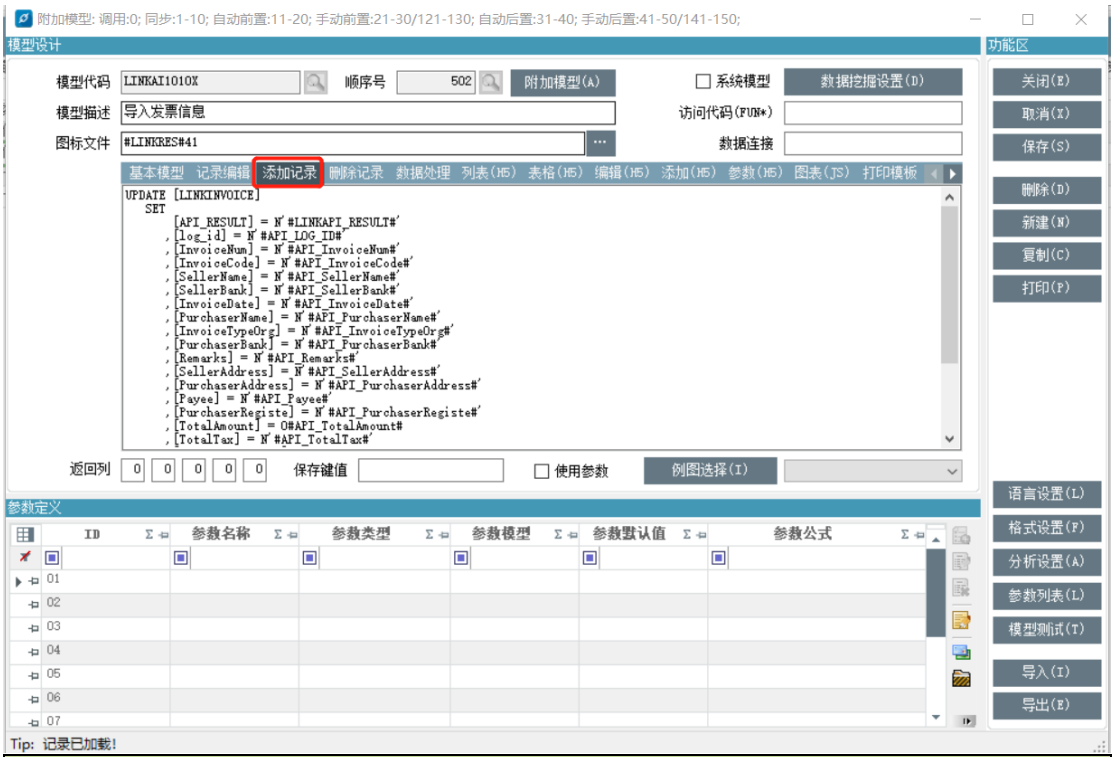
```
)#~json
~application/x-www-form-urlencoded~{"API_LOG_ID":"$.log_id",
"API_InvoiceNum":"$.words_result.InvoiceNum",
"API_InvoiceCode":"$.words_result.InvoiceCode",
"API_SellerName":"$.words_result.SellerName",
"API_SellerBank":"$.words_result.SellerBank",
"API_InvoiceDate":"$.words_result.InvoiceDate",
"API_PurchaserName":"$.words_result.PurchaserName",
"API_InvoiceTypeOrg":"$.words_result.InvoiceTypeOrg",
"API_PurchaserBank":"$.words_result.PurchaserBank",
"API_Remarks":"$.words_result.Remarks",
"API_SellerAddress":"$.words_result.SellerAddress",
"API_PurchaserAddress":"$.words_result.PurchaserAddress",
"API_Payee":"$.words_result.Payee",
"API_PurchaserRegiste":"$.words_result.PurchaserRegiste",
"API_TotalAmount":"$.words_result.TotalAmount",
"API_TotalTax":"$.words_result.TotalTax",
"API_AmountInFiguers":"$.words_result.AmountInFiguers",
"API_CheckCode":"$.words_result.CheckCode",
"API_PurchaserRegisterNum":"$.words_result.PurchaserRegisterNum",
"API_SellerRegisterNum":"$.words_result.SellerRegisterNum",
"API_Checker":"$.words_result.Checker",
"API_NoteDrawer":"$.words_result.NoteDrawer"}
```

