

TotalLINK

产品手册



上海朝识智能科技有限公司

2018 年 05 月

移动应用进阶一

目 录

移动应用进阶一	2
1 参数输入页面模板	5
2 附加模型的页面模板	6
3 附加模型处理	7
4 外部功能调用 CALLFUNCTION	8
5 多行文本扫描操作	10
6 日期格式设置	13
7 页面右滑功能	14
8 页面左滑功能	19
9 操作日志检查	20
10 功能模型授权	21
11 扫描内容解析	23
11.1 扫描内容解析旧版及说明	23
11.2 扫描内容解析新版及说明	24
12 重置页面输入的内容	26
13 输入数据内容校验	27
14 图片及文件上传功能	28
15 获取当前地址信息	31
16 连续添加记录功能	32
17 引用全局变量	33
18 上下文菜单菜单激活按钮	35
19 附加功能菜单激活按钮	36
20 附加操作后返回指定模型	37
21 打印结束后跳转到某个指定的功能页面	38
22 多选操作	39
23 手写签名功能	40
24 使用录音信息	41
25 使用服务端 JavaScript 脚本	42
26 使用表格展现模式	43
27 使用模型参数列表	45
28 启用调试模式	48
29 接口调用方法	49

30	调试实例.....	51
30.1	基本模型脚本调试.....	51
30.2	H5 脚本测试—APP Data List	53
30.3	模型参数脚本测试—APP Params List.....	58
30.4	记录添加、编辑及删除—APP Edit List	59
30.5	eCharts 脚本测试	64
30.6	eCharts 调试技巧	70
31	自定义列表选择按钮.....	71
32	自定义数据过滤.....	73

文档控制

■ 主要内容

本文记录 TotalLINK APP 基本功能使用方法。

■ 更改记录

日期	版本	作者	备注
2018-05-09	1.0	曾诗明	初始发布
2020-04-30	2.0	庄江涛	图片及文件上传功能模型代码
2023-08-04	3.0	潘钰婷-Liz	修订内容、添加扫描内容解析最新操作

■ 支持版本

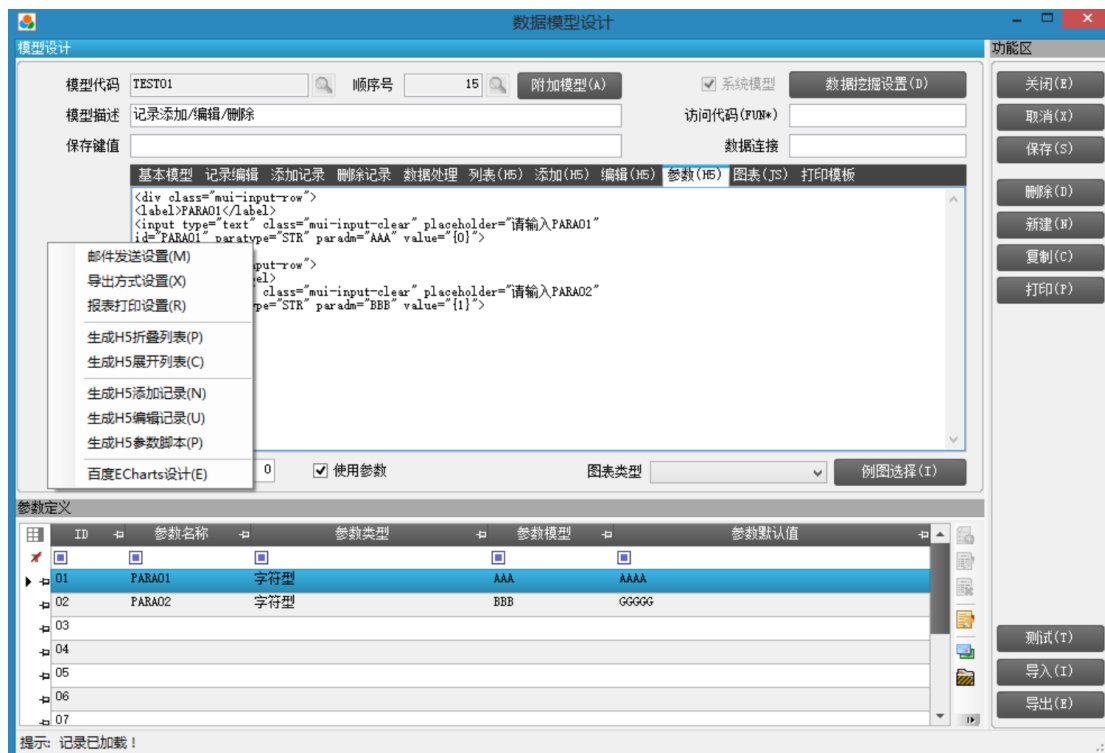
非特殊说明的功能，默认前后版本都支持

仅支持V20版本及以后版本的功能点

■

1 参数输入页面模板

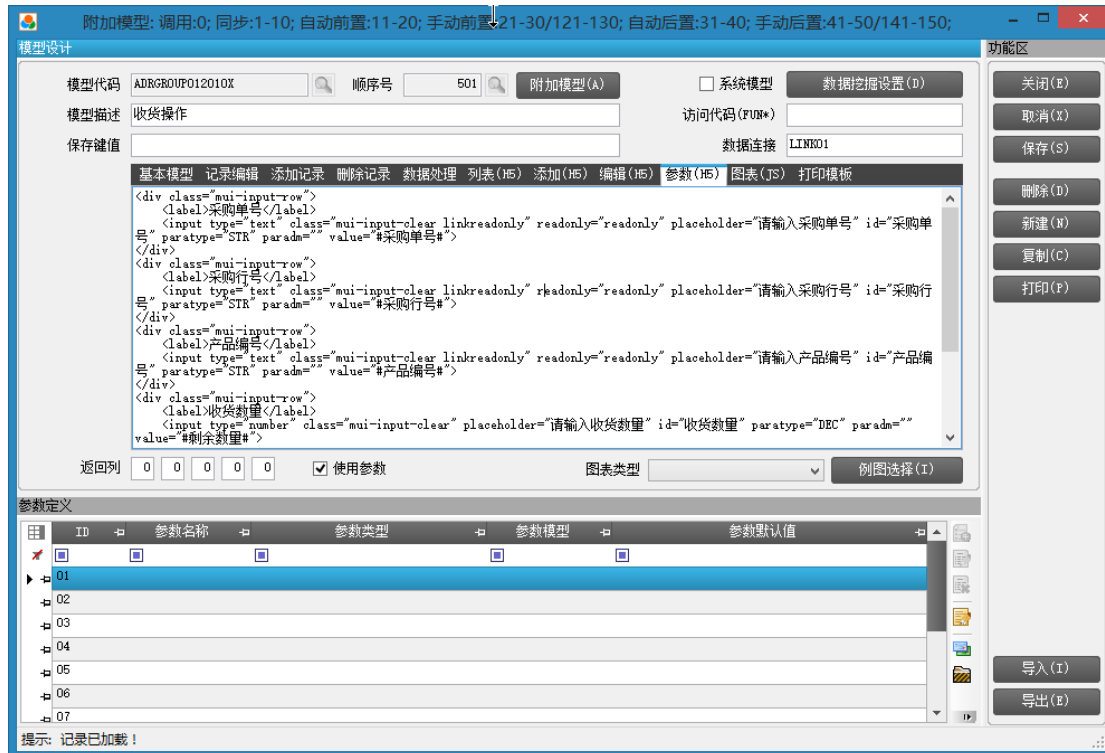
- 对于需要输入参数才可以获取数据的模型，可以为模型定义参数页面
- 当有用户自定义的页面时，使用该页面
- 否则，系统将自动生成参数页面



- 如上图所示，可以使用模型设计页面的“生成 H5 参数脚本”功能为模型自动生成默认的参数脚本
- 设计者可以在此基础上修改，根据实际需求产生智能终端上的显示页面

2 附加模型的页面模板

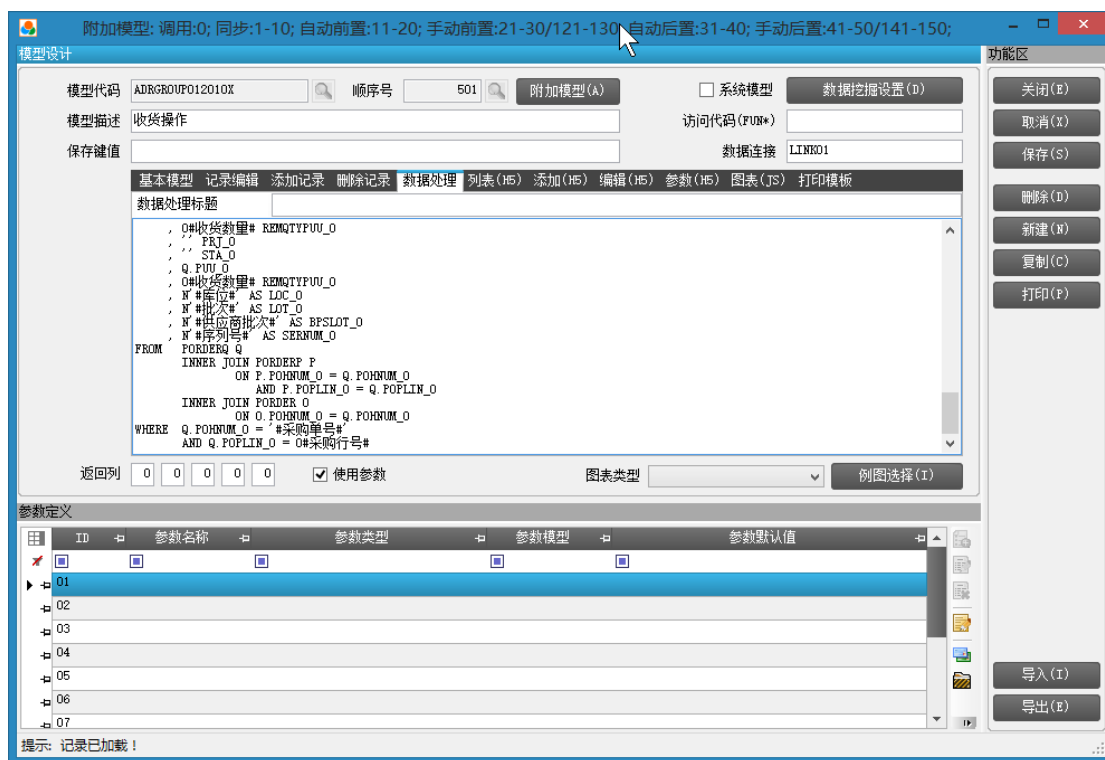
- 对于附加模型处理，如果处理过程中需要用户输入的内容，系统自动加载附加模型的参数页面



- 如图所示，在页面中可以以“#列标题#”的方式引用数据集中的内容
- 用户智能终端上加载的内容，是当前记录的实际值
- 说明
 - 在智能终端上，在加载附加模型的参数页面前，服务器会自动使用“父模型”的当期记录的内容，替换“附加模型参数页面”脚本中的内容
 - 这种方式，可以为智能终端显示的页面生成内容，比如显示的 value 值可以是进行操作时传递的实际数据
 - 在智能终端，用户执行相应“保存处理”操作时，会自动使用该页面上的内容
 - 在数据处理脚本中，如果需要引用页面上的内容，注意需要与 input 行的 id 属性对应

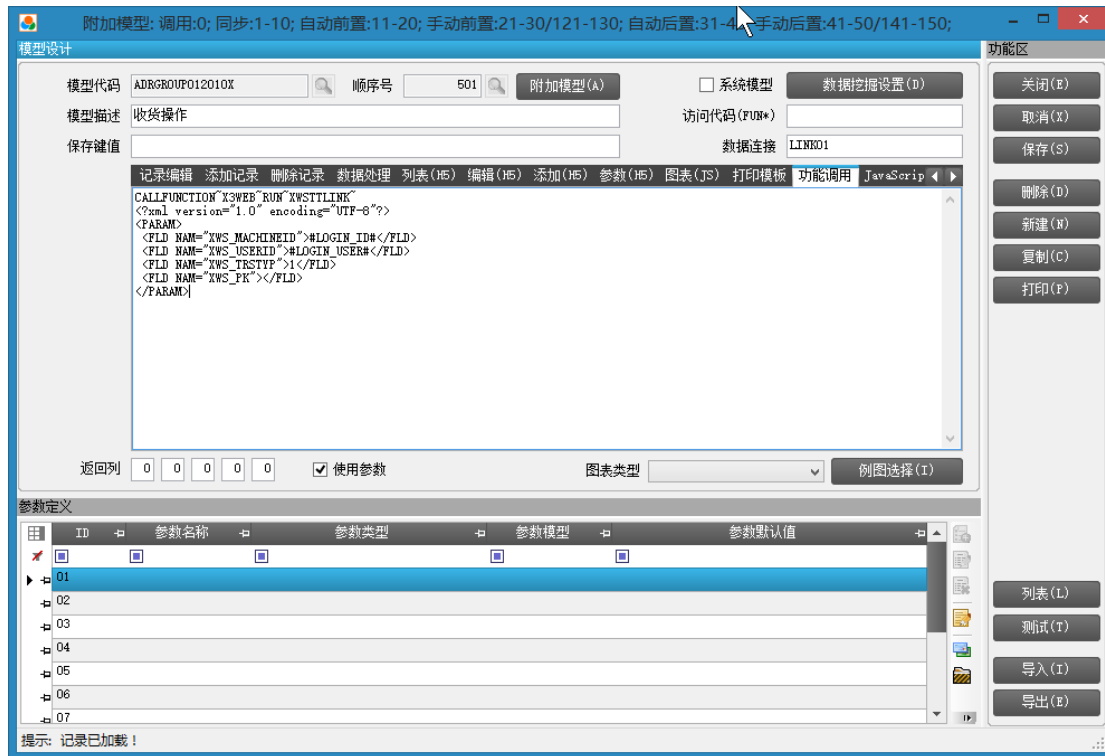
3 附加模型处理

- 附加模型处理时，执行“数据处理”这个页面的脚本
- 脚本中可以引用“附加模型参数页面”的内容
- 附加模型自动继承父模型的参数列表，在附加模型的处理逻辑中，可以与父模型一样引用参数值列表
- 在提交处理时，系统自动使用用户页面的内容



4 外部功能调用 CALLFUNCTION

- 在附件模型的执行事件中,可以通过定义特殊的脚本用于触发对外部系统 API 的调用
- 比如,可以调用 SAGE X3 Web Service 等等



- 如图所示,在某个模型的附加功能“审核”操作中,定义了 X3 Web Service 的调用触发
- 此处定义的触发脚本如下:

```
CALLFUNCTION~X3WEB~RUN~XWSTTLINK~
<?xml version="1.0" encoding="UTF-8"?>
<PARAM>
  <FLD NAM="XWS_MACHINEID">#LOGIN_ID#</FLD>
  <FLD NAM="XWS_USERID">#LOGIN_USER#</FLD>
  <FLD NAM="XWS_TRSTYP">1</FLD>
  <FLD NAM="XWS_PK"></FLD>
</PARAM>
```

- 外部功能调用触发的脚本有如下几项规定
 - 功能以 CALLFUNCTION 开始
 - 各个参数之间以“~”分割
 - 参数传递按照外部系统的要求

- 可以使用#xxx#等形式，以传递数据信息

5 多行文本扫描操作

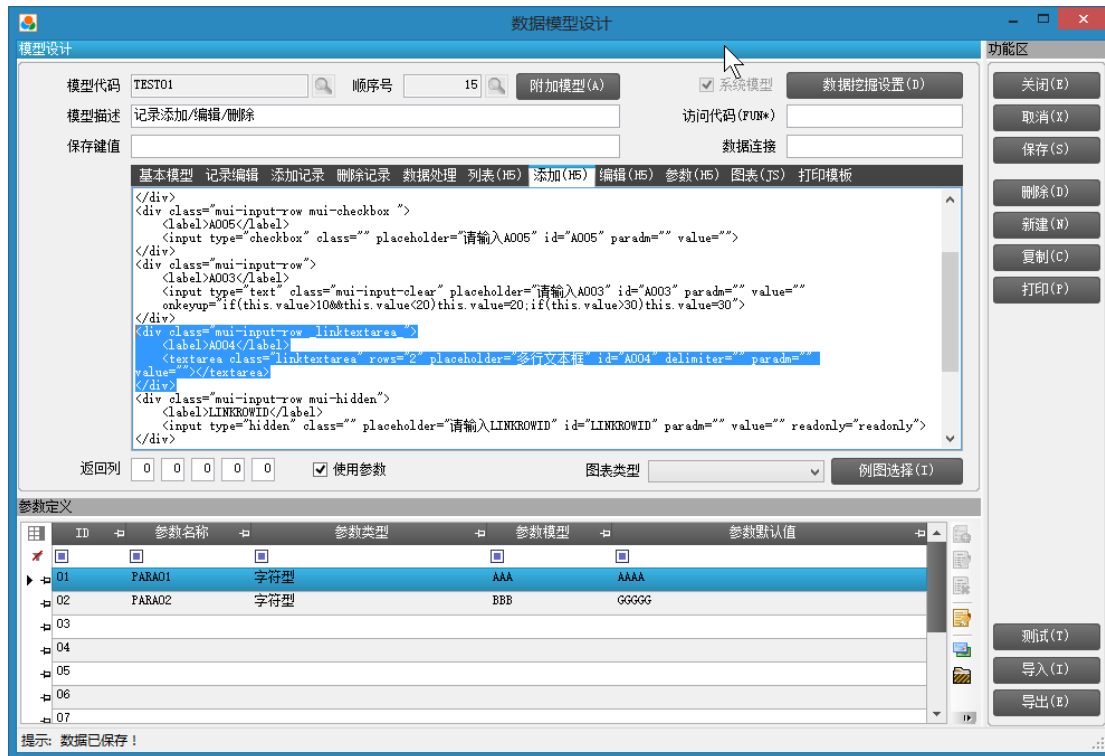
- 在设计输入页面的时候，可以使用多行文本，其操作方式如下
- 在添加模型上的模板示例

- HTML5 代码示例:

```
<div class="mui-input-row _linktextarea_">
<label>A004</label>
<textarea class=" linktextarea" rows="2" placeholder="多行文本框"
id="A004" delimiter="" paradm="" value=""></textarea>
</div>
```

- 关键<textarea></textarea>标签内需要添加一个 class="linktextarea"作为标识
- delimiter 为连接两个扫描结果之间的连接符
可自行定义，默认为换行即 delimiter="", 也可定义为 delimiter="----"等
- linktextarea 标识、delimiter 和 id 值为必填项
- rows= "2" 为文本框的行数
可自行定义，超过则隐藏并启动 scroll 滑块
- placeholder="多行文本框", 用于设置自定义的提示文本。
- <div class="mui-input-row _linktextarea_"></div>
外围的 div 需要包含一个 class 值_linktextarea_用于取消多行文本框的高度限制，
否则样式会错乱，建议加上。