代码解析：

"API_InvoiceNum":"$.words_result.InvoiceNum",

$.words_result.InvoiceNum 表示识别 InvoiceNum 字段的数据；API_InvoiceNum 表示自定义的变量。

- 添加记录模块代码



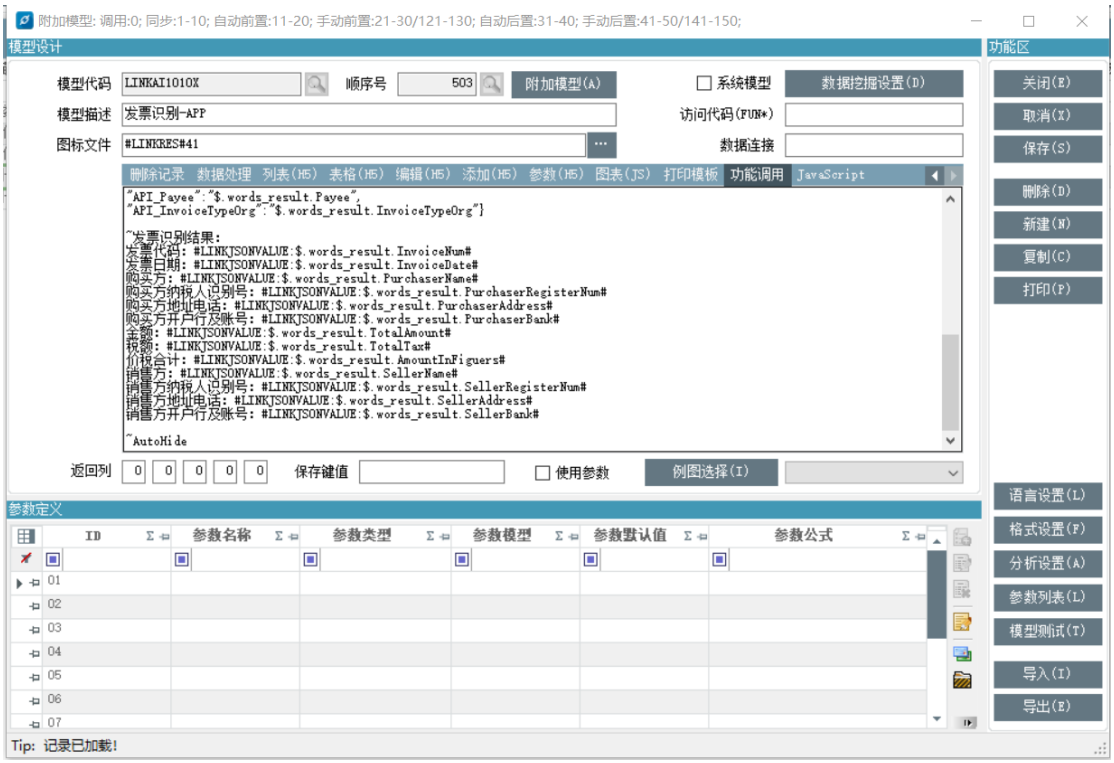

```
UPDATE [LINKINVOICE]
  SET
      [API_RESULT] = N'#LINKAPI_RESULT#'
     ,[log_id] = N'#API_LOG_ID#'
     ,[InvoiceNum] = N'#API_InvoiceNum#'
```

```
      ,[InvoiceCode] = N'#API_InvoiceCode#'
      ,[SellerName] = N'#API_SellerName#'
      ,[SellerBank] = N'#API_SellerBank#'
      ,[InvoiceDate] = N'#API_InvoiceDate#'
      ,[PurchaserName] = N'#API_PurchaserName#'
      ,[InvoiceTypeOrg] = N'#API_InvoiceTypeOrg#'
      ,[PurchaserBank] = N'#API_PurchaserBank#'
      ,[Remarks] = N'#API_Remarks#'
      ,[SellerAddress] = N'#API_SellerAddress#'
      ,[PurchaserAddress] = N'#API_PurchaserAddress#'
      ,[Payee] = N'#API_Payee#'
      ,[PurchaserRegiste] = N'#API_PurchaserRegiste#'
      ,[TotalAmount] = 0#API_TotalAmount#
      ,[TotalTax] = N'#API_TotalTax#'
      ,[AmountInFiguers] = 0#API_AmountInFiguers#
      ,[CheckCode] = N'#API_CheckCode#'
      ,[PurchaserRegisterNum] = N'#API_PurchaserRegisterNum#'
      ,[SellerRegisterNum] = N'#API_SellerRegisterNum#'
      ,[Checker] = N'#API_Checker#'
      ,[NoteDrawer] = N'#API_NoteDrawer#'
      ,[STA]=1
 WHERE LOGINUSER='#LOGIN_USER#'  AND TASKKEY='INVOICE'  AND FULLNAME='#TFULLNAME#'
```