■ 示例、效果截图



delimiter="" 默认效果截图

delimiter="----" 效果截图

添加: 记录添加/编辑/删除

扫描输入区

A004	#123456# 05585656
------	----------------------

继续添加 ☐ OFF

自动保存 ☐ OFF

添加: 记录添加/编辑/删除

扫描输入区

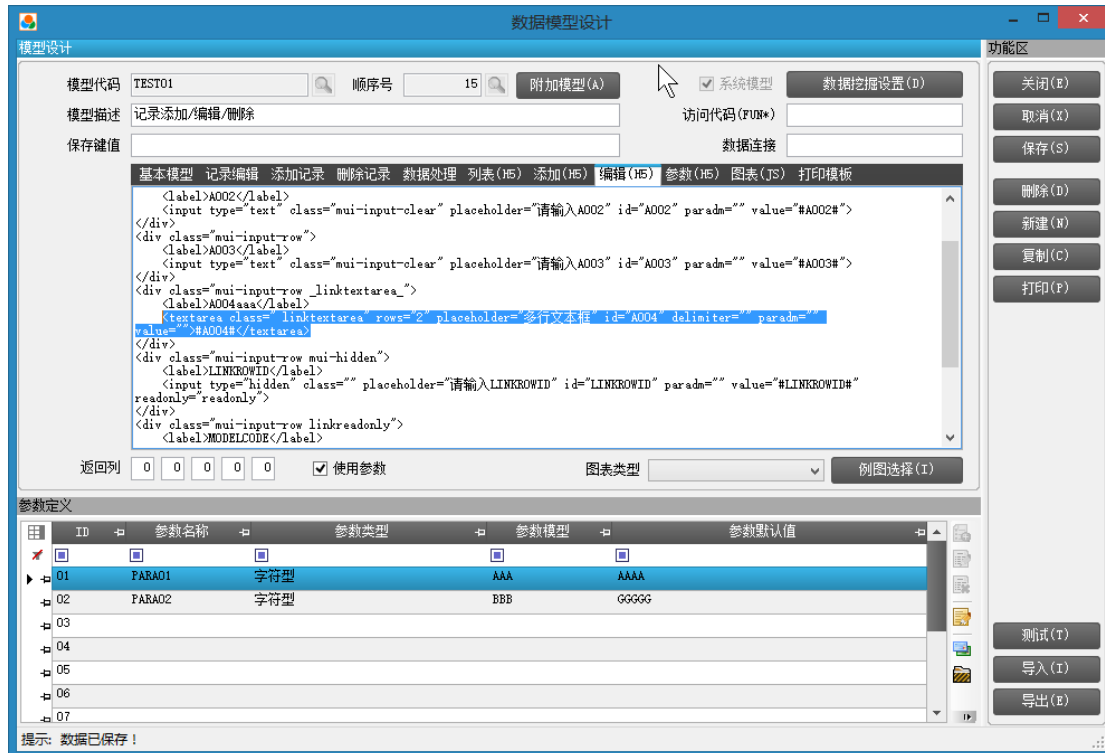
A004	#123456#----05585656
------	----------------------

继续添加 ☐ OFF

自动保存 ☐ OFF

- 在编辑模型上的模板示例
 - 原理与上面的添加模型类似，不同在于，对记录进行编辑的时候，需要传入原来的数据，因此需要把动态变量填入<textarea>标签里面
 - 示例脚本如下

```
<textarea class=" linktextarea" rows="2" placeholder="多行文本框"
id="A004" delimiter="" paramd="" value="">#A004#</textarea>
```





6 日期格式设置

- 在 APP 上，为了能够正常显示日期格式，需要使用“YYYY-MM-DD”的格式显示日期
 - 在脚本中请通过数据转换使得系统可以获得正确的日期
- MSSQL: `CONVERT(VARCHAR(10), GETDATE(), 120)`
 - ORACLE: `TO_CHAR(SYSDATE, 'YYYY-MM-DD')`

7 页面右滑功能

- 在下列页面操作过程中，通过向右滑的操作，可以直接返回功能菜单页面
 - 返回的功能菜单页面，例如



- 下列页面操作过程中可通过向右滑的操作返回功能菜单界面
 - 参数输入页面



■ 查询内容页面



■ 长按列表参数更新页面



■ 添加、编辑界面

<
添加: 记录添加/编辑/删除

扫描输入区

A001-UD	请输入A001
A002	请输入A002
A005	☐
A003	请输入A003
A004	多行文本框

MODELCODE	请输入MODELCODE
MODELNUM	请输入MODELNUM

继续添加 ☐ OFF

自动保存 ☐ OFF

取消 重置 保存

<
编辑: 记录添加/编辑/删除

扫描输入区

A001	eee
A002	请输入A002
A003	请输入A003
A004	多行文本框

MODELCODE	TEST01
MODELNUM	15

取消 重置 保存

■ 反馈记录的添加/编辑界面

<
添加: 反馈记录

扫描输入区

单据类型	请输入单据类型
单据号码	请输入单据号码
单据标识	请输入单据标识
记录日期	
关键字	请输入关键字
描述	请输入描述
备注	请输入备注
FLAG01	☐
FLAG02	☐
图片	

取消 重置 保存

<
编辑: 反馈记录

扫描输入区

单据类型	请输入单据类型
单据号码	yyh
单据标识	ggg
记录日期	2017/12/28
关键字	请输入关键字
描述	请输入描述
备注	请输入备注
FLAG01	☐
FLAG02	☐
图片	

取消 重置 保存

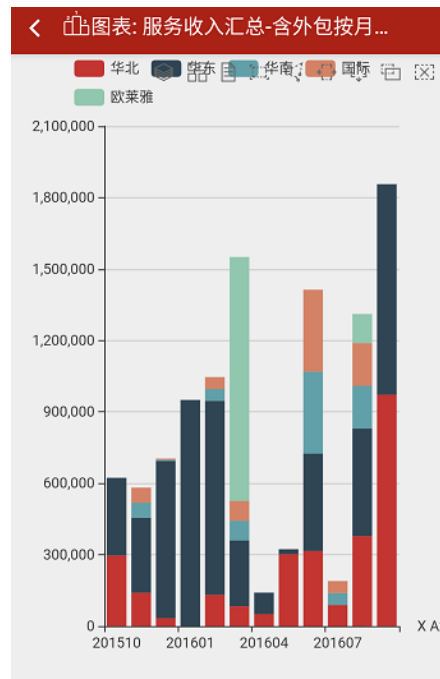
■ 查看地图界面



■ 标签打印界面

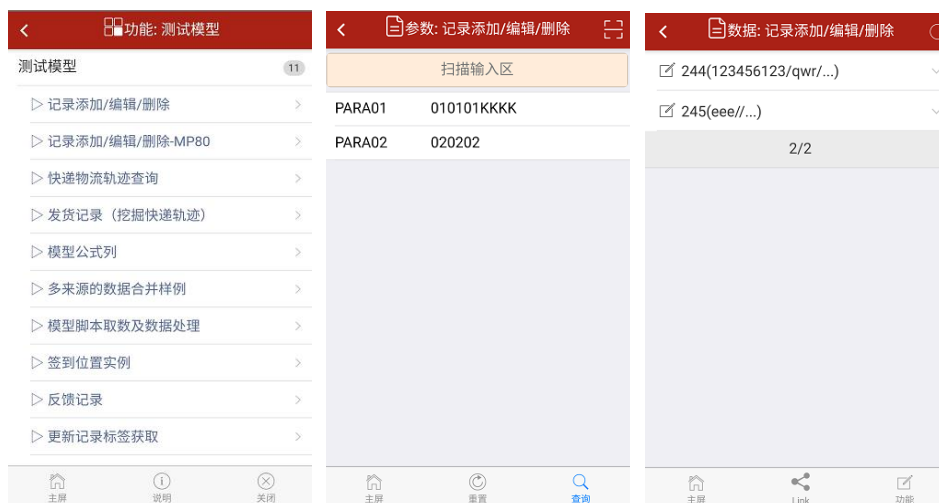


■ 图表界面



8 页面左滑功能

- 说明，为了更加明确每个功能的操作，新版本系统中已经停止使用左滑返回上一页的功能。
- 在下列页面，执行向左滑动的操作，可以跳转到上一次操作的界面
 - 例如，在查询结果的界面
 - 通过返回键返回或者通过向右滑动返回到功能菜单列表界面
 - 然后即可通过向左滑动的操作返回上一次界面
 - 通过向左滑动可依次返回，如下图所示顺序返回
 - IOS 系统不支持这一操作



- 可执行向左滑动操作的界面（程序截图与向右滑动功能类似）
- ◆ 查询结果界面
- ◆ 参数输入界面
- ◆ 添加编辑界面
- ◆ 反馈记录的添加编辑界面
- ◆ 参数更新界面
- ◆ 功能菜单列表界面

9 操作日志检查

数据模型

用户操作日志

查询(S) 模型(M)

取消(R) 退出(X)

新表名称: 参数值

起始日期: 2018年01月13日

截止日期: 2018年01月14日

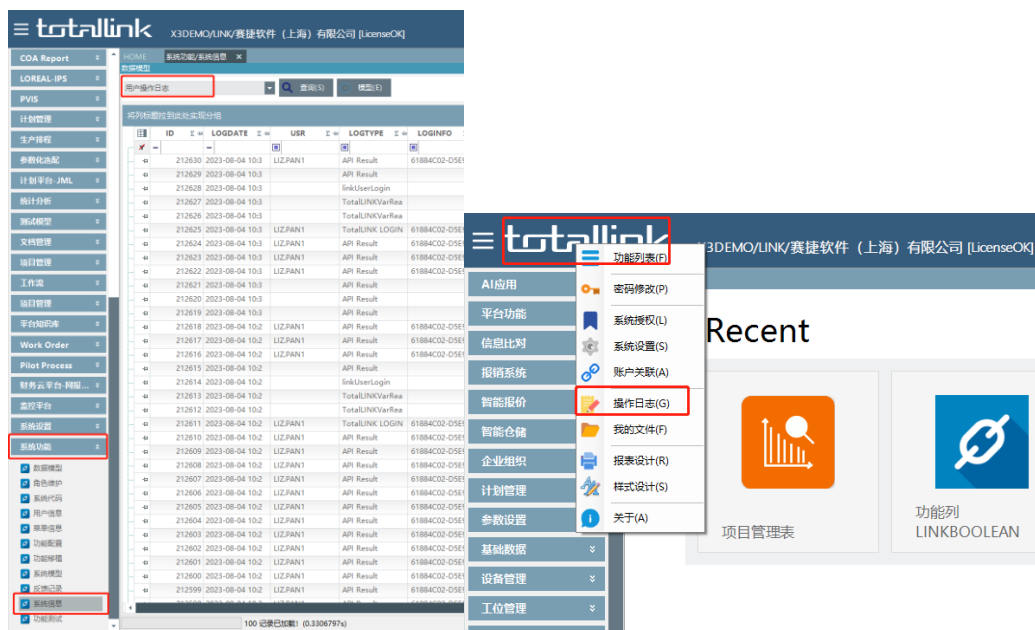
用户代码

将列标题拉到此处实现分组

ID	LOGDATE	LOGTYPE	LOGINFO	KEY01	KEY02	KEY03	KEY04	NOTE01	USER	NOTE02
786313	2018-01-14 14:13:40.677	ModelDataGetItem	590P95C3-67A9-46	SYSTEMINFORMATION	10	用户操作日志	LINKBI	SELECT ID	AS (模型代码)	AS (模型代码)
786312	2018-01-14 14:13:32.673	ModelDataGetItem	590P95C3-67A9-46	SYSTEMINFORMATION	10	用户操作日志	LINKBI	SELECT ID	AS (模型代码)	AS (模型代码)
786311	2018-01-14 14:13:29.523	GetLoginInfo	590P95C3-67A9-46							
786310	2018-01-14 13:50:15.483	GetLoginInfo	590P95C3-67A9-46							
786309	2018-01-14 13:48:38.200	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelList	15	记录添加/编辑/删除	LINKBI	SELECT A001	RANDY.LIU	
786308	2018-01-14 13:48:34.523	ModelDataGetItem	C4748B2-EB7F-4D63-9705-B96	TEST01						
786307	2018-01-14 13:48:34.517	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786306	2018-01-14 13:48:32.777	Link Web API	C4748B2-EB7F-4D63-9705-B96	TEST01						
786305	2018-01-14 13:48:32.767	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786304	2018-01-14 13:48:27.053	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786303	2018-01-14 13:48:18.710	GetLoginInfo	590P95C3-67A9-46							
786302	2018-01-14 13:47:57.370	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786301	2018-01-14 13:47:48.643	GetLoginInfo	590P95C3-67A9-46							
786300	2018-01-14 13:47:03.113	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786299	2018-01-14 13:46:58.793	ModelDataGetItem	C4748B2-EB7F-4D63-9705-B96	TEST01						
786298	2018-01-14 13:46:58.787	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786297	2018-01-14 13:46:58.297	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786296	2018-01-14 13:46:50.187	ModelDataGetItem	C4748B2-EB7F-4D63-9705-B96	LINKSYS	20	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786295	2018-01-14 13:46:50.180	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786294	2018-01-14 13:46:46.287	ModelDataGetItem	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786293	2018-01-14 13:46:46.280	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786292	2018-01-14 13:46:45.690	Login	C4748B2-EB7F-4D63-9705-B96							
786291	2018-01-14 13:46:45.683	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786290	2018-01-14 13:46:45.680	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786289	2018-01-14 13:46:45.620	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786288	2018-01-14 13:46:40.367	ModelDataGetItem	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786287	2018-01-14 13:46:40.327	Link Web API	C4748B2-EB7F-4D63-9705-B96	linkModelData	15	记录添加/编辑/删除	LINKBI	UPDATE LINKMDP	RANDY.LIU	
786286	2018-01-14 13:46:19.787	Login	C4748B2-EB7F-4D63-9705-B96							
786285	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786284	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786283	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786282	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786281	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786280	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786279	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786278	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786277	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786276	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786275	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	
786274	2018-01-14 13:46:19.740	Link Web API	C4748B2-EB7F-4D63-9705-B96	LINKSYS	10	用户菜单及模型(不可修)	LINKBI	GETUSERMODEL	RANDY.LIU	

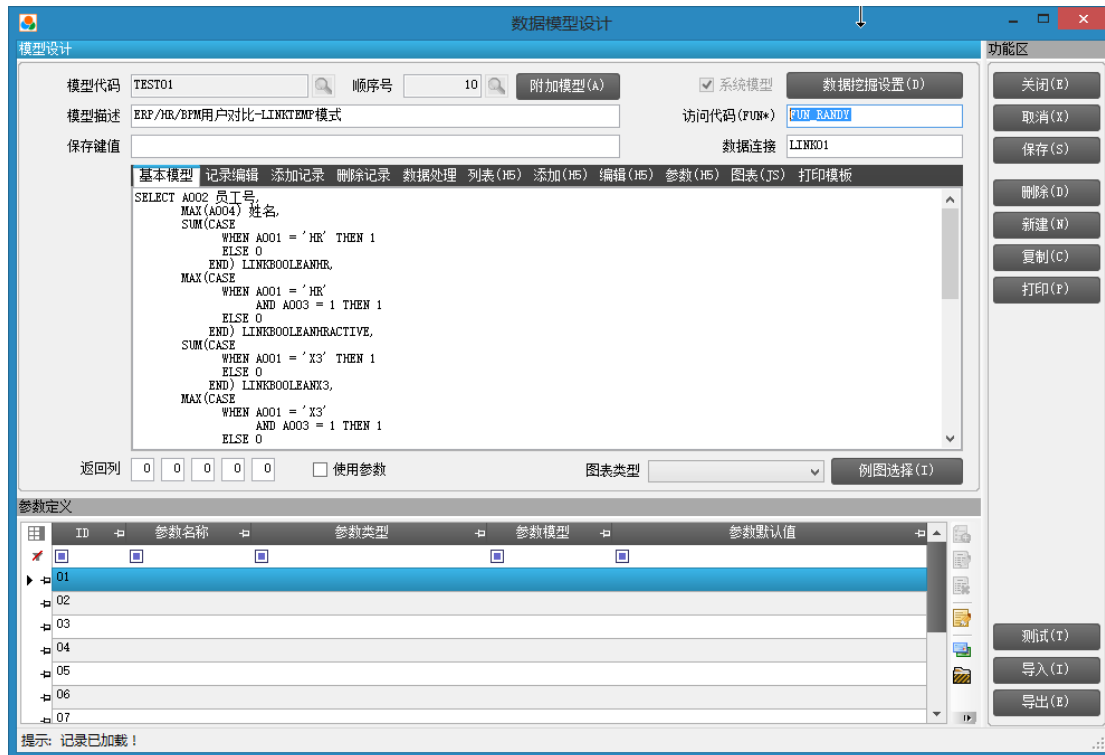
513 记录已加载 (0.531435s)

- 如图所示，在用户使用系统平台的过程中，根据设置，可以保留详细的日志记录
- 模型日志记录的操作路径有两种：
 - 系统功能→系统信息→用户操作日志
 - 或者鼠标右击 TotalLINK 图标选中操作日志



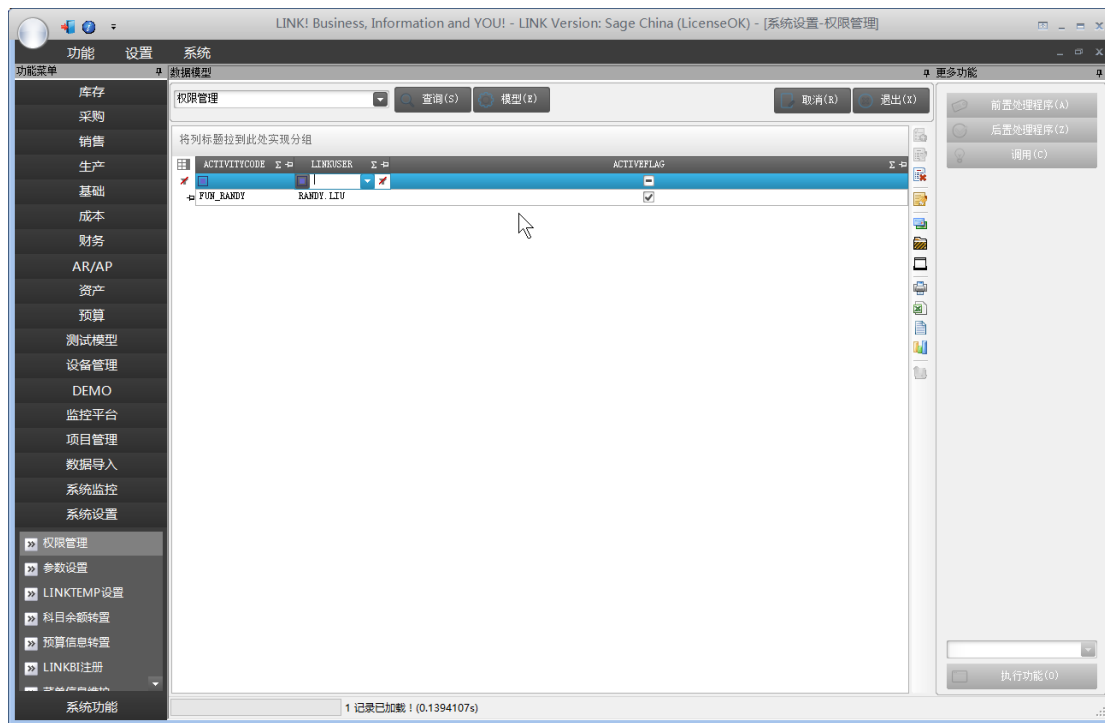
10 功能模型授权

- 可以通过在模型中设定“授权代码”的方式，为模型授权
- 仅当模型的授权代码赋值给某个用户的时候，该用户才有该模型的访问权限
- 未经过授权的功能，对用户是不可见的