3. 3. 2APP 应用

在 APP 选择发票图片并返回发票详细信息：

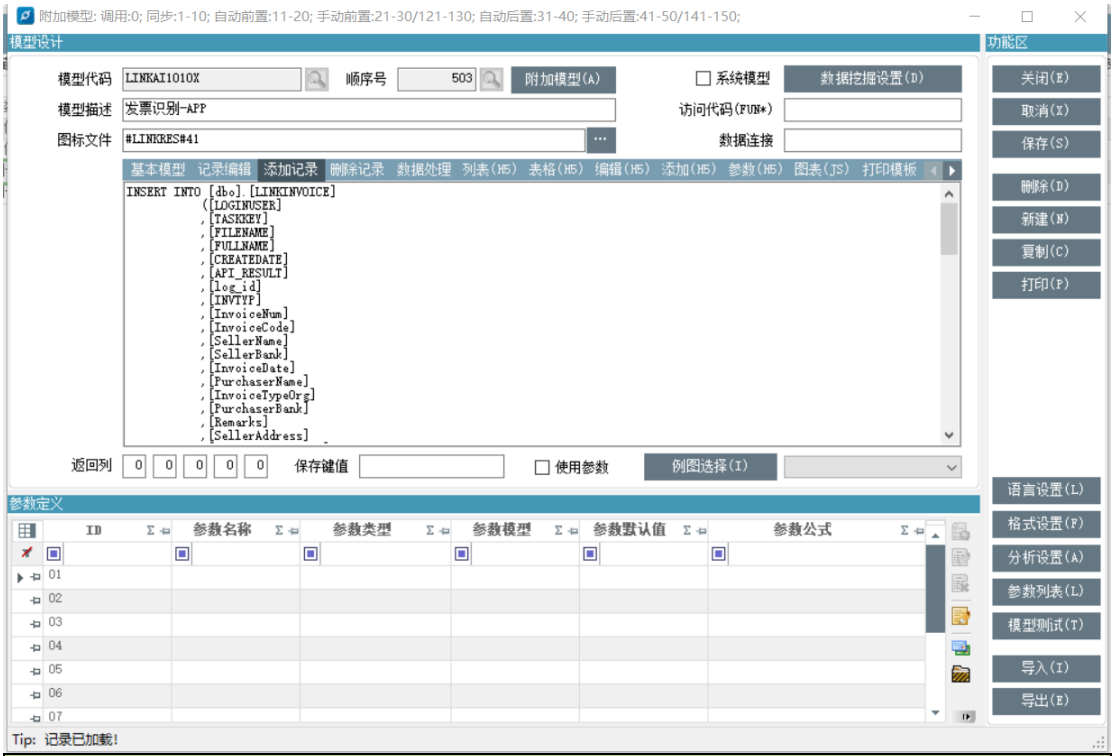
- 增加 503 附加模型，在功能调用加入以下代码，识别比发票信息




```
 CALLFUNCTION~WEBAPI~POST
~https://aip.baidubce.com/rest/2.0/ocr/v1/vat_invoice?access_token=#LINKBAIDUAITO
KEN#
~image=#LINKIMAGEBASE64URLENCODE#
~json
~application/x-www-form-urlencoded
~{"API_log_id":"$.log_id",
"API_InvoiceNum":"$.words_result.InvoiceNum",
"API_InvoiceCode":"$.words_result.InvoiceCode",
"API_InvoiceDate":"$.words_result.InvoiceDate",
"API_SellerName":"$.words_result.SellerName",
"API_SellerRegisterNum":"$.words_result.SellerRegisterNum",
"API_SellerAddress":"$.words_result.SellerAddress",
"API_SellerBank":"$.words_result.SellerBank",
"API_PurchaserName":"$.words_result.PurchaserName",
"API_PurchaserRegisterNum":"$.words_result.PurchaserRegisterNum",
"API_PurchaserAddress":"$.words_result.PurchaserAddress",
"API_PurchaserBank":"$.words_result.PurchaserBank",
"API_TotalAmount":"$.words_result.TotalAmount",
"API_TotalTax":"$.words_result.TotalTax",
"API_AmountInFiguers":"$.words_result.AmountInFiguers",
"API_PurchaserRegiste":"$.words_result.PurchaserRegiste",
"API_CheckCode":"$.words_result.CheckCode",
"API_Checker":"$.words_result.Checker",
"API_NoteDrawer":"$.words_result.NoteDrawer",
```

```
"API_Remarks":"$.words_result.Remarks",
"API_Payee":"$.words_result.Payee",
"API_InvoiceTypeOrg":"$.words_result.InvoiceTypeOrg"}

~发票识别结果：
发票代码：#LINKJSONVALUE:$.words_result.InvoiceNum#
发票日期：#LINKJSONVALUE:$.words_result.InvoiceDate#
购买方：#LINKJSONVALUE:$.words_result.PurchaserName#
购买方纳税人识别号：#LINKJSONVALUE:$.words_result.PurchaserRegisterNum#
购买方地址电话：#LINKJSONVALUE:$.words_result.PurchaserAddress#
购买方开户行及账号：#LINKJSONVALUE:$.words_result.PurchaserBank#
金额：#LINKJSONVALUE:$.words_result.TotalAmount#
税额：#LINKJSONVALUE:$.words_result.TotalTax#
价税合计：#LINKJSONVALUE:$.words_result.AmountInFiguers#
销售方：#LINKJSONVALUE:$.words_result.SellerName#
销售方纳税人识别号：#LINKJSONVALUE:$.words_result.SellerRegisterNum#
销售方地址电话：#LINKJSONVALUE:$.words_result.SellerAddress#
销售方开户行及账号：#LINKJSONVALUE:$.words_result.SellerBank#

~AutoHide
```