- 如图所示，通过填写访问代码的方式为模型进行授权

- 模型的访问代码授权给特定的用户

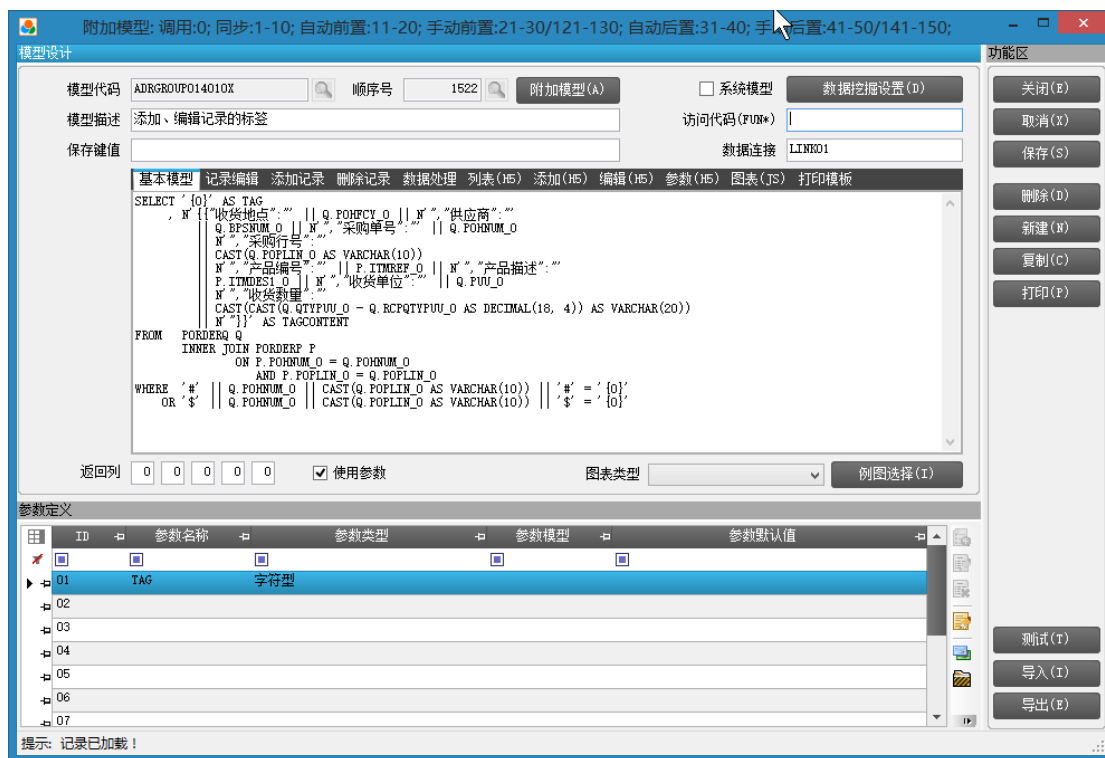


- 可以进行授权的模型包括
 - 基本模型
 - 各类附加模型等
 - 关于如何设置访问代码，请参考文档《[LINK5 专家功能](#)》

11 扫描内容解析

11.1 扫描内容解析旧版及说明

- 在参数输入、添加编辑、附件功能处理等页面上，有一个可以进行解析的扫描输入框（linkinput）
- 通过该输入框进行扫描输入，可以实现如下功能
 - JSON 解析
 - ◆ 如果扫描得到的结果是 JSON 格式的内容，系统将自动对内容进行解析，然后将解析出的结果自动填写到页面中
 - ◆ 填写时，按照 input 输入框的 id 进行对应匹配
 - 标签解析
 - ◆ 如果扫描得到的结果是类似#TAG#或者\$TAG\$格式的内容，系统自动向服务器请求需要解析的内容
 - ◆ 请求解析的规则如下：
 - 如果是模型查询参数输入页面，使用附加模型 1521 进行解析处理
 - 如果是模型的添加或者编辑记录页面，使用附件模型的 1522 进行解析处理
 - 如果是附加处理模型 501-520，使用附加模型 1501-1520 进行解析处理



- 如图所示，为某个模型定义了添加、编辑记录的标签扫描解析功能

- 在智能终端上扫描到类似于
#POH2017010100011000#或者\$ POH2017010100011000\$
的内容时，系统会根据要求返回标签所对应的内容

- 这里包括如下各项内容

- 收货地点
- 供应商
- 采购单号
- 采购行号
- 产品编号
- 产品描述
- 收货数量

11.2 扫描内容解析新版及说明

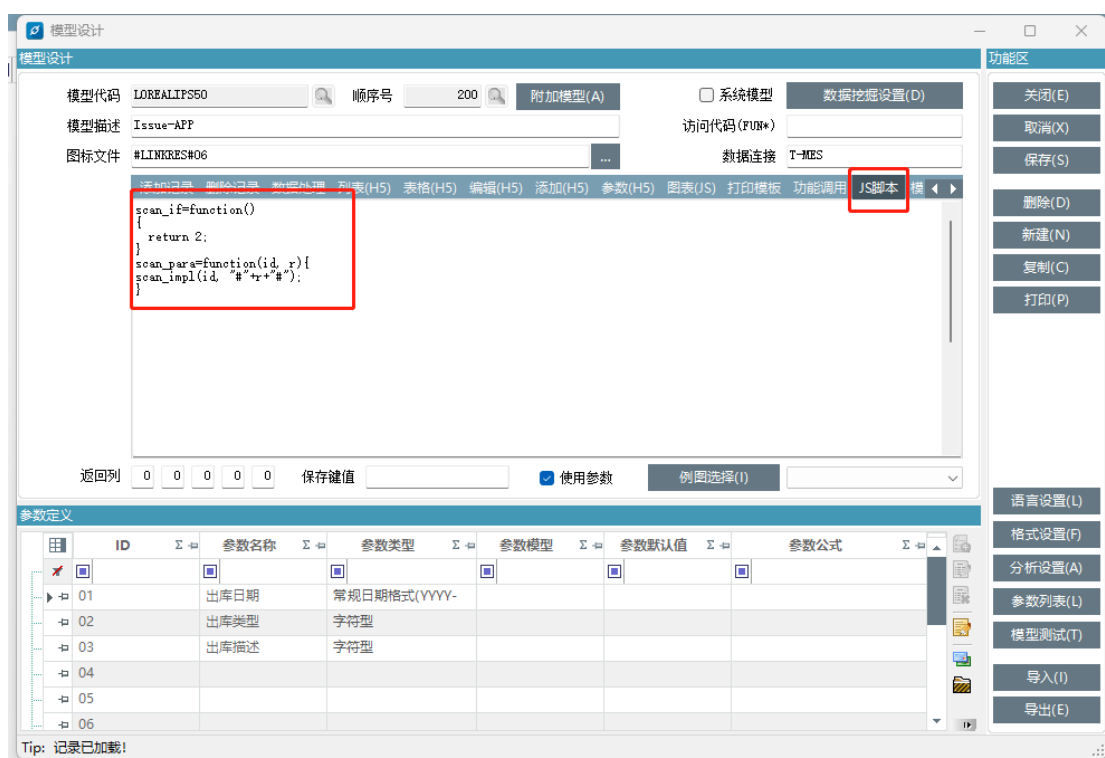
- 新版的解析不用编写复杂的 Json 结构，不需要#TAG#或者\$TAG\$格式的内容，示例具体如下：

- 创建一个扫描解析附加模型，这里用于添加界面，使用 1522 的模型，待扫描的标签内容用 ‘{0}’ 代替



Tip: 记录已加载!

- 在基础模型中的 JS 脚本页签中加上一段解析代码，用于给扫描内容附上##，便于解析，若扫描内容为#内容#、\$内容\$的格式则不需要加上 JS 脚本页签中的这段解析代码



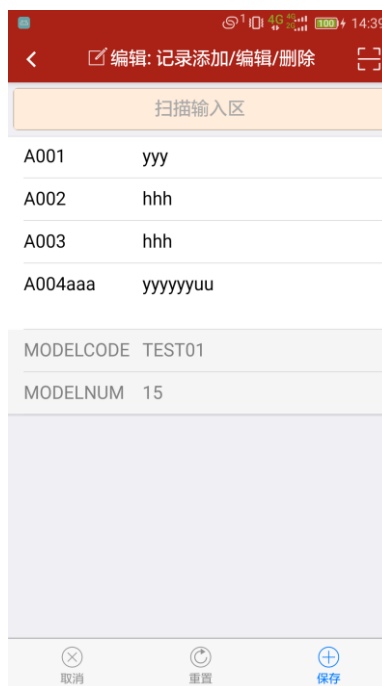
- 代码内容如下：

```
scan_if=function()
{
    return 2;
}

scan_para=function(id, r){
    scan_impl(id, "#"+r+"#");
}
```

12 重置页面输入的内容

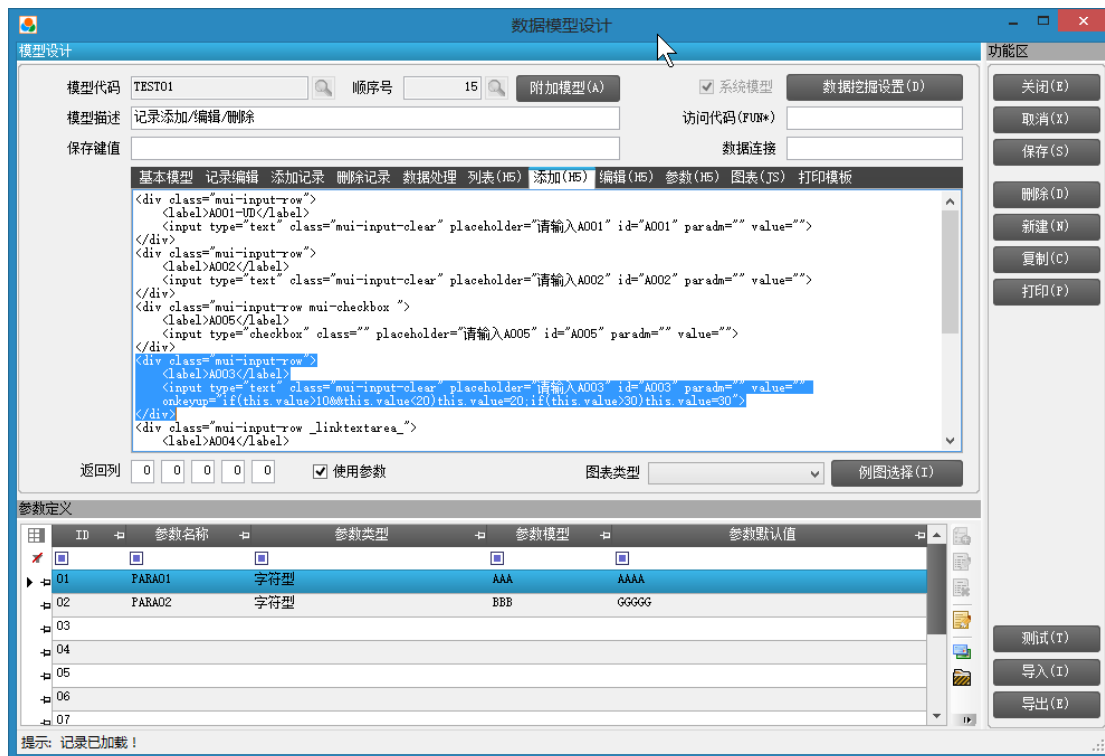
- 在有扫描输入的页面上,包括“添加记录、编辑记录、参数输入、附加参数输入……”等页面上,可以使用底部的“重置(RESET)”按钮清除页面上输入的记录



- 执行该操作时,可清除的内容包括
 - class 中包含 mui-input-clear 的各个输入框
 - linkinput 输入框等
- 如果不希望清除某个输入框的内容,请清除 class 中的 mui-input-clear 标记

13 输入数据内容校验

- 对于用户输入的数据，可以在模型的页面脚本中植入校验脚本，以便对用户的数据合法性进行校验



- 在上述模型中嵌入了如下脚本

```
<div class="mui-input-row">
  <label>A003</label>
  <input type="text" class="mui-input-clear" placeholder="请输入 A003" id="A003"
  paradm="" value=""
  onkeyup="if(this.value>10&&this.value<20)this.value=20;if(this.value>30)this.value=30
">
</div>
```

- 这样，就可以对数据进行合法性校验，控制用户输入的数据在 20-30 之间
- 在实际模型的设计过程中，可以使用动态的校验规则
- 也可以使用 JS 控制提示判断，来实现输入数据内容校验，具体可以参考文档 [《LINK26 移动应用进阶二》](#)

14 图片及文件上传功能

- 调用上传附件的功能有两种：
 - 模型代码为 SYSTEMLINKMEMO
 - 模型代码后缀为_LKMM
- 在“系统功能”的模型上“反馈记录”（模型代码为 SYSTEMLINKMEMO）里面，点击“功能”，选择添加或编辑即可上传或修改图片以及文件
 - 图片可选择相册或拍照模式上传，最多允许上传 9 张图片
 - 文件上传可选择本地存储空间里面的文件，最多允许上传 9 个文件
- 模型实例
 - 添加的关键代码

1. `<li class="mui-table-view-cell _linkimage_">` 图片的关键样式需要加上 `_linkimage_`，建议加上，否则有些功能会显示不正确

2. `<div class="mui-row mui-content-padded" id="carema_pic_foot">`

`<label class="">图片</label>`

`<div class="mui-icon mui-icon-image mui-pull-right"></div>`

`</div>`

选择图片的按钮 `id="carema_pic_foot"` 必须加上，上传文件是 `id="link-file-btn"` 并需要加上

`<input id="file_btn" style="display: none;" type="file" multiple size="3" onchange="choose_file(this.files)" />`

3. `<div class="linkingDiv" id="linkingDiv"></div>`

红色字体为图片预览关键,上传文件的预览为 `id="link-file"`

`</li>`

- 说明：在“系统功能”——“反馈记录”下的模型设计下的添加截图如下

模型代码 顺序号 附加模型(A) ☐ 系统模型 数据挖掘设置(D)

模型描述 访问代码(FUN*)

保存键值 数据连接

基本模型 记录编辑 添加记录 删除记录 数据处理 列表(H5) 编辑(H5) 添加(H5) 参数(H5) 图表(JS) 打印模板 功能调用

```

<ul class="mui-table-view">
  <li class="mui-table-view-cell linkimage">
    <div class="mui-row mui-content-padded" id="carema_pic_foot">
      <label class="">图片</label>
      <div class="mui-icon mui-icon-image mui-pull-right"></div>
    </div>
    <div class="linkingDiv" id="linkingDiv">
    </div>
  </li>
</ul>
<ul class="mui-table-view">
  <li class="mui-table-view-cell linkimage" id="">
    <div class="mui-row mui-content-padded" id="link-file-btn">
      <label>附件</label>
      <div class="mui-icon mui-icon-paperclip mui-pull-right"></div>
    </div>
    <input id="file_btn" style="display: none;" type="file" multiple size="3" onchange="choose_file(this.files)" />
    <div id="link-file">
    </div>
  </li>
</ul>

```

■ 编辑关键代码

与添加的代码大概一致，区别在于

1. 图片，需要在 id=" linkingDIv" 的 div 下添加 9 个预览图片 div 如下：

```

<div class="linkimage" >
  <span class="mui-icon mui-icon-close link-icon-del"></span>
  
</div>

```

2. 文件，需要在 id=" link-file" 的 div 下添加 9 个预览文件的 div ，如下：

```

<div class="mui-card-header mui-card-media linkfile"
data-src="#ATTPATH01#">

<div class="mui-media-body">
<span class="mui-icon mui-icon-close link-icon-del"></span>
#ATT01#
<p>文件</p>
</div>
</div>

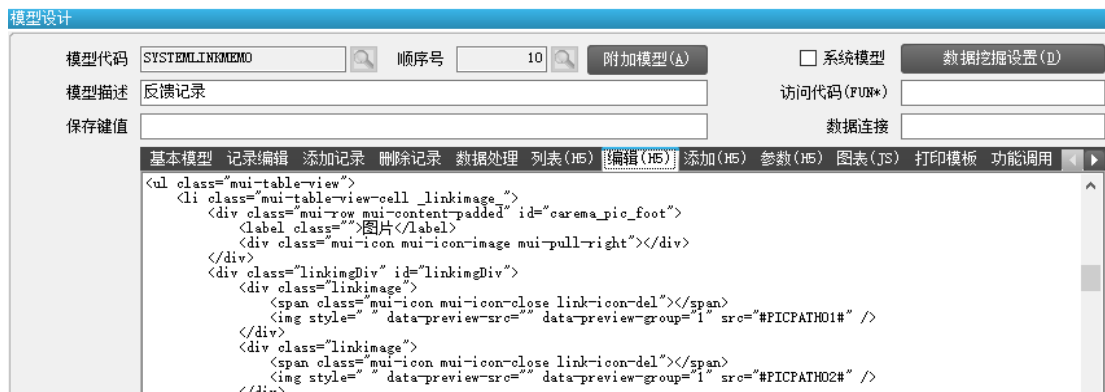
```

■ 说明

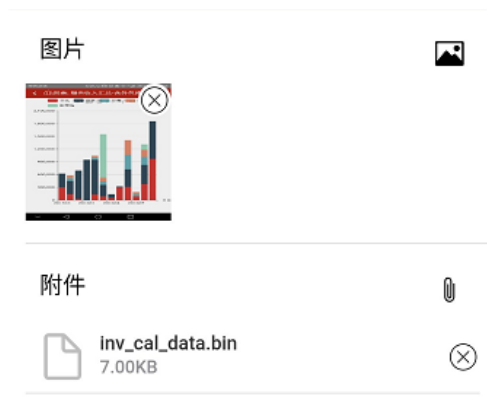
#PICPATH01# ~ #PICPATH09#以及#ATTPATH01# ~ #ATTPATH09#和#ATT01# ~ #ATT09#为系统自动赋值给 9 个图片或文件的地址资源数据等



部分截图



效果截图



15 获取当前地址信息

- 模型实例

- 添加关键代码

```
<li class="mui-table-view-cell link-location" id="linklocDiv"> 样式 class

<div id="linklocation">样式 id

<a id="link_a_location"></a>显示地址

<button type="button" class="mui-btn mui-pull-right" style="padding: 3px;"
id="showMapBtn"><span class="mui-icon mui-icon-map"></span>地图</button>

显示地图的按钮需要加上此 id
```

- 编辑与添加类似区别在于加入#LOCATION#自动赋予数据库里的相应地址信息

```
<a id="link_a_location">#LOCATION#</a>
```

相关截图如下

```
<ul class="mui-table-view">
<li class="mui-table-view-cell link-location" id="linklocDiv">
<div class="mui-row">
<div id="linklocation">
<span class="mui-icon mui-icon-location"></span> 所在位置:
<a id="link_a_location"></a>
<button type="button" class="mui-btn mui-pull-right" style="padding: 3px;" id="showMapBtn"><span
class="mui-icon mui-icon-map"></span>地图</button>
</div>
</div>
</li>
</ul>
```

- 效果实例

下图为地理定位，右图为查看地图



16 连续添加记录功能

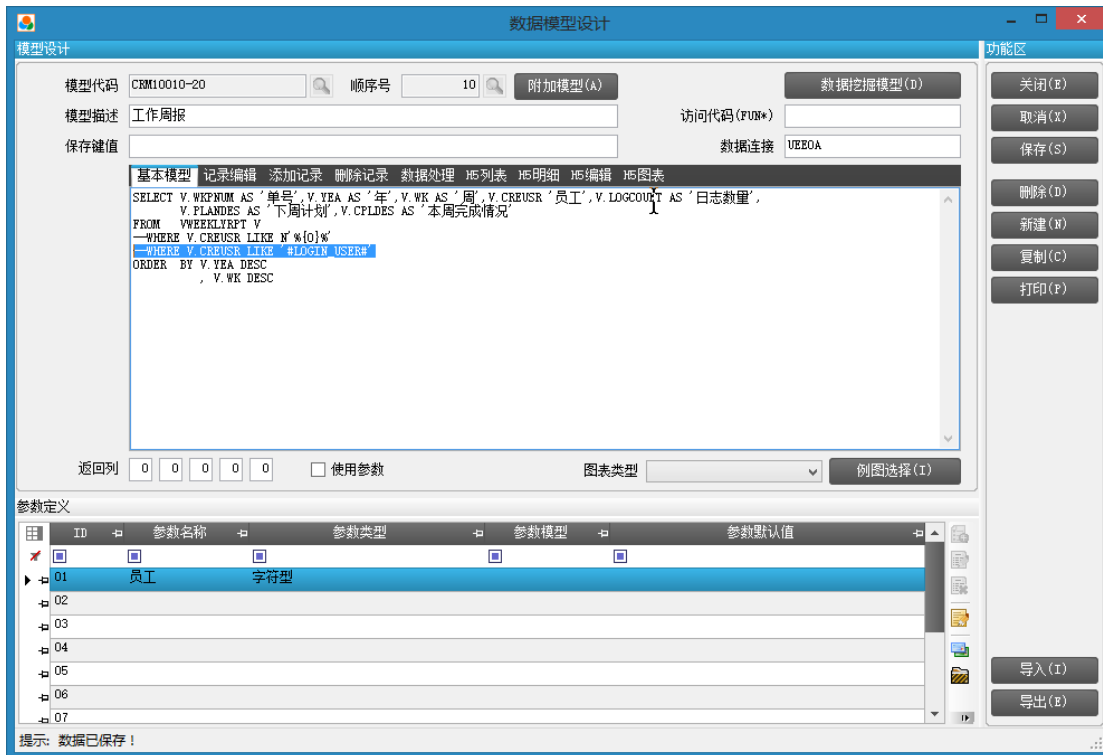
- 在添加记录的最下方会有个“连续添加”按钮，如图：

继续添加



- 当该按钮处于 on 状态的时候，在保存后，会留在添加界面，重置内容框并可继续添加

17 引用全局变量



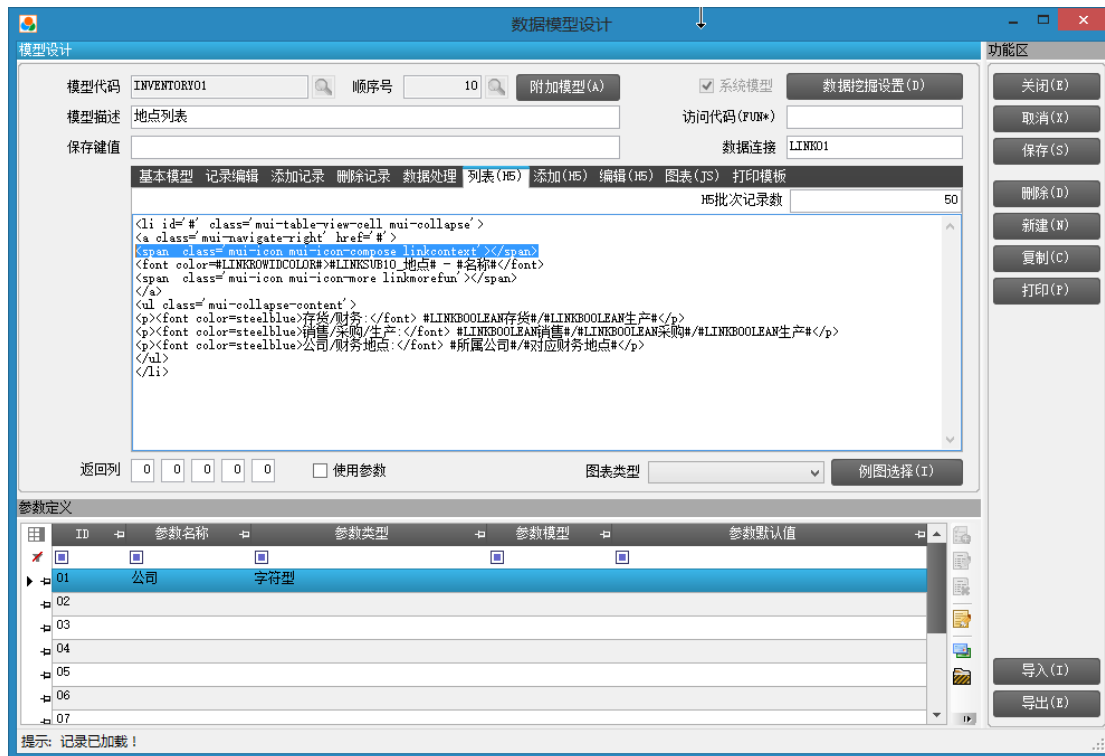
- 在模型的脚本中，引用“#LOGIN_USER#”，即可使用登录的用户代码作为脚本的一部分过滤数据，从而实现用户只能查询自己的数据
- 如果用户需要查询的数据与“用户代码”并不是简单直接的过滤关系，在模型设计的时候，可以建立一个关系表，使用数据关联的方式进行处理
 - 比如有一个表，记录了用户与其管理者的关系，在表中记录了 USER-MANAGE 之间的关系，则可以实现 MANAGER 登录系统，可以查看其管理的 User(s)所创建的业务单据
 - 更加复杂的权限管理规则也可以参考类似的方式实现，也可以在数据层面通过存储过程或者函数实现
- 处理数据过滤的基本逻辑是，以用户登录的代码作为入口，通过逻辑实现该入口信息与数据之间的关系，从而实现数据行的权限管理
- 可以直接引用的全局代码包括
- 灵活运用特殊的全局变量可以使得模型的编制更加简单
- 系统中提供的全局变量包括（可参考《[LINK07 专家功能](#)》中有关全局变量使用的描述）
 - #LOGIN_USER#——当前登录的用户代码
 - #LOGIN_ROLE#——当前登录的用户对应的角色代码



- #LOGIN_ID#——当前登录用户对应的 ID
- #LOGIN_COMPUTER#——当前登录的电脑名称
- #CURR_MODELCODE#——当前执行的模型的模型代码
- #CURR_MODELNUM#——当前执行的模型的模型编号
- #ATTACH_MODELCODE#——当前执行的模型的附加模型代码
- #LINKSERVER#——基本服务地址
- #LINKCUSCODE#——服务端设置的客户代码

18 上下文菜单菜单激活按钮

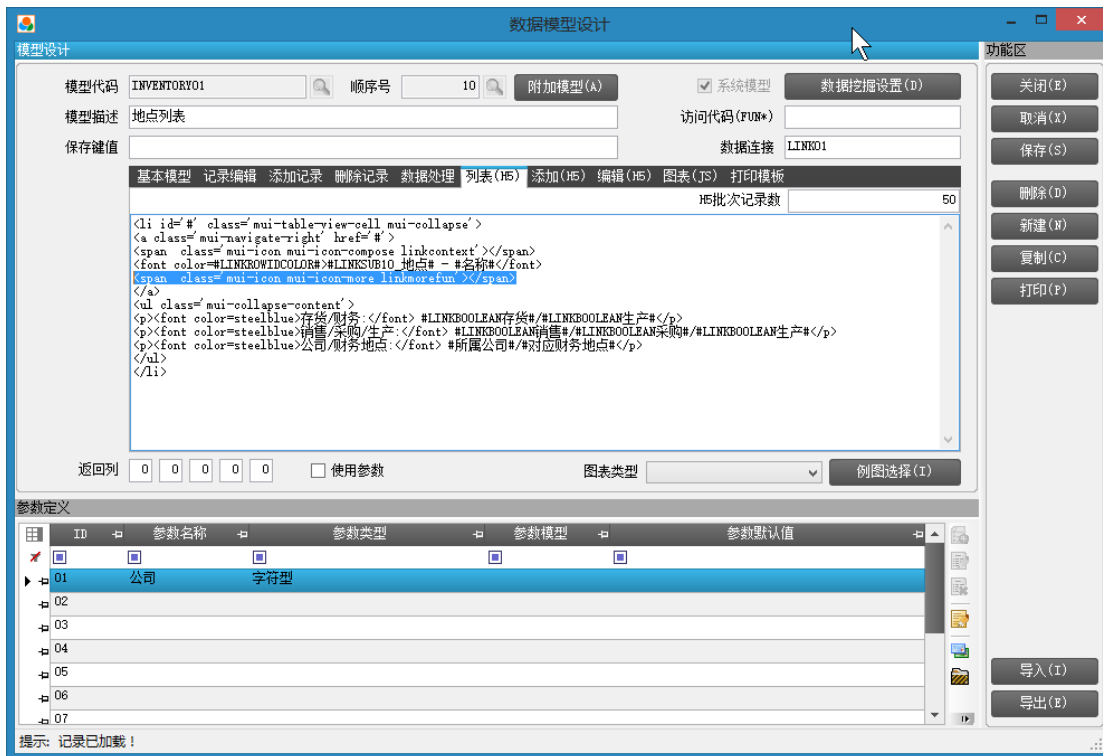
- 在列表内容中，可以通过定义一个 class=linkcontext 的按钮，用于实现单击激活上下文菜单
- 默认上下文菜单的激活方式是通过“长按”操作实现的



- 在模型的内容中嵌入如下内容
 - `<span class='mui-icon mui-icon-compose linkcontext'></span>`
 - 即可实现单击这个 icon 时，实现“长按”的行为
- 这里，请确保这个图标的 class 中包含 linkcontext
- 图标的样式可选，图标在内容中出现的位置是可以根据用户的需求定义的
- 同样，用户也可以使用其他的按钮等方式将 class 中包含 linkcontext 实现激活上下文菜单功能

19 附加功能菜单激活按钮

- 在列表内容中，可以通过定义一个 `class=linkmorefun` 的按钮，用于实现单击激活附加功能菜单
- 默认上下文菜单的激活方式是通过内容“右下角的按钮”操作实现的



- 在模型的内容中嵌入如下内容

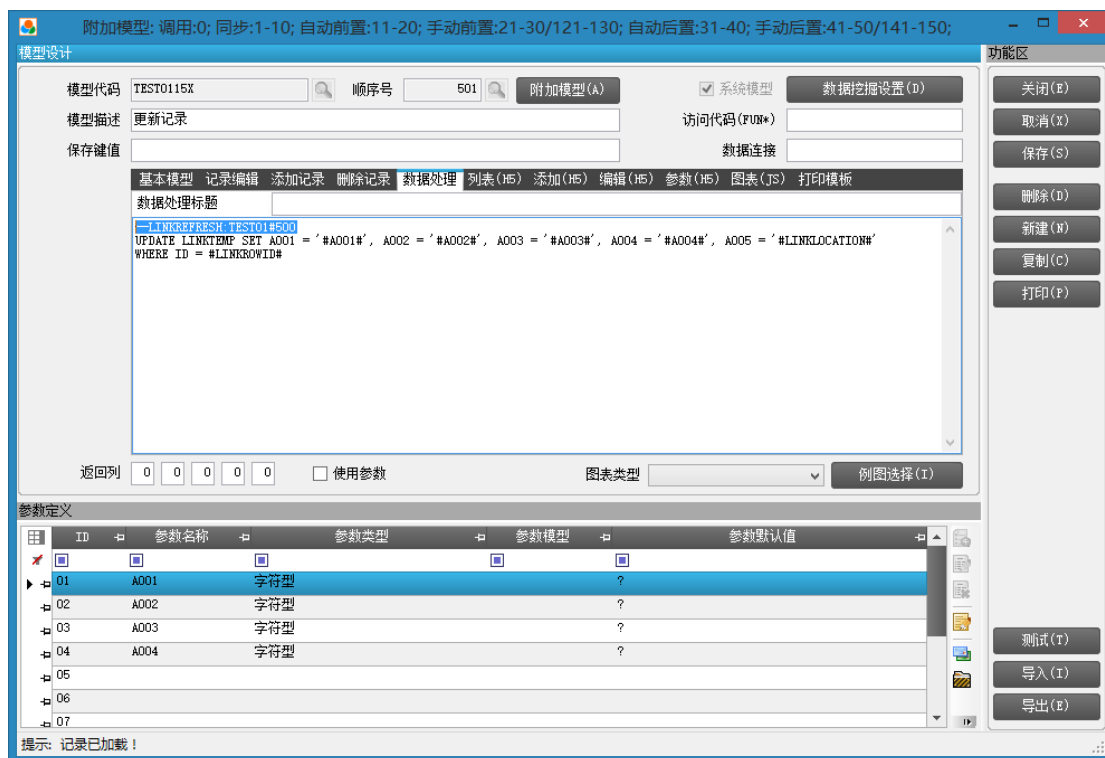
```
<span class='mui-icon mui-icon-more linkmorefun'></span>
```

■ 即可实现单击这个 icon 时，实现右下角按钮“附加功能”的行为

- 这里，请确保这个图标的 `class` 中包含 `linkmorefun`
- 图标的样式可选，图标在内容中出现的位置可选
- 同样，用户也可以使用其他的按钮等方式将 `class` 中包含 `linkmorefun` 实现激活附加功能

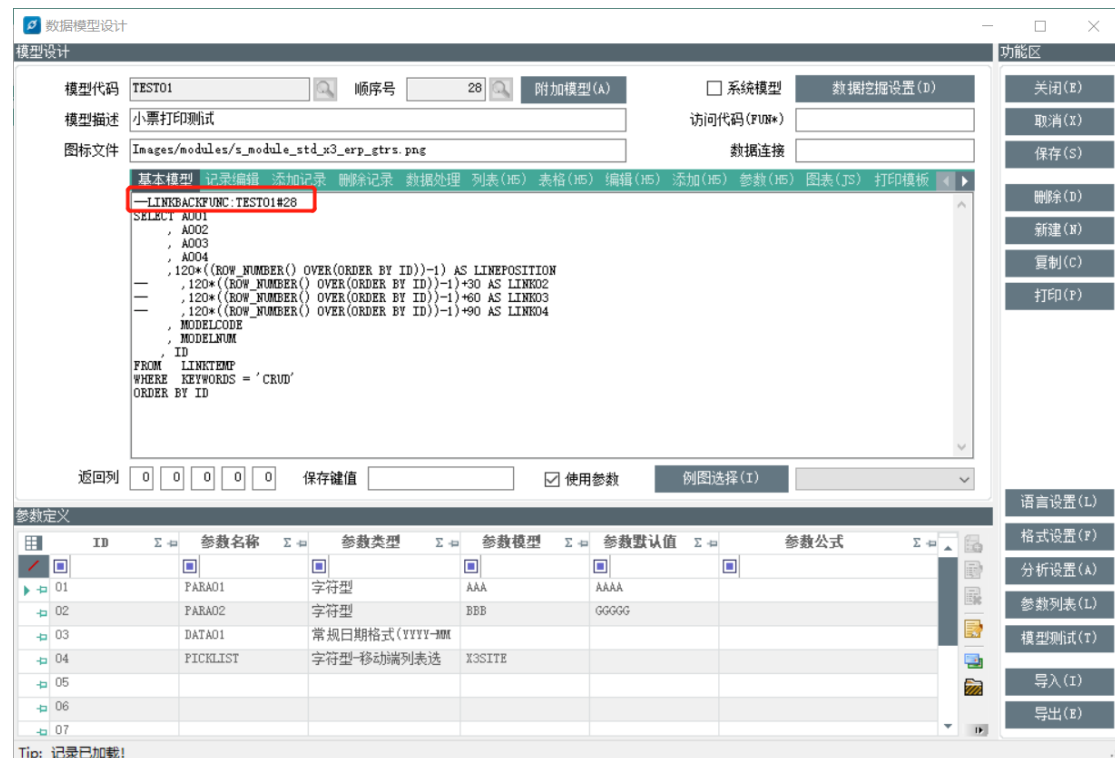
20 附加操作后返回指定模型

- 有些时候，根据功能要求，需要在执行某些操作后返回用户指定的页面
- 这时，可以在模型的脚本中通过如下方式设置需要“返回到的页面”



- 如图所示，这是模型“TEST01#15”的一个附加模型
- 按照正常的执行顺序，该模型的操作执行完成以后，会返回到 TEST01#15 模型的数据
- 在这个例子的脚本中，第一行按照如下方式定义
 - --LINKREFRESH:TEST01#500
 - --LINKREFRESH, 标记该模型操作后会重新加载数据
 - TEST01#500, 标记需要返回到 TEST01#500 模型，并显示 TEST01#500 模型的数据，而不是返回到默认的 TEST01#15

21 打印结束后跳转到某个指定的功能页面



数据模型设计

模型设计

模型代码: TEST01 顺序号: 28 附加模型(A) ☐ 系统模型 数据挖掘设置(D)

模型描述: 小票打印测试 访问代码(FUN*):

图标文件: Images/modules/s_module_std_x3_erp_gtrs.png 数据连接:

基本模型 记录/编辑 添加记录 删除记录 数据处理 列表(H5) 表格(H5) 编辑(H5) 添加(H5) 参数(H5) 图表(JS) 打印模板

--LINKBACKFUNC: TEST01#28

```
SELECT A001
, A002
, A003
, A004
, 120*(ROW_NUMBER() OVER(ORDER BY ID))-1 AS LINEPOSITION
, 120*(ROW_NUMBER() OVER(ORDER BY ID))-1+30 AS LINK02
, 120*(ROW_NUMBER() OVER(ORDER BY ID))-1+60 AS LINK03
, 120*(ROW_NUMBER() OVER(ORDER BY ID))-1+90 AS LINK04
, MODELCODE
, MODELNUM
, ID
FROM LINKTEMP
WHERE KEYWORDS = 'CRUD'
ORDER BY ID
```

返回列: 0 0 0 0 0 保存键值: ☒ 使用参数 例图选择(I)

参数定义

ID	参数名称	参数类型	参数模型	参数默认值	参数公式
01	PARA01	字符型	AAA	AAAA	
02	PARA02	字符型	BBB	GGGGG	
03	DATA01	常规日期格式(YYYY-MM)			
04	PICKLIST	字符型-移动端列表	X3SITE		
05					
06					
07					

Tip: 记录已加载!