- 添加记录模块加入以下代码，将发票信息插入发票表




```
INSERT INTO [dbo].[LINKINVOICE]
           ([LOGINUSER]
           ,[TASKKEY]
           ,[FILENAME]
           ,[FULLNAME]
```

```
            ,[CREATEDATE]
            ,[API_RESULT]
            ,[log_id]
            ,[INVTYP]
            ,[InvoiceNum]
            ,[InvoiceCode]
            ,[SellerName]
            ,[SellerBank]
            ,[InvoiceDate]
            ,[PurchaserName]
            ,[InvoiceTypeOrg]
            ,[PurchaserBank]
            ,[Remarks]
            ,[SellerAddress]
            ,[PurchaserAddress]
            ,[Payee]
            ,[PurchaserRegiste]
            ,[FLAG01]
            ,[FLAG02]
            ,[FLAG03]
            ,[FLAG04]
            ,[FLAG05]
            ,[NOTE01]
            ,[NOTE02]
            ,[NOTE03]
            ,[NOTE04]
            ,[NOTE05]
            ,[TotalAmount]
            ,[TotalTax]
            ,[AmountInFiguers]
            ,[CheckCode]
            ,[PurchaserRegisterNum]
            ,[SellerRegisterNum]
            ,[Checker]
            ,[NoteDrawer]
            ,[STA])
      VALUES
            ('#LOGIN_USER#'
            ,'INVOICE'
            ,N'#API_InvoiceNum#'
            ,N'#API_InvoiceCode#'
            ,getdate()
            ,N'#LINKAPI_RESULT#'
            ,N'#API_log_id#'
```

```
            ,N'APP-INVOICE'
            ,N'#API_InvoiceNum#'
            ,N'#API_InvoiceCode#'
            ,N'#API_SellerName#'
            ,N'#API_SellerBank#'
            ,N'#API_InvoiceDate#'
            ,N'#API_PurchaserName#'
            ,N'#API_InvoiceTypeOrg#'
            ,N'#API_PurchaserBank#'
            ,N'#API_Remarks#'
            ,N'#API_SellerAddress#'
            ,N'#API_PurchaserAddress#'
            ,N'#API_Payee#'
            ,N'#API_PurchaserRegiste#'
            ,0
            ,0
            ,0
            ,0
            ,0
            ,''
            ,''
            ,''
            ,''
            ,''
            ,0#API_TotalAmount#
            ,N'#API_TotalTax#'
            ,0#API_AmountInFiguers#
            ,N'#API_CheckCode#'
            ,N'#API_PurchaserRegisterNum#'
            ,N'#API_SellerRegisterNum#'
            ,N'#API_Checker#'
            ,N'#API_NoteDrawer#'
            ,2)
```

执行结果：

3. 3. 3火车票识别

火车票识别与发票识别用法一致，火车票识别的功能调用代码如下:

```
CALLFUNCTION~WEBAPI~POST
~https://aip.baidubce.com/rest/2.0/ocr/v1/train_ticket?access_token=#LINKBAIDUAIT
OKEN#
~image=#LINKIMAGEBASE64URLENCODE#
~json
~application/x-www-form-urlencoded
~{
"API_LOG_ID":"$.log_id",
"API_starting_station":"$.words_result.starting_station",
"API_ticket_num":"$.words_result.ticket_num",
"API_train_num":"$.words_result.train_num",
"API_destination_station":"$.words_result.destination_station",
"API_date":"$.words_result.date",
"API_ticket_rates":"$.words_result.ticket_rates",
"API_seat_category":"$.words_result.seat_category",
"API_name":"$.words_result.name",
```

```
"API_id_num":"$.words_result.id_num",
"API_serial_number":"$.words_result.serial_number",
"API_sales_station":"$.words_result.sales_station",
"API_time":"$.words_result.time",
"API_seat_num":"$.words_result.seat_num"
}

~火车票识别结果：
车票号：#LINKJSONVALUE:$.words_result.ticket_num#
始发站：#LINKJSONVALUE:$.words_result.starting_station#
车次号：#LINKJSONVALUE:$.words_result.train_num#
到达站：#LINKJSONVALUE:$.words_result.destination_station#
出发日期：#LINKJSONVALUE:$.words_result.date#
车票金额：#LINKJSONVALUE:$.words_result.ticket_rates#
席别：#LINKJSONVALUE:$.words_result.seat_category#
乘客姓名：#LINKJSONVALUE:$.words_result.name#
身份证号：#LINKJSONVALUE:$.words_result.id_num#
序列号：#LINKJSONVALUE:$.words_result.serial_number#
售站：#LINKJSONVALUE:$.words_result.sales_station#
时间：#LINKJSONVALUE:$.words_result.time#
座位号：#LINKJSONVALUE:$.words_result.seat_num#

~AutoHide
```