- 按照正常的执行顺序，该模型的打印操作执行完成以后，会返回到 TEST01#28 模型查询列表数据
- 在这个例子的脚本中，第一行按照如下方式定义
 - --LINKBACKFUNC: TEST01#28
 - --LINKBACKFUNC，标记该模型操作后会重新加载数据
 - TEST01#28，标记需要返回到 TEST01#28 模型，如果该模型有参数，则会返回参数页面



22 多选操作

- 对于模型查询的结果，当出现多条记录的时候，可以使用多选操作功能。
- 对于数据的多选操作，可以对以下功能生效
 - 删除记录
 - 编辑记录
 - 预处理功能（附加模型 21-30）
 - 后处理功能（附加模型 41-50）
- 当用户选择了全部或者部分数据之后，执行上述各项操作时，系统自动按照选中的记录循环执行相应的操作

23 手写签名功能

- 模型实例

- 在添加模型上添加如下图所示，关键部分为 **id="linksignopen"**

```
<div class="mui-input-row">
  <label>签名</label>
  <button type="button" id="linksignopen" class="mui-btn mui-btn-outlined" style="width:100px">打开画布</button>
</div>
```

- 在编辑模型上加入，关键部分为 **sign-src="#LINKSIG#"**

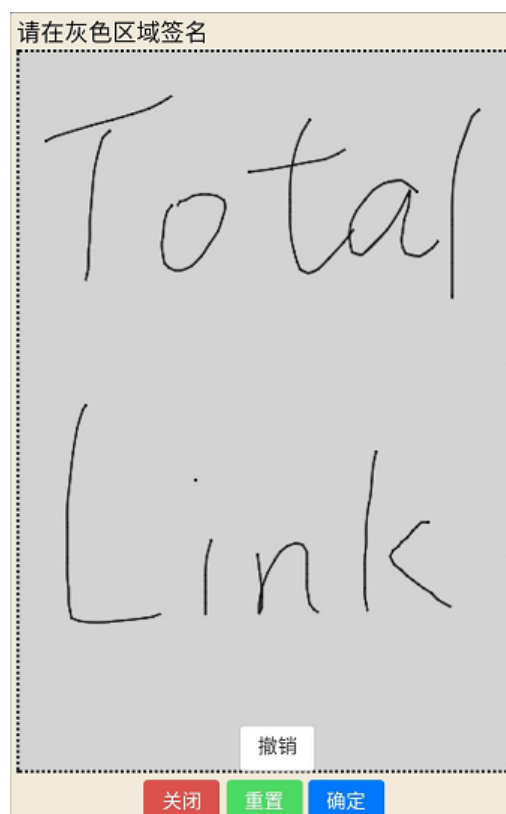
```
<div class="mui-input-row">
  <label>签名</label>
  <button type="button" id="linksignopen" sign-src="#LINKSIG#" class="mui-btn mui-btn-outlined" style="width:100px">打
  开画布</button>
</div>
```

- 效果实例

在“系统功能”下的“反馈记录”里面点击“如图”的“打开画布”按钮即可手写
签名

打开画布

如下图所示：



24 使用录音信息

● 模型实例

- 在添加模型上添加如下图所示，关键部分为 **id="linkaudio"**

```
<div class="mui-input-row">
  <label>录音</label>
  <button type="button" id="linkaudio" class="mui-btn mui-btn-outlined" style="width:100px">按住录音</button>
  <a class="linkrec" rec-src=""></a>
</div>
```

- 在编辑模型上加入，关键的两个地方的 **rec-src="#LINKREC#"**

```
<div class="mui-input-row">
  <label>录音</label>
  <button type="button" id="linkaudio" rec-src="#LINKREC#" class="mui-btn mui-btn-outlined" style="width:100px">按住录音</button>
  <a class="linkrec" rec-src="#LINKREC#">点击试听</a>
</div>
```

■ 效果实例

在“系统功能”下的“反馈记录”里面按钮“如图”的“录音”按钮即可录音，可松开保存或者上滑取消保存等操作

录音

按住录音

录音时

备注	请输入备注
签名	打开相机
录音	按住录音 手指上划，取消发送
FLAG01	☐
FLAG02	☐

可执行试听等操作

录音

按住录音

点击试听

25 使用服务端 JavaScript 脚本

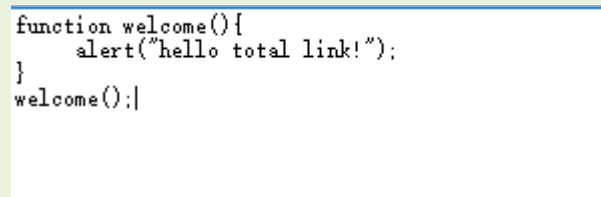
- 模型实例

- 在相应的模型上的“JavaScript”选项下编写 JavaScript 代码，如图



```
function welcome(){
    alert("hello total link!");
}
```

- 如果需要立即执行，则直接调用相应函数即可，如

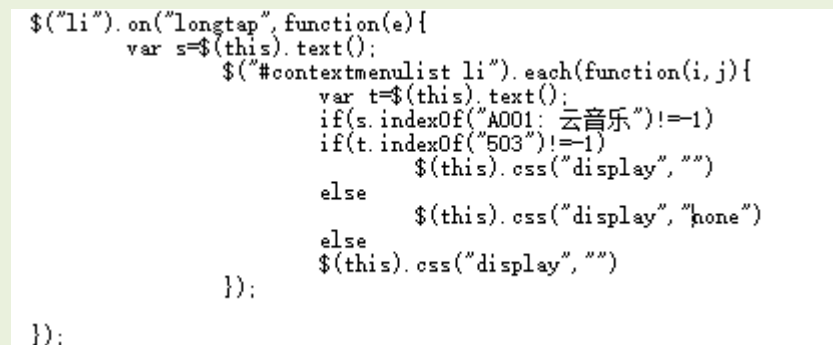


```
function welcome(){
    alert("hello total link!");
}
welcome();
```

- 使用技巧

- 可用来处理长按等操作弹出的功能项目

- 使用方法加入下列代码即可（ps：下列方法适用于长按操作）



```
$( "li" ).on( "longtap", function( e ) {
    var s = $( this ).text();
    $( "#contextmenulist li" ).each( function( i, j ) {
        var t = $( this ).text();
        if( s.indexOf( "A001: 云音乐" ) !== -1 )
            if( t.indexOf( "503" ) !== -1 )
                $( this ).css( "display", "" )
            else
                $( this ).css( "display", "none" )
        else
            $( this ).css( "display", "" )
    } );
} );
```

- 详细解释

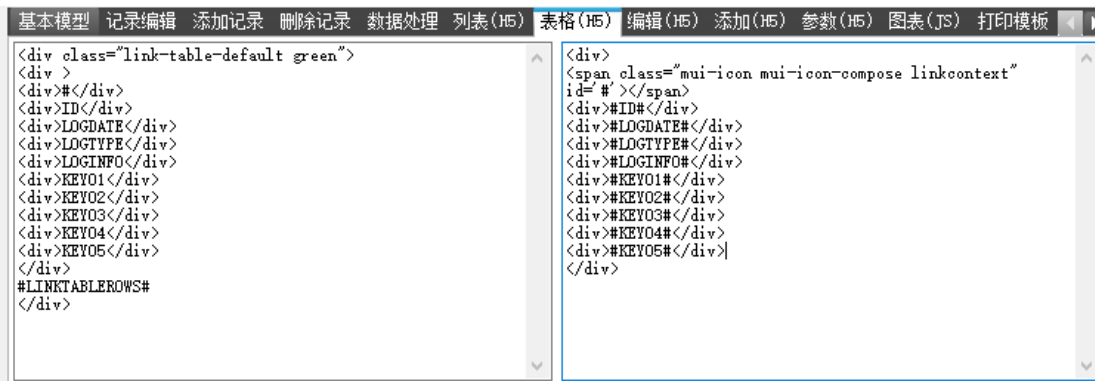
\$("li")可改为其他的对象，Longtap 可改为 tap 等其他操作，“indexOf('xxxx')” xxxx 可为任意需要的值，根据实际需求来编写。第一个 indexOf 里面的值是当前点击或长按行的内容判断，第二个 indexOf 的内容是显示指定的功能项按钮。

原理：通过循环遍历对应的要弹出的对象，根据需求进行隐藏或者显示并赋值。

26 使用表格展现模式

● 模型实例

- 在相应的模型上的“表格”选项下编写如下代码，如图



■ 详细说明

1. 在“表格”下有两个部分，左边的是表格的标题即表头，右边的是数据内容
2. 左边的部分主要是用来生成表头，即表格中的第一行数据,右边的部分是生成第二行到 N 行的数据，值得注意的是，只需要填写第二行数据，用#字段名#来获取服务端的数据。然后在左边的表头后面加入#LINKTABLEROWS#作为标识，系统即会把右边的表格行内容循环遍历输出
3. 表格样式 `class="link-table-default"` 为默认的样式（红色表头），还可以在后面加 `green`（绿色）或 `blue`（蓝色）来改变表头的颜色。

即 `class="link-table-default green"`

4. 默认标准的格式如下图，需要在表格标签里加 `class="link-table-default"`

```
<div class="link-table-default"> //表格标签
  <div> //行标签
  <div></div><div></div> //列标签
</div>
<div> //行标签
<div></div><div></div> //列标签
</div>
</div>
```

5. 当然也可以用标准的 HTML5 表格的标签填写，我们也提供一个默认的样式，即只需要在 `table` 标签里加入表格样式 `class="link-table"`。如下图

```
<table border="1" class="link-table"> //表格标签
  <tr> //行标签
    <th></th><th></th> //表头
  </tr>
  <tr> //行标签
    <td></td><td></td> //列标签
  </tr>
</table>
```

该效果表头不固定

- 效果图

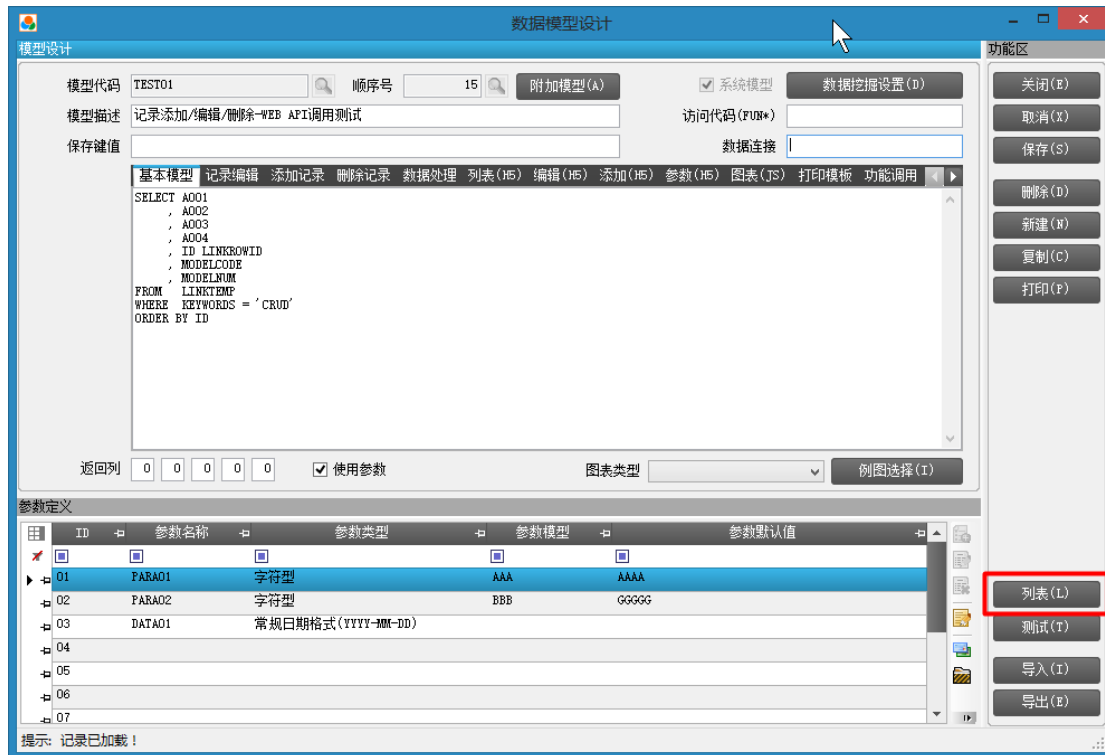
1.默认效果 link-table-default

2.table 标签 link-table 效果

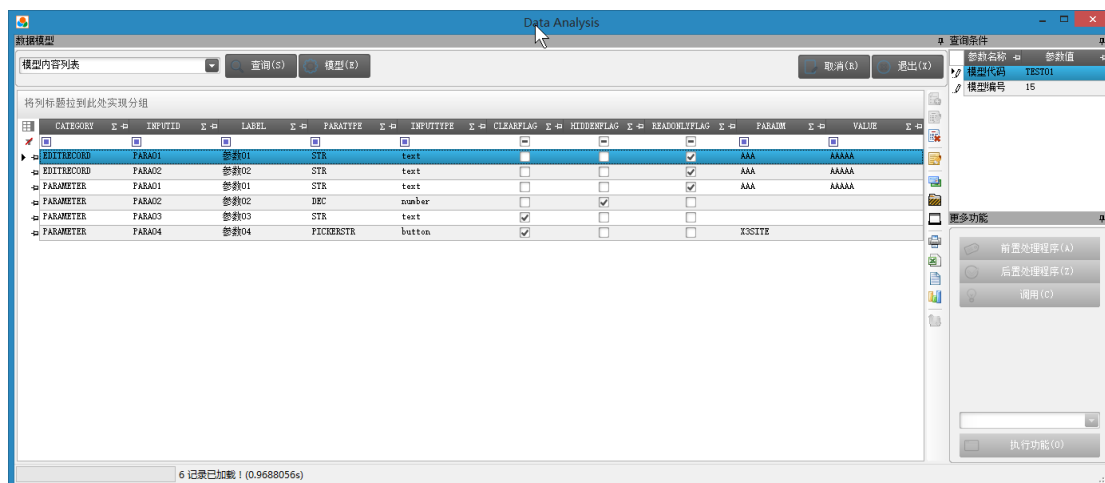
表格信息测试			
#	ID	LOGDATE	LOGTYPE
☒	789154	2018/1/17 13:08:00	Link Web API
☒	789155	2018/1/17 13:08:01	ModelDataSetResult
☒	789156	2018/1/17 13:08:03	Link Web API
☒	789157	2018/1/17 13:08:06	Link Web API
☒	789158	2018/1/17 13:08:06	ModelDataSetResult

表格信息测试			
#	ID	LOGDATE	LOGTYPE
☒	789154	2018/1/17 13:08:00	Link Web API
☒	789155	2018/1/17 13:08:01	ModelDataSetResult
☒	789156	2018/1/17 13:08:03	Link Web API
☒	789157	2018/1/17 13:08:06	Link Web API
☒	789158	2018/1/17	ModelDataSetResult

27 使用模型参数列表



- 模型设计时，可以使用“列表”功能定义移动端页面中需要使用到的数据列



- 可以为每列数据定义基本属性，包括
 - CATEGORY: 定义列表的使用目标，有三个选项，可以是
 - EDITRECORD, 此类型的列表用于生成记录编辑的内容
 - NEWRECORD, 此类型的列表用于生成添加记录的内容
 - PARAMETER, 此类型的列表用于生成内容查询的参数页面或者附件功能参数页面

- INPUTID, 页面数据列表中每列数据的 ID, 用于生成脚本中的 id 信息, 在模型的处理脚本中, 引用数据需要按照#id#的方式引用
- LABEL, 用于生成页面中的 label 信息
- PARATYPE, 内容列的数据类型, 通过列表选择, 包括如下几个选项

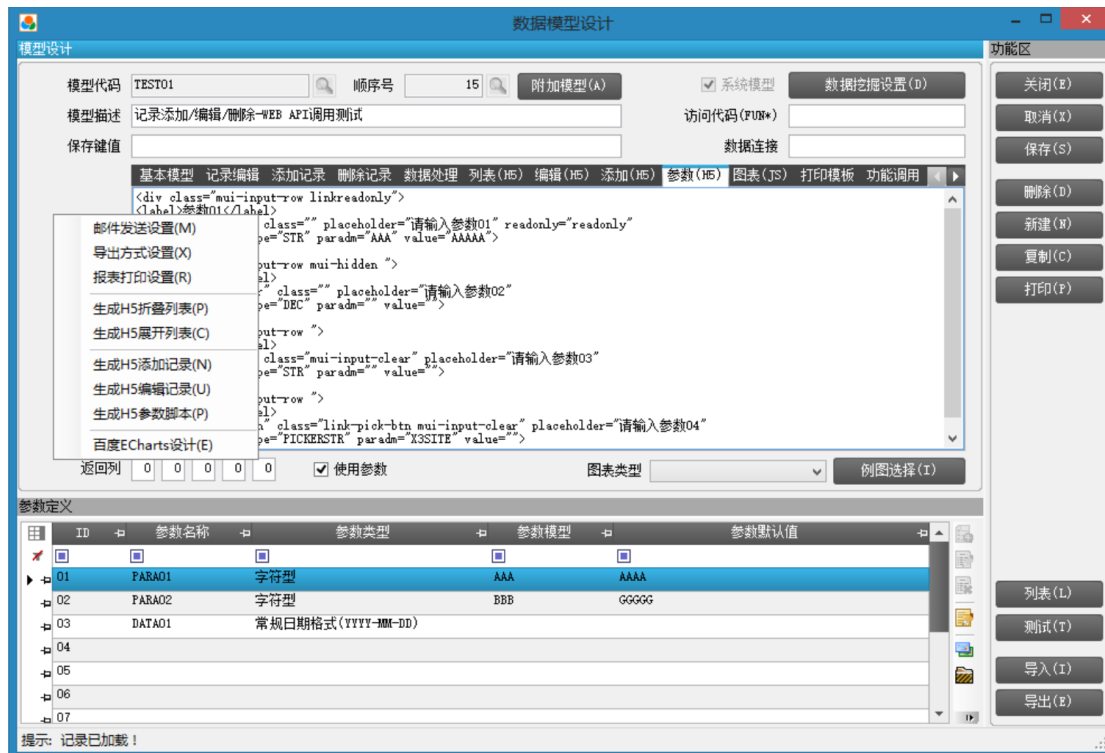


◆ 此处列表选择, 包括扩展, 包括:

- 多行文本
- 地理位置定位
- 拍照或图片选择上传
- 文件上传
- 手写签名
- 录音

- INPUTTYPE, 用于生成页面中的 type 信息
- CLEARFLAG, 是否“允许清除”标识, 用于生成页面中的 class 信息, 该标识为“真”的时候, class 中包含“mui-input-clear”
- HIDDENFLAG, 内容“隐藏”标识, 表示该列的内容是否隐藏, 用于生成页面中的 class 信息, 该标识为“真”的时候, class 中包含“mui-hidden”
- READONLYFLAG, 是否“只读信息”标识, 用于生成页面中的 readonly 信息, 该标识为“真”的时候, class 中包含“linkreadonly”, 用于使用对应 CSS 属性, 并且有 readonly="readonly"选项
- PARADM, 参数开窗选择模型代码, 对于 PARATYPE=PICKERSTR 时使用, 此时设置 INPUTTYPE=button, 可以使用 PARADM 定义的模型实现在移动端的列表选择

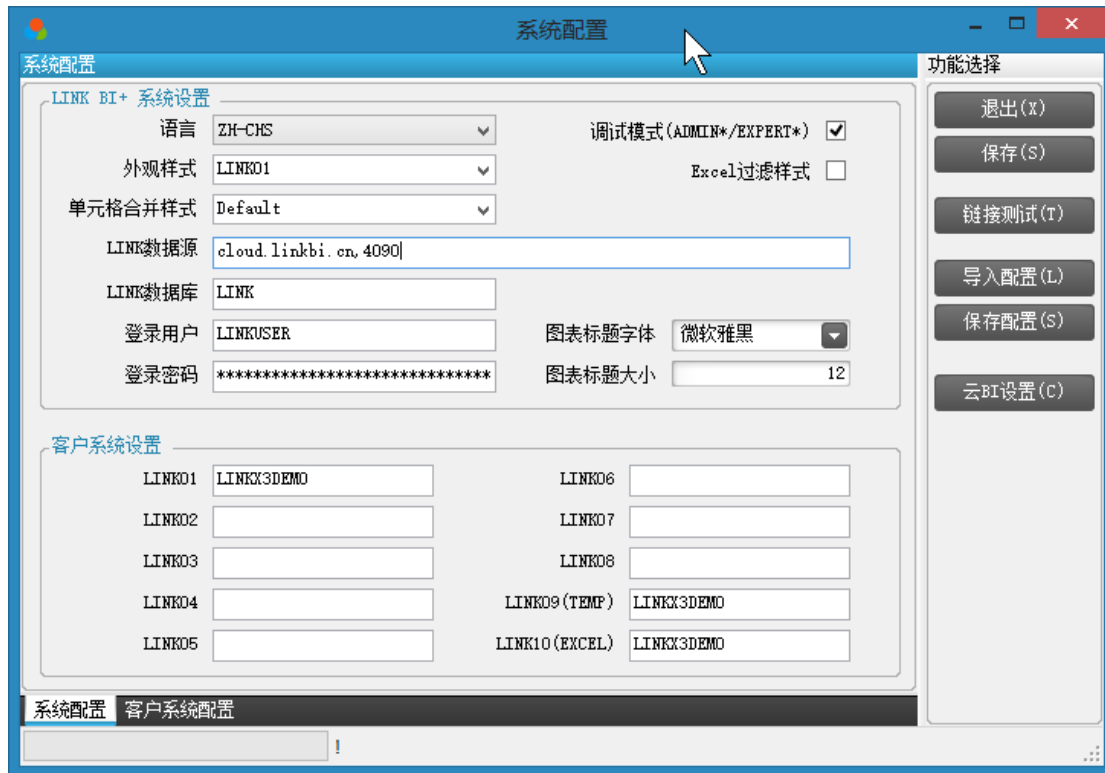
- VALUE，值来源，用于设置生成页面内容时该列的值（或默认值）
- 对于上图中的列表定义，可以自动自动生成脚本的功能，生成的脚本如下：



```
<div class="mui-input-row linkreadonly">
<label>参数 01</label>
<input type="text" class="" placeholder="请输入参数 01" readonly="readonly"
id="PARA01" paratype="STR" paradm="AAA" value="AAAAA">
</div>
<div class="mui-input-row mui-hidden ">
<label>参数 02</label>
<input type="number" class="" placeholder="请输入参数 02"
id="PARA02" paratype="DEC" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>参数 03</label>
<input type="text" class="mui-input-clear" placeholder="请输入参数 03"
id="PARA03" paratype="STR" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>参数 04</label>
<input type="button" class="link-pick-btn mui-input-clear" placeholder="请输入参数
04" id="PARA04" paratype="PICKERSTR" paradm="X3SITE" value=""></div>
```

- 在所生成的内容的基础上，用户可以根据需要修改，嵌入更多专业的内容即可。

28 启用调试模式



- 在系统设计过程中，可以使用系统提供的各种脚本调试功能，用以检查根据数据产生的脚本的正确性。
- 通过系统主菜单的“设置-系统设置”打开相应功能
- 勾选“调试模式（ADMIN*/EXPERT*）”即可启动调试模式
- 该模式只针对“ADMIN*/EXPERT*”组的用户有效
- 某些 SQL 脚本的调试信息只在 Traditional 模式下有效，需注意 LINK 数据源配置模式
- 在调试模式下，系统对 SQL 脚本等的运行信息可以随时显示给开发人员
- 启动调试模式，也同时启用了 H5、JS 等脚本的测试检查功能

29 接口调用方法

- 服务端的 WEB API 服务也可以接受其他的应用系统调用
- 如果需要通过 WEB API 获取数据，需要首先在系统中建立对应的模型,并使用在系统中正常注册的 ID 执行调用操作
- 典型的调用方法如下请参考下面的例子

POST api/DataModel/linkDMDatasetResult

根据模型及参数获取数据

Request Information

Parameters

Name	Description	Additional information
p	数据模型及参数列表	Define this parameter in the request body.

Request body formats

application/json, text/json

Sample:

```
{
  "loginID": "sample string 1",
  "par": {
    "dmCode": "sample string 1",
    "dmNum": 2,
    "Para": [
      "sample string 1",
      "sample string 2",
      "sample string 3"
    ]
  }
}
```

- 服务地址:

api/DataModel/linkDMDatasetResult

- 方法: POST

- 调用参数:

```
{
  "loginID": "注册用户的 ID",
  "par": {
    "dmCode": "模型代码",
    "dmNum": 模型编号,
    "Para": [
      "参数 1",
      "参数 2",
      "参数 3"
    ]
  }
}
```

■ 返回结果:

```
{
  "isSuccess": "是否成功的标识",
  "message": "执行信息",
  "dataType": "返回的结果类型，通常为 Dataset",
  "data": {JSON 格式的数据，通常为 Dataset 的内容}
}
```

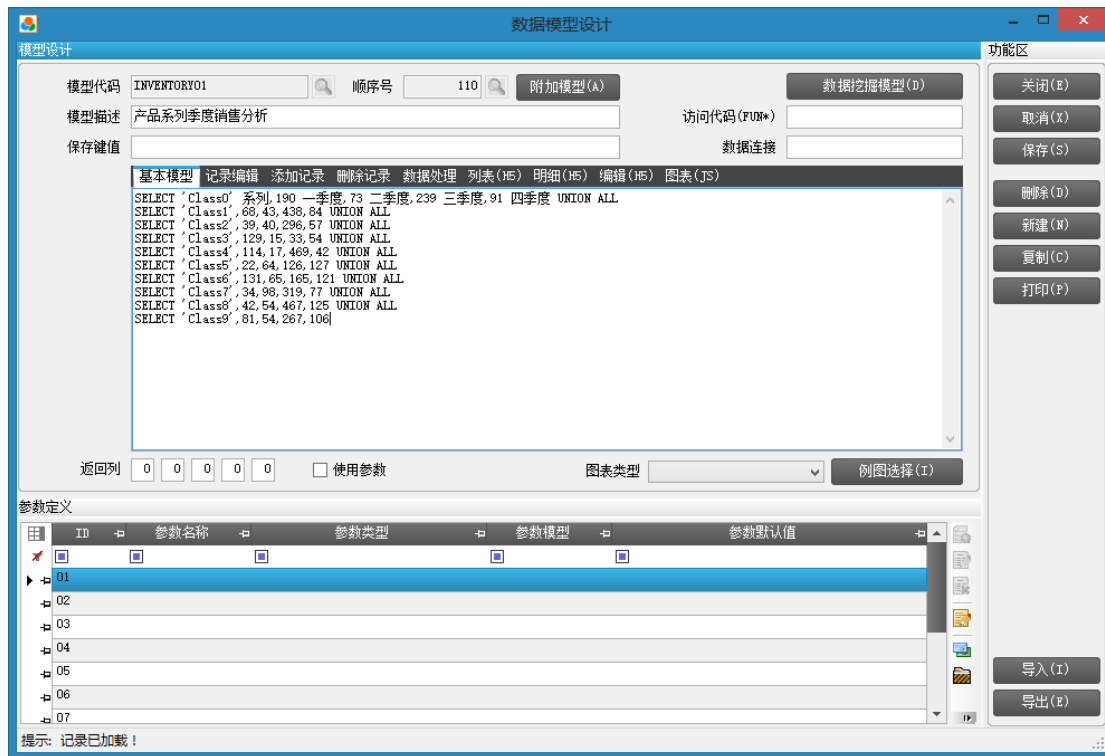
■ 如下为通过 Javascript 调用的例子

```
var linkurl = 'http://47.96.137.47:8088/api/DataModel/linkDMDatasetResult';
var parajson = {
  loginID: LinkID,
  par: {
    dmCode: '_LINKSYS_',
    dmNum: 20,
    Para: [
      glocaleUser,
      glocaleMenuGroup
    ]
  }
}
mui.ajax(linkurl, {
  data: parajson,
  async: false,
  datatype: 'json',
  type: 'post',
  timeout: 30000,
  success: function(r) {
    if(r.isSuccess == 'true') {
      var t = r.data.Table;
      var finallist = geneMenuList(t);
    } else {
    };
  },
  error: function(r) {
  }
});
```

30 调试实例

以下以一个实例介绍系统的各种不同脚本的调试功能。

30.1 基本模型脚本调试



- 创建一个基本模型，脚本如下：

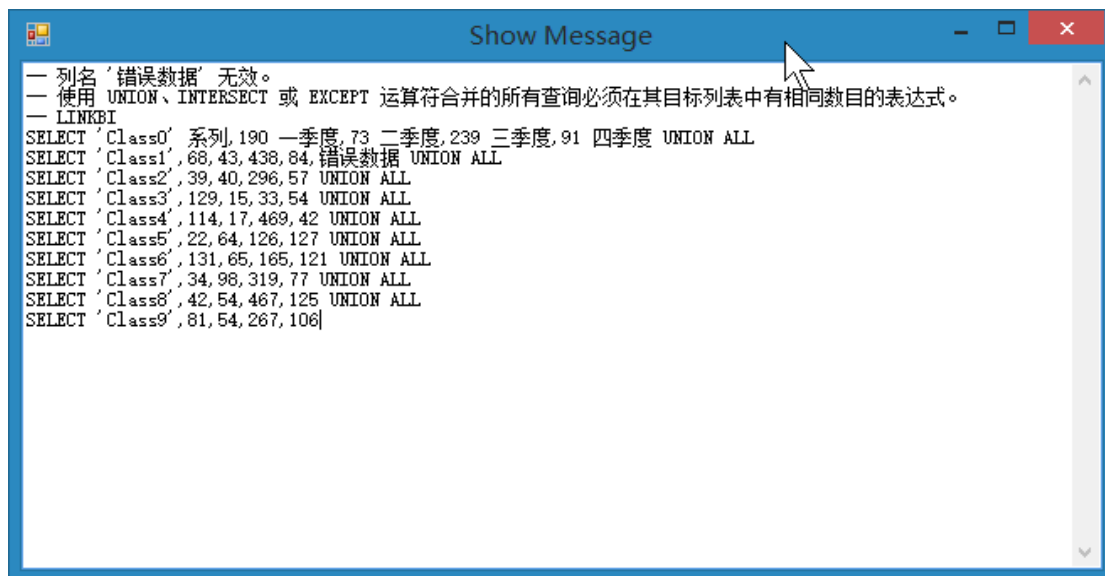
```
SELECT 'Class0' 系列,190 一季度,73 二季度,239 三季度,91 四季度 UNION ALL
SELECT 'Class1',68,43,438,84 UNION ALL
SELECT 'Class2',39,40,296,57 UNION ALL
SELECT 'Class3',129,15,33,54 UNION ALL
SELECT 'Class4',114,17,469,42 UNION ALL
SELECT 'Class5',22,64,126,127 UNION ALL
SELECT 'Class6',131,65,165,121 UNION ALL
SELECT 'Class7',34,98,319,77 UNION ALL
SELECT 'Class8',42,54,467,125 UNION ALL
SELECT 'Class9',81,54,267,106
```

- 在数据脚本可以正常执行的情况下，系统正常运行，不返回错误信息
- 对脚本做修改如下：

```
SELECT 'Class0' 系列,190 一季度,73 二季度,239 三季度,91 四季度 UNION ALL
SELECT 'Class1',68,43,438,84,错误数据 UNION ALL
```

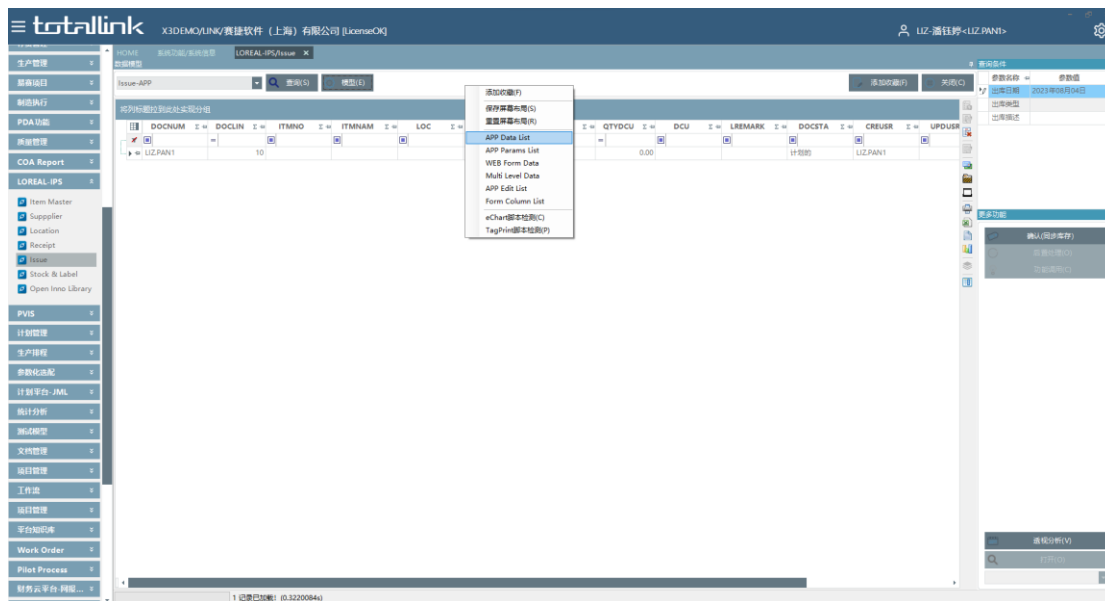
```
SELECT 'Class2',39,40,296,57 UNION ALL
SELECT 'Class3',129,15,33,54 UNION ALL
SELECT 'Class4',114,17,469,42 UNION ALL
SELECT 'Class5',22,64,126,127 UNION ALL
SELECT 'Class6',131,65,165,121 UNION ALL
SELECT 'Class7',34,98,319,77 UNION ALL
SELECT 'Class8',42,54,467,125 UNION ALL
SELECT 'Class9',81,54,267,106
```

- 重新执行模型的数据查询功能

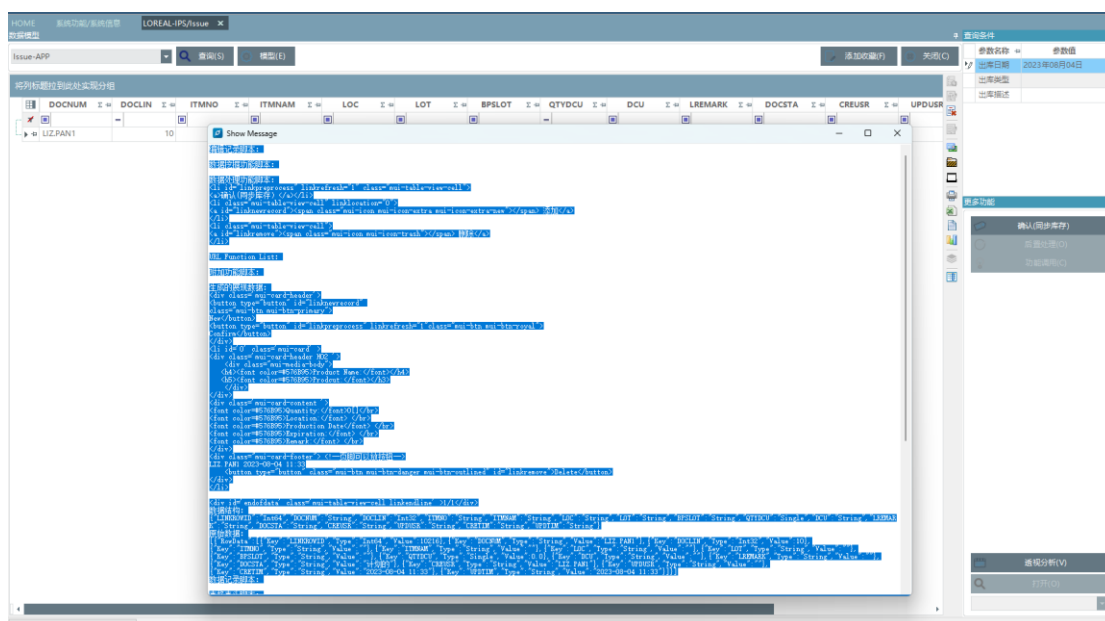


- 提供提示信息如图所示，经过数据处理的脚本完整的显示给用户
- 同时，服务器反馈的脚本执行错误信息以“—”标记为注释行，显示在脚本的前面部分
- 在脚本测试时，脚本中如果用的参数、全局变量、引用的表格数据等，都被替换为实际执行的数据，这样有利于用户发现错误
- 用户可以将脚本复制到相应的数据库服务器进行调试，完成后重新粘贴到系统中即可
- 同样，模型脚本中的各种数据操作都可以通过该模型进行调试，以便发现脚本的错误并修正

30.2 H5 脚本测试—APP Data List



- 完成模型设计后，可以通过“数据分析”功能的右键菜单找到对应的脚本测试功能



- 点击“APP Data List”，可以得到模型的各类脚本，包括：
 - 添加记录的脚本（仅在数据模型允许添加记录时生成）
 - 数据挖掘功能脚本（仅在数据模型有对应的数据挖掘功能时生成）
 - 前置后置处理功能脚本（仅在模型有对应的手动前置及后置处理功能时生成）
 - 附加功能脚本（仅在模型定义了附加功能时生成）
 - 生成的展现数据（默认以可折叠列表形式形成的数据脚本，如果有用户自定义的列表展现格式，则以用户定义的方式生成数据）
- 通过调试功能，可以看到生成的 H5 数据脚本如下：



编辑记录脚本:

数据挖掘功能脚本:

数据处理功能脚本:

```
<li id="linkpreprocess" linkrefresh="1" class="mui-table-view-cell">
<a>确认(同步库存) </a></li>
<li class="mui-table-view-cell" linklocation="0">
<a id="linknewrecord"><span class="mui-icon mui-icon-extra
mui-icon-extra-new"></span> 添加</a>
</li>
<li class="mui-table-view-cell">
<a id="linkremove"><span class="mui-icon mui-icon-trash"></span> 删除</a>
</li>
```

URL Function List:

附加功能脚本:

生成的展现数据:

```
<div class='mui-card-header'>
<button type="button" id="linknewrecord"
class="mui-btn mui-btn-primary">
New</button>
<button type="button" id="linkpreprocess" linkrefresh="1" class="mui-btn
mui-btn-royal">
Confirm</button>
</div>
<li id='0' class='mui-card'>
<div class='mui-card-header H02 '>
<div class="mui-media-body">
<h4><font color=#576B95>Product Name:</font></h4>
<h5><font color=#576B95>Prodcut:</font></h3>
</div>
</div>
<div class='mui-card-content '>
<font color=#576B95>Quantity:</font>0[]</br>
<font color=#576B95>Location:</font> </br>
<font color=#576B95>Production Date</font> </br>
<font color=#576B95>Expiration:</font> </br>
<font color=#576B95>Remark:</font> </br>
</div>
<div class="mui-card-footer"> <!--页脚可以放按钮-->
```



LIZ.PAN1 2023-08-04 11:33

```
<button type="button" class="mui-btn mui-btn-danger mui-btn-outlined"
id="linkremove">Delete</button>
</div>
</li>
```

```
<div id='endofdata' class='mui-table-view-cell linkendline' >1/1</div>
```

数据结构:

```
{"LINKROWID":"Int64","DOCNUM":"String","DOCLIN":"Int32","ITMNO":"String","ITMNA
M":"String","LOC":"String","LOT":"String","BPSLOT":"String","QTYDCU":"Single","
DCU":"String","LREMARK":"String","DOCSTA":"String","CREUSR":"String","UPDUSR":"
String","CRETIM":"String","UPDTIM":"String"}
```