- 添加模块代码：

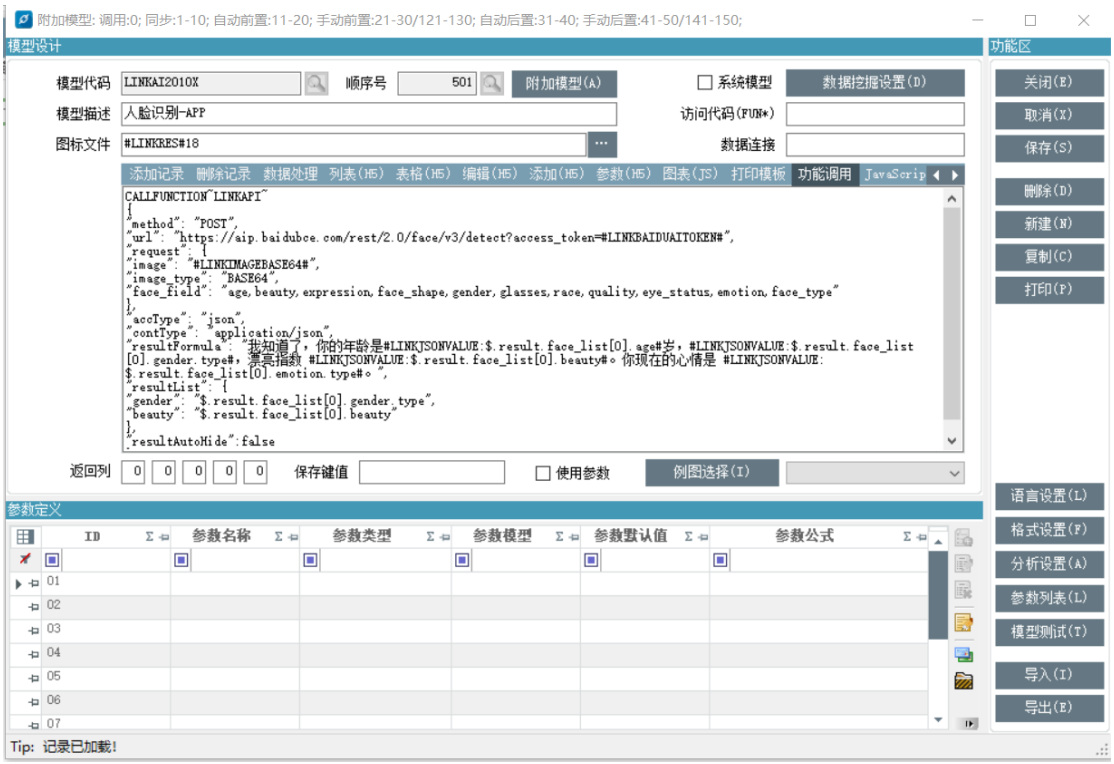
```
 IF NOT EXISTS(SELECT *FROM LINKTRAINTICKET WHERE FULLNAME='#FULLNAME#')
BEGIN
INSERT INTO [dbo].[LINKTRAINTICKET]
           ([LOGINUSER]
           ,[TASKKEY]
           ,[FILENAME]
           ,[FULLNAME]
           ,[CREATEDATE]
           ,[API_RESULT]
           ,[log_id]
           ,[INVTYP]
           ,[ticket_num]
           ,[starting_station]
           ,[train_num]
           ,[destination_station]
           ,[date]
           ,[ticket_rates]
           ,[seat_category]
           ,[name]
           ,[id_num]
```

```
            ,[serial_number]
            ,[sales_station]
            ,[time]
            ,[seat_num]
            ,[STA])
      VALUES
            ('#LOGIN_USER#'
            ,'TRAIN_TICKET'
            ,'#FILENAME#'
            ,'#FULLNAME#'
            ,getdate()
            ,N'#LINKAPI_RESULT#'
            ,N'#API_LOG_ID#'
            ,'APP-TRAIN-TICKET'
            ,N'#API_ticket_num#'
            ,N'#API_starting_station#'
        ,N'#API_train_num#'
        ,N'#API_destination_station#'
            ,N'#API_date#'
            ,N'#API_ticket_rates#'
            ,N'#API_seat_category#'
            ,N'#API_name#'
            ,N'#API_id_num#'
            ,N'#API_serial_number#'
            ,N'#API_sales_station#'
            , N'#API_time#'
            ,N'#API_seat_num#'
            ,1)
           END
```

- APP 调用效果

3.4 人脸识别

- 添加附加模型，在添加模块写入以下代码



参考代码：

```
CALLFUNCTION~LINKAPI~
{
"method": "POST",
"url":
"https://aip.baidubce.com/rest/2.0/face/v3/detect?access_token=#LINKBAIDUAITOKEN#
",
"request": {
"image": "#LINKIMAGEBASE64#",
"image_type": "BASE64",
"face_field":
"age,beauty,expression,face_shape,gender,glasses,race,quality,eye_status,emotion,
face_type"
},
"accType": "json",
"contType": "application/json",
"resultFormula": "我知道了，你的年龄是#LINKJSONVALUE:$.result.face_list[0].age#岁，
#LINKJSONVALUE:$.result.face_list[0].gender.type#，漂亮指数
#LINKJSONVALUE:$.result.face_list[0].beauty#。你现在的心情是
#LINKJSONVALUE:$.result.face_list[0].emotion.type#。",
"resultList": {
"gender": "$.result.face_list[0].gender.type",
"beauty": "$.result.face_list[0].beauty"
},
"resultAutoHide":false
```

```
}
```

- 调试效果


上午11:38
5.6K/s
人脸识别
USR:JOZEYEE
COMPUTER: AAA
KEYWORDS: CBF5B7A7-96B3-4C01-9FAD-364FC0FA2A25
OPEDATE:
A001: hh
A002: [1,1,1]
A003: 010
A004: 111
A005: F11
A006: 仅提示信息
A007: true
执行结束
我知道了，你的年龄是22岁，female，漂亮指数 59.19。你现在的心情是 neutral。
确定
A018:
A019:
A020:
A021:
A022:
A023:
A024:
A025:
A026:
A027:
A028:
A029:
首页
Link
功能

4 SOAP Webservice 接口调用

在模型“功能调用”页签中，通过 CALLFUNCTION~SOAPWEB 脚本格式，可实现 Total LINK 系统与 SOAP Webservice 类型接口的信息对接。

调用格式如下：

```
CALLFUNCTION~SOAPWEB~
{ "url": "",
    "reqMethod": "",
    "codeUser": "",
    "password": "",
    "soapAction": "" }
~
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns
:web="webservices.essilor.weaver.com.cn">
   <soapenv:Header/>
     <soapenv:Body>
         <web:getDatabaseInfo/>
     </soapenv:Body>
</soapenv:Envelope>
```

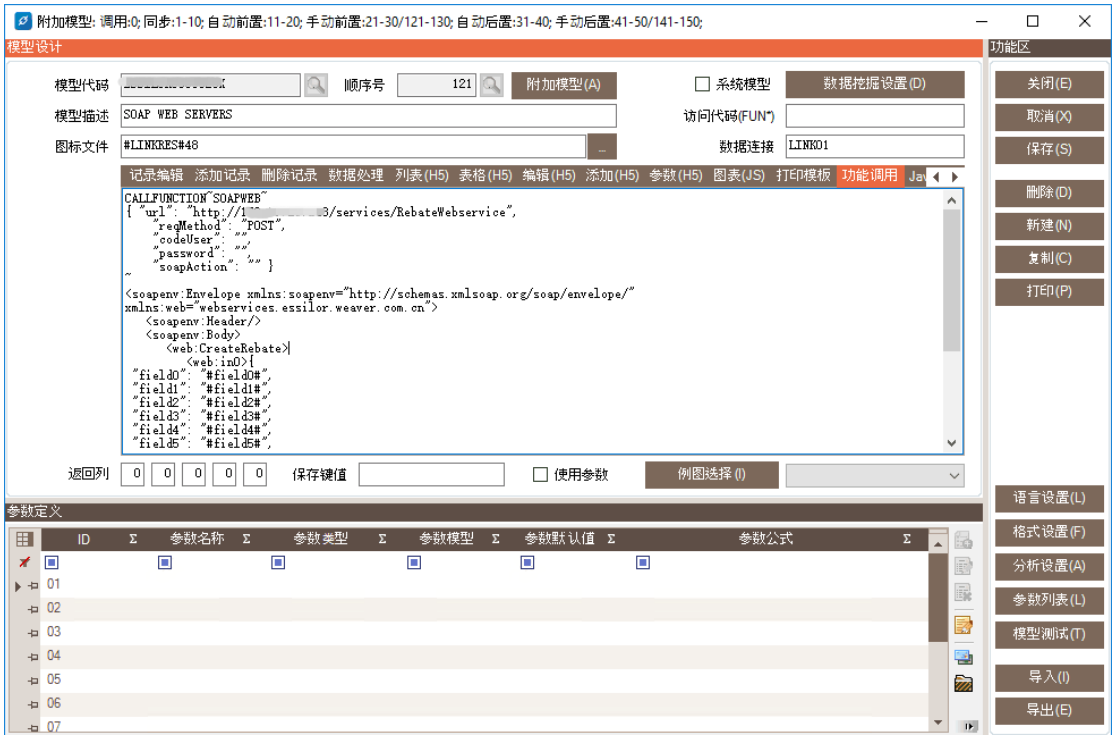
需要传输数据的关键字：

序号	关键字	关键字解释
1	url	调用的接口地址
2	reqMethod	请求方式（get\post）