原始数据:

```
[{"rowData":[{"Key":"LINKROWID","Type":"Int64","Value":10216}, {"Key":"DOCNUM",
Type":"String","Value":"LIZ.PAN1"}, {"Key":"DOCLIN","Type":"Int32","Value":10}, {
"Key":"ITMNO","Type":"String","Value":""}, {"Key":"ITMNAM","Type":"String","Valu
e":""}, {"Key":"LOC","Type":"String","Value":""}, {"Key":"LOT","Type":"String","V
alue":""}, {"Key":"BPSLOT","Type":"String","Value":""}, {"Key":"QTYDCU","Type":"S
ingle","Value":0.0}, {"Key":"DCU","Type":"String","Value":""}, {"Key":"LREMARK",
Type":"String","Value":""}, {"Key":"DOCSTA","Type":"String","Value":"计划的
"}, {"Key":"CREUSR","Type":"String","Value":"LIZ.PAN1"}, {"Key":"UPDUSR","Type":"
String","Value":""}, {"Key":"CRETIM","Type":"String","Value":"2023-08-04
11:33"}, {"Key":"UPDTIM","Type":"String","Value":"2023-08-04 11:33"}]]
```

数据记录脚本:

表格表头脚本:

```
<div class='mui-card-header'>
<button type="button" id="linknewrecord"
class="mui-btn mui-btn-primary">
New</button>
<button type="button" id="linkpreprocess" linkrefresh="1" class="mui-btn
mui-btn-royal">
Confirm</button>
</div>
```

#LINKTABLEROWS#

表格明细脚本:

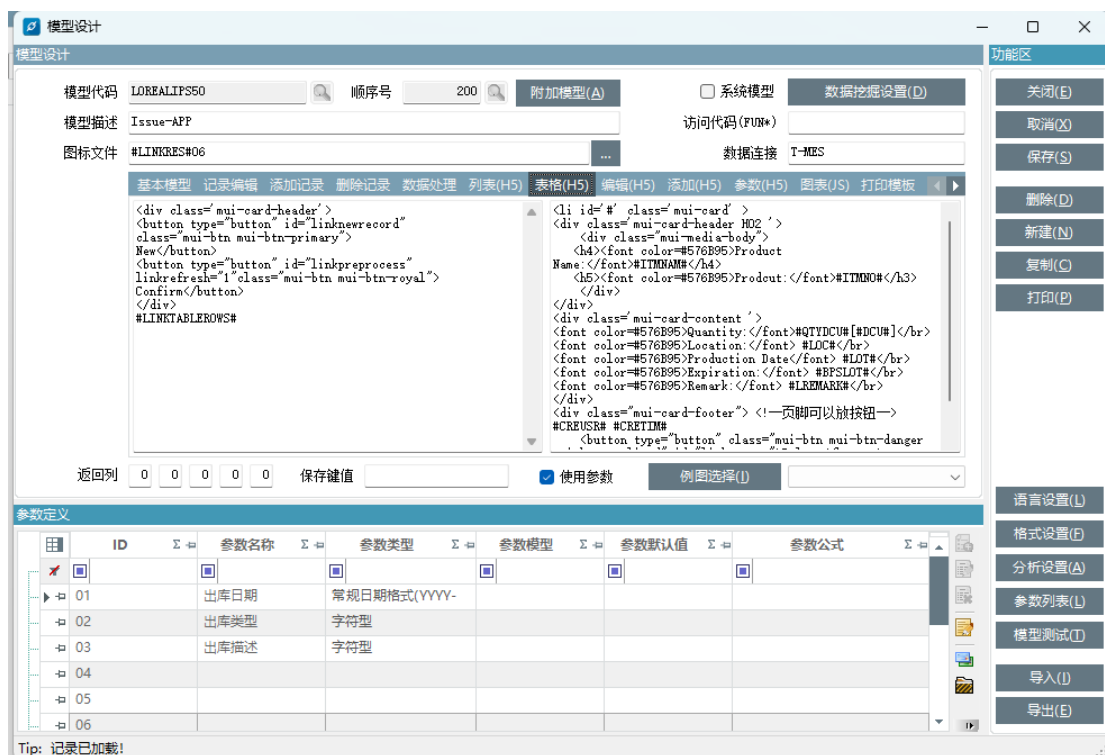
```
<li id='#' class='mui-card' >
<div class='mui-card-header H02 ' >
<div class="mui-media-body">
<h4><font color=#576B95>Product Name:</font>#ITMNAM#</h4>
```

```

<h5><font color=#576B95>Prodcut:</font>#ITMNO#</h3>
</div>
</div>
<div class='mui-card-content' >
<font color=#576B95>Quantity:</font>#QTYDCU#[#DCU#]</br>
<font color=#576B95>Location:</font> #LOC#</br>
<font color=#576B95>Production Date</font> #LOT#</br>
<font color=#576B95>Expiration:</font> #BPSLOT#</br>
<font color=#576B95>Remark:</font> #LREMARK#</br>
</div>
<div class="mui-card-footer"> <!--页脚可以放按钮-->
#CREUSR# #CRETIM#
    <button type="button" class="mui-btn mui-btn-danger mui-btn-outlined"
id="linkremove">Delete</button>
</div>

```

- 该模型的展现数据用户已经进行了自定义设置





- 相对应的内容，在移动端显示如下

13:17

Issue-APP

New

Confirm

Product Name:APP test 01

Prodcut:0000009

Quantity:1[UN]

Location:

Production Date 2022-09-15

Expiration: 2022-11-17

Remark:

LIZ.PAN1 2023-08-04 13:17

Delete

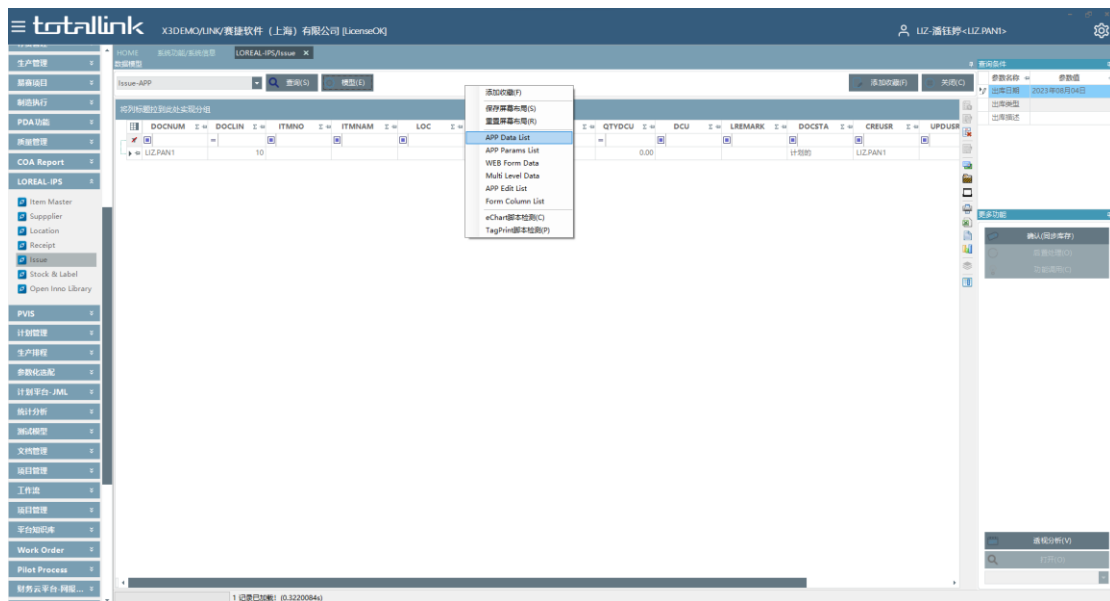
1/1

首页

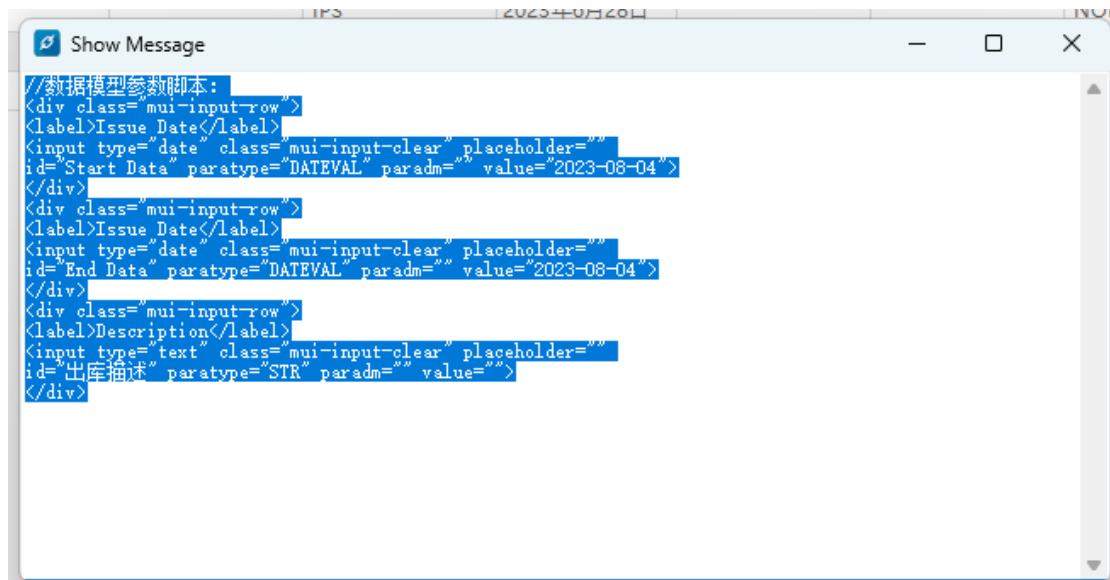
Link

功能

30.3 模型参数脚本测试—APP Params List



- 如图所示，该模型有三个参数，分别为：
 - 出库日期
 - 出库类型
 - 出库描述
- 通过点击“APP Params List”功能，可以得到在移动端显示参数列表时的脚本如下：



- 进行参数输入时，在移动端显示效果如果所示



The image shows a mobile application interface for 'Issue-APP'. At the top, there is a status bar with the time 13:28 and signal indicators. Below the status bar is a dark blue header with a back arrow, a funnel icon, and the text 'Issue-APP'. To the right of the header is a square icon with four arrows pointing outwards. Below the header is a light orange rectangular button with the text '扫描输入区'. Underneath the button are three input fields: 'Issue Date' with the value '2023年8月4日', 'Issue Type', and 'Description'. The 'Description' field is a large, empty text area.

- 作为附加参数时，会根据参数默认值是否为“?”判断是否需要显示出来
- 本例中为不显示，用<div class="mui-input-row mui-hidden">标记

30.4 记录添加、编辑及删除—APP Edit List

- 延续上述例子中的模型，在模型的定义中，该模型是允许添加、删除和编辑记录的

数据模型设计

模型设计

模型代码: PROJECT00 顺序号: 180 附加模型(A) 数据模型(D)

模型描述: 项目活动记录-按项目 访问代码(FUN*): 数据连接: #LINK#SAGECRM

保存键值:

基本模型 记录编辑 添加记录 删除记录 数据处理 列表(H5) 明细(H5) 编辑(H5) 图表(JS)

```

UPDATE ProjectMonitor
SET
  pmon_actdate = '@活动日期@'
  , pmon_acttype = '@活动类型@'
  , pmon_resource = '@资源@'
  , pmon_contact = '@联系人@'
  , pmon_content = '@LINKSFEECH活动内容@'
  , pmon_loguser = '@LOGIN_USER#'
  , pmon_logdate = GETDATE()
WHERE pmon_id = '#LINKROWID#'
  
```

返回列: 0 0 0 0 0 0 使用参数 图表类型: 例图选择(I)

参数定义

ID	参数名称	参数类型	参数模型	参数默认值
01	项目编码	字符型	X3CRMProjCODE	
02	项目名称	字符型-移动端语音输入		
03				
04				
05				
06				
07				

提示: 记录已加载!

功能区

关闭(E) 取消(X) 保存(S) 删除(D) 新建(N) 复制(C) 打印(P) 导入(I) 导出(E)

● 记录添加脚本、记录编辑脚本

```

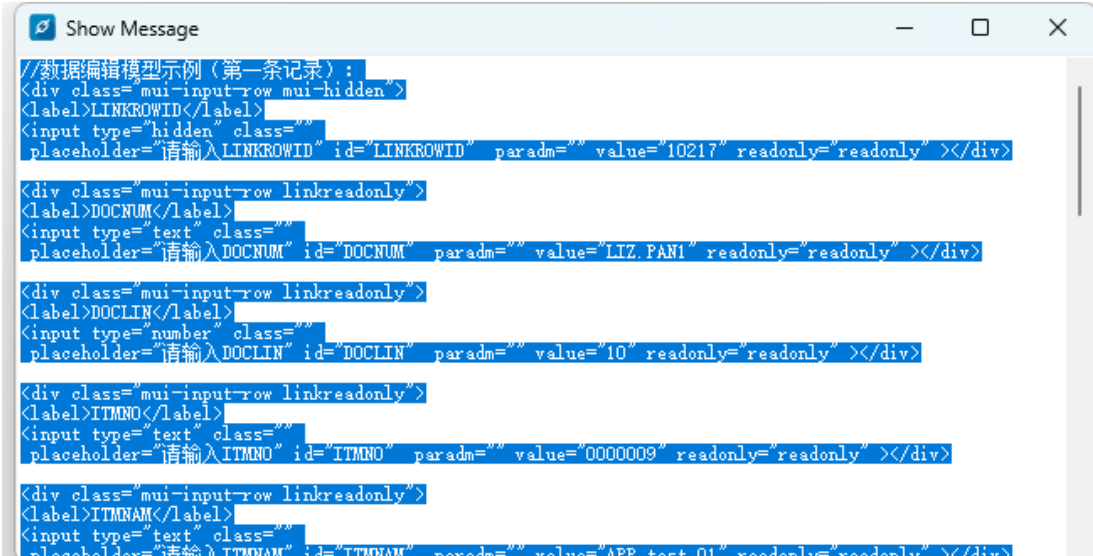
--LINKADDNEWROWS
--LINKAUTOREPLACEEMPTYCOLUMNS
MERGE
INTO
[STKISSH] D
  USING (
    SELECT
      N'#LOGIN_USER#' DOCNUM
    ,N'LOREAL' SITE
    ,N'{1}' DOCGRP
    ,N'IPS' DOCTYP
    ,'{0}' DOCDAT
    ,N'{2}' DOCDES
    ,N'@ REMARK @' REMARK
  ) S ON
(S.[DOCNUM] = D.[DOCNUM])
WHEN MATCHED THEN
UPDATE
SET
  UPDUSR='#LOGIN_USER#'
  ,UPDTIM=GETDATE()
WHEN NOT MATCHED THEN
INSERT
  
```

```
(DOCNUM
,SITE
,DOCGRP
,DOCTYP
,DOCDAT
,DOCDES
,REMARK
)
VALUES (
    S.DOCNUM
,S.SITE
,S.DOCGRP
,S.DOCTYP
,S.DOCDAT
,S.DOCDES
,S.REMARK
);
```

- 记录删除脚本

```
DELETE FROM STKISS WHERE ID=#LINKROWID#;
```

- 因此，在 PC 及移动端，可以使用记录的“APP Edit List”功能
- 记录编辑模型比较复杂，通过调试功能可以得到脚本如下



```
//数据编辑模型示例（第一条记录）：
<div class="mui-input-row mui-hidden">
<label>LINKROWID</label>
<input type="hidden" class=""
placeholder="请输入 LINKROWID" id="LINKROWID" param="" value="10217" readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>DOCNUM</label>
<input type="text" class=""
placeholder="请输入 DOCNUM" id="DOCNUM" param="" value="LIZ PANI" readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>DOCLIN</label>
<input type="number" class=""
placeholder="请输入 DOCLIN" id="DOCLIN" param="" value="10" readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>ITMNO</label>
<input type="text" class=""
placeholder="请输入 ITMNO" id="ITMNO" param="" value="0000009" readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>ITMNOAM</label>
<input type="text" class=""
placeholder="请输入 ITMNOAM" id="ITMNOAM" param="" value="APP test 01" readonly="readonly" ></div>
```

- 数据编辑模型脚本实例（使用第一条数据记录）

```
//数据编辑模型示例（第一条记录）：
<div class="mui-input-row mui-hidden">
<label>LINKROWID</label>
<input type="hidden" class=""
```



```
placeholder="请输入 LINKROWID" id="LINKROWID" paradm="" value="10217"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>DOCNUM</label>
<input type="text" class=""
placeholder="请输入 DOCNUM" id="DOCNUM" paradm="" value="LIZ.PAN1"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>DOCLIN</label>
<input type="number" class=""
placeholder="请输入 DOCLIN" id="DOCLIN" paradm="" value="10"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>ITMNO</label>
<input type="text" class=""
placeholder="请输入 ITMNO" id="ITMNO" paradm="" value="0000009"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>ITMNAM</label>
<input type="text" class=""
placeholder="请输入 ITMNAM" id="ITMNAM" paradm="" value="APP test 01"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>LOC</label>
<input type="text" class=""
placeholder="请输入 LOC" id="LOC" paradm="" value=""
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>LOT</label>
<input type="text" class=""
placeholder="请输入 LOT" id="LOT" paradm="" value="2022-09-15"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>BPSLOT</label>
<input type="text" class=""
placeholder="请输入 BPSLOT" id="BPSLOT" paradm="" value="2022-11-17"
```

```
readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>QTYDCU</label>
<input type="text" class=""
placeholder=" 请 输 入 QTYDCU" id="QTYDCU" paradm="" value="1"
readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>DCU</label>
<input type="text" class=""
placeholder=" 请 输 入 DCU" id="DCU" paradm="" value="UN"
readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>LREMARK</label>
<input type="text" class=""
placeholder=" 请 输 入 LREMARK" id="LREMARK" paradm="" value=""
readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>DOCSTA</label>
<input type="text" class=""
placeholder=" 请 输 入 DOCSTA" id="DOCSTA" paradm="" value="计划的"
readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>CREUSR</label>
<input type="text" class=""
placeholder=" 请 输 入 CREUSR" id="CREUSR" paradm="" value="LIZ.PAN1"
readonly="readonly" ></div>

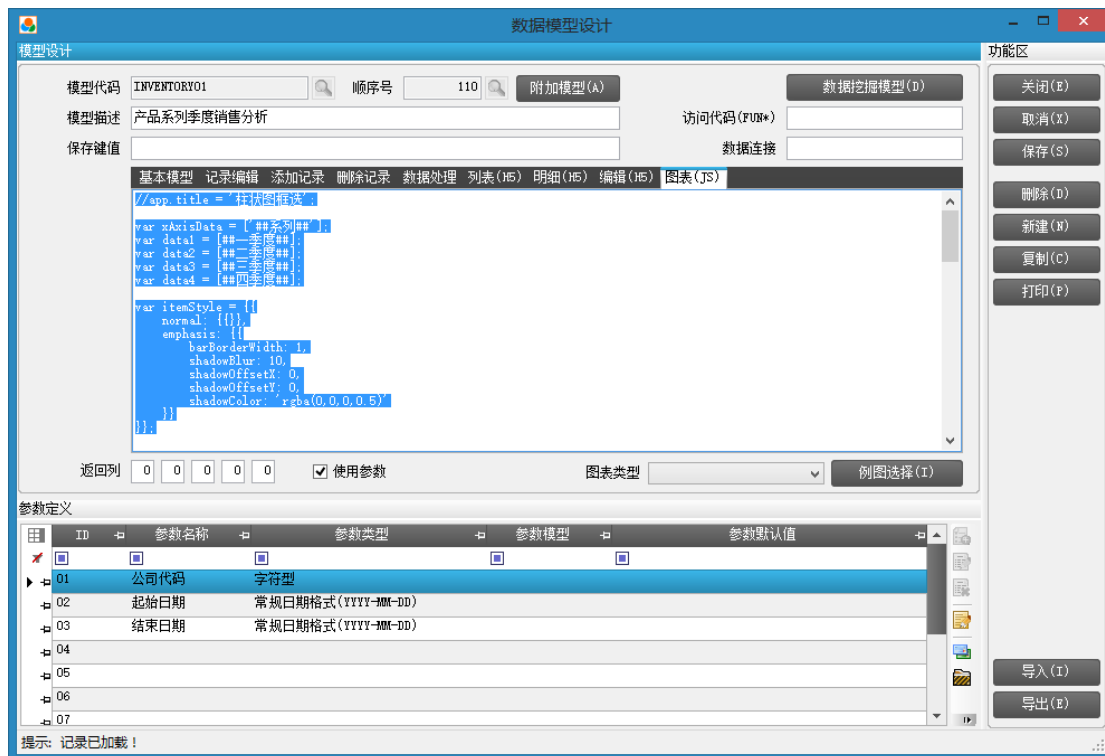
<div class="mui-input-row linkreadonly">
<label>UPDUSR</label>
<input type="text" class=""
placeholder=" 请 输 入 UPDUSR" id="UPDUSR" paradm="" value=""
readonly="readonly" ></div>

<div class="mui-input-row linkreadonly">
<label>CRETIM</label>
<input type="text" class=""
placeholder="请输入 CRETIM" id="CRETIM" paradm="" value="2023-08-04 13:17"
readonly="readonly" ></div>
```

```
<div class="mui-input-row linkreadonly">
<label>UPDTIM</label>
<input type="text" class=""
placeholder="请输入 UPDTIM" id="UPDTIM" paradm="" value="2023-08-04 13:17"
readonly="readonly" ></div>
```

- 数据添加的脚本与编辑脚本类似，仅是没有包含默认数据，即 value=""

30.5 eCharts 脚本测试



- 如图所示，为模型设计图表脚本如下：

```
//app.title = '柱状图框选';
```

```
var xAxisData = ['##系列##'];
```

```
var data1 = [##一季度##];
```

```
var data2 = [##二季度##];
```

```
var data3 = [##三季度##];
```

```
var data4 = [##四季度##];
```

```
var itemStyle = {{
  normal: {{}},
  emphasis: {{
    barBorderWidth: 1,
    shadowBlur: 10,
```

```
        shadowOffsetX: 0,
        shadowOffsetY: 0,
        shadowColor: 'rgba(0,0,0,0.5)'
    }
  });

  option = {
    backgroundColor: '#eee',
    legend: {
      data: ['一季度', '二季度', '三季度', '四季度'],
      align: 'left',
      left: 10
    },
    brush: {
      toolbox: ['rect', 'polygon', 'lineX', 'lineY', 'keep', 'clear'],
      xAxisIndex: 0
    },
    toolbox: {
      feature: {
        magicType: {
          type: ['stack', 'tiled']
        },
        dataView: {}
      }
    },
    tooltip: {},
    xAxis: {
      data: xAxisData,
      name: 'X Axis',
      silent: false,
      axisLine: { onZero: true },
      splitLine: { show: false },
      splitArea: { show: false }
    },
    yAxis: {
      inverse: false,
      splitArea: { show: false }
    },
    grid: {
      left: 100
    },
    series: [{
```

```
        name: '一季度',
        type: 'bar',
        stack: 'one',
        itemStyle: itemStyle,
        data: data1
    }, {
        name: '二季度',
        type: 'bar',
        stack: 'one',
        itemStyle: itemStyle,
        data: data2
    }, {
        name: '三季度',
        type: 'bar',
        stack: 'one',
        itemStyle: itemStyle,
        data: data3
    }, {
        name: '四季度',
        type: 'bar',
        stack: 'one',
        itemStyle: itemStyle,
        data: data4
    }
  ]
};

myChart.setOption(option);
```

- 该数据脚本的功能是，利用产品系列 CLASS0-9 的四个季度的销售数据，形成数据图表
- 请注意，上述脚本中使用了系列数据和四列动态数据
`var xAxisData = ['##系列##'];`
`var data1 = ['##一季度##'];`
`var data2 = ['##二季度##'];`
`var data3 = ['##三季度##'];`
`var data4 = ['##四季度##'];`



- 通过 eChart 脚本检测功能，可以得到经过动态数据处理后的用于生成数据图表的脚本如下：

```
//app.title = '柱状图框选';

var xAxisData =
['Class0','Class1','Class2','Class3','Class4','Class5','Class6','Class7','Class8','Class9'];
var data1 = [190,68,39,129,114,22,131,34,42,81];
var data2 = [73,43,40,15,17,64,65,98,54,54];
var data3 = [239,438,296,33,469,126,165,319,467,267];
var data4 = [91,84,57,54,42,127,121,77,125,106];

var itemStyle = {
  normal: {},
  emphasis: {
    barBorderWidth: 1,
    shadowBlur: 10,
    shadowOffsetX: 0,
    shadowOffsetY: 0,
    shadowColor: 'rgba(0,0,0,0.5)'
  }
};

option = {
  backgroundColor: '#eee',
  legend: {
    data: ['一季度', '二季度', '三季度', '四季度'],
    align: 'left',
    left: 10
```

```
    },
    brush: {
      toolbox: ['rect', 'polygon', 'lineX', 'lineY', 'keep', 'clear'],
      xAxisIndex: 0
    },
    toolbox: {
      feature: {
        magicType: {
          type: ['stack', 'tiled']
        },
        dataView: {}
      }
    },
    tooltip: {},
    xAxis: {
      data: xAxisData,
      name: 'X Axis',
      silent: false,
      axisLine: { onZero: true },
      splitLine: { show: false },
      splitArea: { show: false }
    },
    yAxis: {
      inverse: false,
      splitArea: { show: false }
    },
    grid: {
      left: 100
    },
    series: [{
      name: '一季度',
      type: 'bar',
      stack: 'one',
      itemStyle: itemStyle,
      data: data1
    }, {
      name: '二季度',
      type: 'bar',
      stack: 'one',
      itemStyle: itemStyle,
      data: data2
    }, {
```

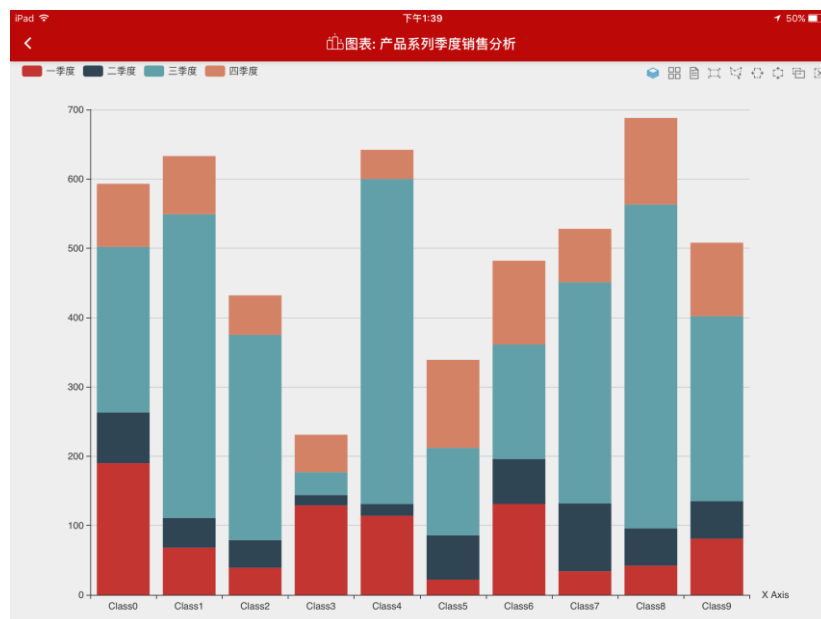
```

        name: '三季度',
        type: 'bar',
        stack: 'one',
        itemStyle: itemStyle,
        data: data3
    }, {
        name: '四季度',
        type: 'bar',
        stack: 'one',
        itemStyle: itemStyle,
        data: data4
    }
  ];

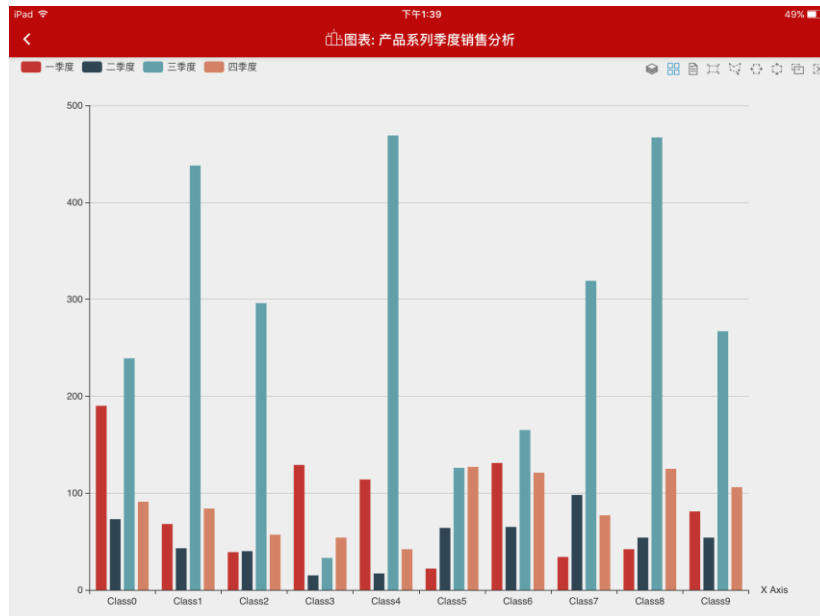
myChart.setOption(option);

```

- 请注意
 - 脚本中的“动态变量”已经被替换成了真实的数据
 - 在检测功能中，使用最多 20 行记录作为测试数据
- 该模型在移动端执行，显示的图表如图所示

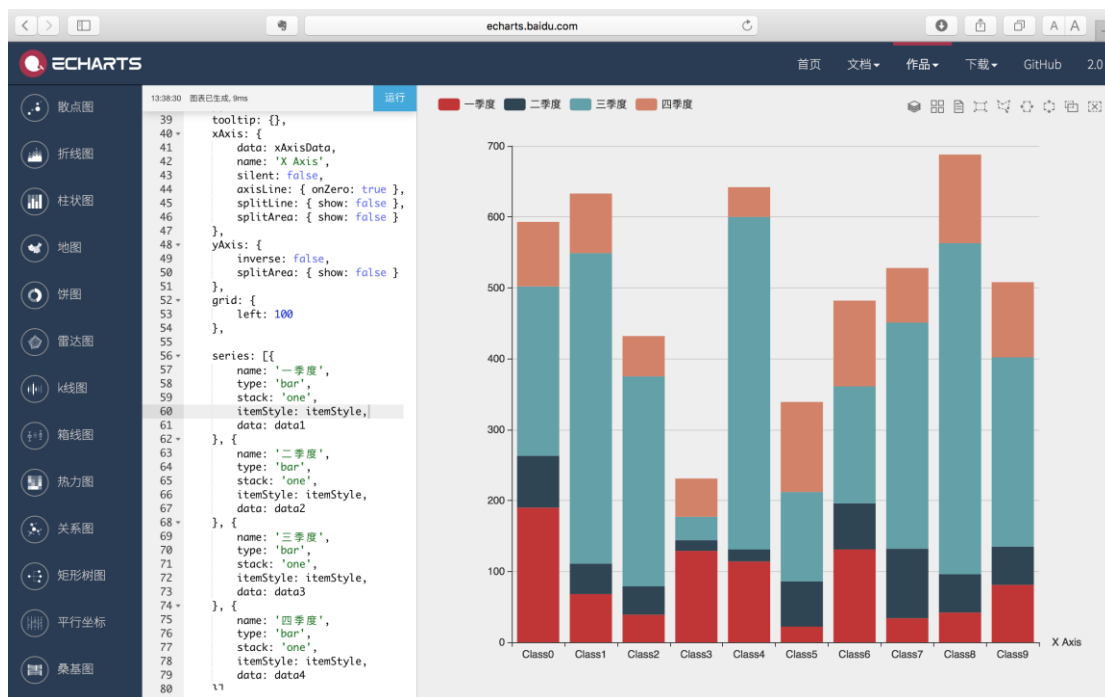


- 该图表还可以通过变换，得到如下的图表展示



30.6 eCharts 调试技巧

在实际项目模型配置过程中，可以可用百度 eChart 平台进行脚本检测。



- 将系统生成的图表脚本粘贴在百度 eCharts 平台，运行即可看到效果

31 自定义列表选择按钮

- 模型代码

```
<div class="mui-input-row">
  <label>PICKLIST</label>
  <button type="button" class="link-pick-btn mui-input-clear linkpicklist" id="self" paratype="PICKERSTR" param=""
value="{3}">{3}</button>
</div>
```

- 详细说明

- 需要在“参数”加一个<button class="link-pick-btn linkpicklist" id="xxxx"></button>标签，其中 class 值中 link-pick-btn 代表该按钮的样式，建议加上，linkpicklist 为默认标识，必须加上的值，然后需要给 button 一个唯一的 id 即可；
paratype="PICKERSTR"

- 效果截图



- 前端操作

点击参数界面的标题即可弹出操作窗口，在相关的位置通过输入不同的值，用英文的逗号隔开“,”即可，也可以用英文的分号“;”来显示不同的文本而调用相应的值，如输入 “男,女;a,b” 即可在选择列表上显示 a 和 b，而调用的值是男和女。PS:标点符号需要是全英文的，中文的标点符号不会识别。

- 后端 JAVASCRIPT 操作



当然也可以在 JAVASCRIPT 脚本里面编写`$("#linkPicklistDiv[data='B3'] textarea").val("男,女");`data 等于按钮的 id，val 里面的值和前端的格式一致。

32 自定义数据过滤

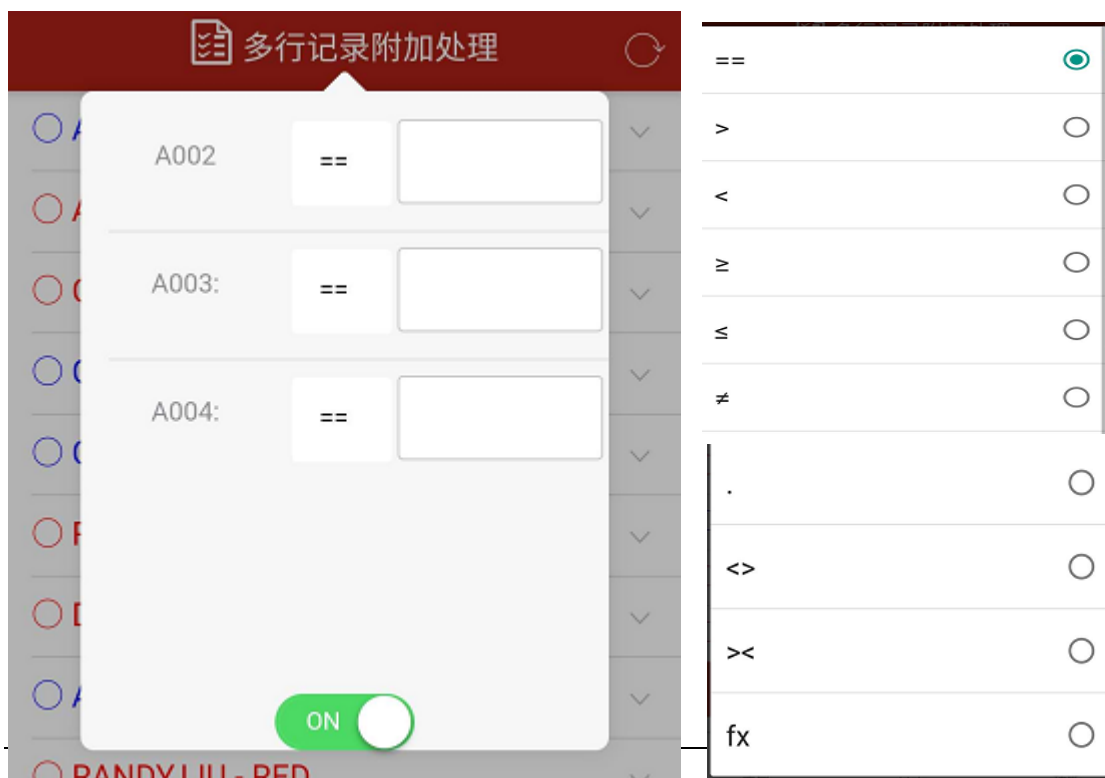
- 模型代码

```
基本模型 记录编辑 添加记录 删除记录 数据处理 列表(H5) 表格(H5) 编辑(H5) 添加(H5) 参
H5批次
<li id='#' class='mui-table-view-cell mui-collapse linkchecks' >
<a class='mui-navigate-right' href='#' >
<font color=#ROW_FORECOLOR#>
○ #A001# - #ROW_FORECOLOR#</font>
</a>
<ul class='mui-collapse-content' >
<p><font color=#576B95 class="linkcheck" data="#A002#">A002</font>: #A002#</p>
<p><font color=#576B95 class="linkcheck" data="#A003#">A003:</font> #A003#</p>
<p><font color=#576B95 class="linkcheck" data="#A004#">A004:</font> #A004#</p>
<p><font color=#576B95>ID:</font> #ID#</p>
</ul></li>
```

- 详细说明

- 在需要过滤的字段的标签里加入 class="linkcheck" data="#A002#"即可
- Linkcheck 为功能标识，data 里面的值是从服务器的获取的值，所以用#xx#获取

- 效果截图



- 前端操作

点击内容界面的标题即可弹出操作窗口，在相关的位置通过输入不同的值，可选择不同的运算符，以及填写不同的值，当前按钮开启后即可执行过滤功能，当前行的数据不符合表达式的时候不进行之后的操作，可支持多选的过滤操作

- 后端 JAVASCRIPT 操作

通过动态的赋值即可，获取相应的 `select` 框和 `textarea` 框进行赋值。

- 表达式符号的解释

- `>`、`<`、`>=`、`<=`、`==`、`!=`

这些符号为基础的表达式，即左边的值与 `textarea` 的值作比较，返回 `true` 或 `false`

- `“.”`

点号的功能未可以直接调用相关的 JAVASCRIPT 函数

- `“<>”`

输出两个值，用空格隔开，代表两个值之外，如 `2 3` 即表示左边的值小于 2 或者大于 3，符合表达式则返回 `true`，否则返回 `false`

- `“><”`

输出两个值，用空格隔开，代表两个值之内，如 `2 3` 即表示左边的值大于 2 并且者小于 3，符合表达式则返回 `true`，否则返回 `false`

- `“fx”`

`fx` 表示左边的值，即可通过完整的书写表达式来处理，如：`fx>3` 表示左边的值大于 3，`fx.length>2` 表示左边的值的长度大于 2