3	codeUser	接口连接登录名
4	password	接口连接登录密码
5	soapAction	用来标识 SOAP HTTP 请求的目的地址

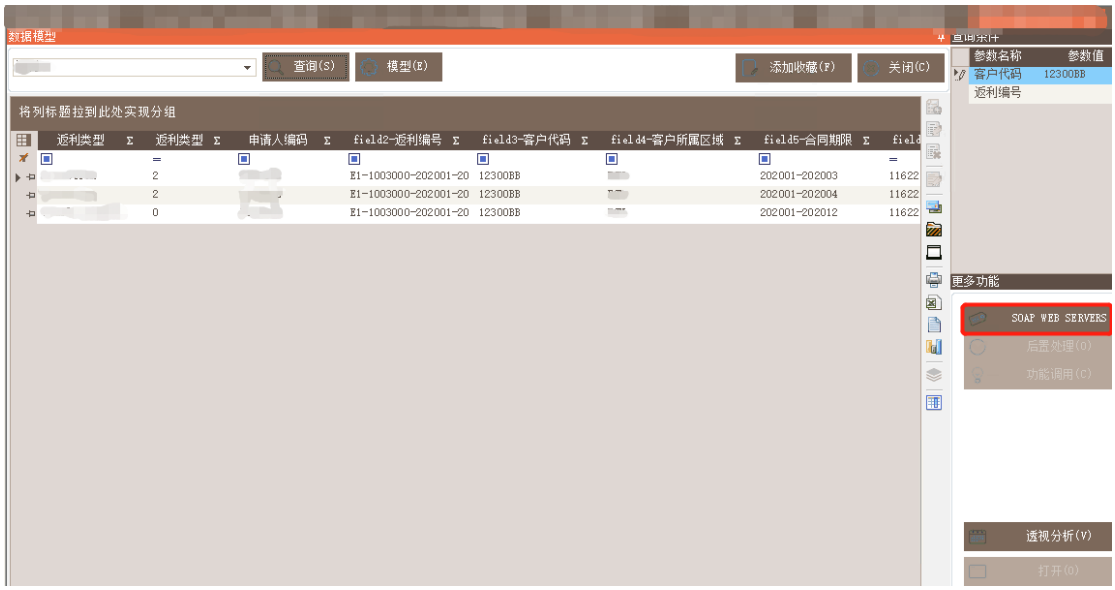
注意：以上信息均是客户提供。

在<soapenv:Body></soapenv:Body>中书写传递的信息（根据客户需求填写信息），可以用#字段名#的方式，获取主模型查询的数据。

例如：TotalLINK 系统与 SOAP Webservice 类型接口的信息对接。



CALLFUNCTION~SOAPWEB 脚本模型创建好，在主模型进行相关信息查询，点击“SAOP WEB SERVERS”按钮，提交查询信息给 SOAP Webservice 类型接口进行信息对接。



可以在操作日志中 Note02 查看是否传递成功，以下是返回成功的相关信息。

注意：每个接口返回的信息是不一样的，以下信息仅供参考。

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd=
"                    http://www.w3.org/2001/XMLSchema" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<ns1:CreateRebateResponse xmlns:ns1="webservices.****.weaver.com.cn">
<ns1:out>{"O_MSG":"0","O_REMARK":"流程[122803]创建成功！"}</ns1:out>
```

```
</ns1:CreateRebateResponse>
</soap:Body>
</soap:Envelope>
```

注意：点击“SAOP WEB SERVERS”按钮，提交查询信息给 SOAP Webservice 接口出现以下情况，原因：接口地址无法访问，需要联系相关人员检查原因。