

**TotalLINK**

# 产品手册



上海朝识智能科技有限公司

2020年6月

# 移动应用进阶二

## 目 录

移动应用进阶二 .....	2
1 选择列表 .....	5
1.1 下拉列表选项.....	5
1.2 弹出列表选项.....	8
1.3 搜索列表选项.....	11
2 选项自动带出.....	13
3 重置 .....	18
4 打开第三方应用.....	19
5 自定义快捷功能.....	21
6 提交后直接跳转界面.....	33
7 JS 控制提示判断 .....	34
8 必填追加提示信息.....	39
9 扫描后自动保存--LINK_SCAN_AFTER.....	40
10 formula 公式赋值 .....	42
12.1 案例 1 -val() (常用) .....	43
12.2 案例 2-text() .....	48
12.3 案例 3-attr() .....	50
11 MQTT .....	54
13.1 PC 端 .....	54
13.1.1 案例 1—给订阅默认主题的用户发送消息 .....	54
13.1.2 案例 2—订阅主题并发送消息 .....	56
13.1.3 案例 3—LINKADCTION 发送指令写法一.....	59
13.1.4 案例 4—LINKACTION 发送指令写法二.....	61
13.2 APP 端 .....	62
13.2.1 案例 1—订阅主题 .....	62

13.2.2 案例 2—给固定主题发送消息 ..... 65

12 APP 数据类型控制 ..... 69

# 文档控制

## ■ 主要内容

本文记录 APP 大版本更新的功能点及使用方式。

## ■ 更改记录

日期	版本	作者	备注
2020-03-10	1.0	庄江涛	初始发布
2020-06-16	2.0	Jozey	1. 补充执行固定页面时实现拦截 2. 更新自动带出描述 3. 扫描后自动保存 4. 连续添加
2020-10-10	3.0	Jozey	1. 添加MQTT的内容 2. Formula公式赋值
2020-10-11	3.1	Liz	1. 增加并丰富LINKQUICK内容
2020-10-19	3.2	Jozey	新增app数据类型
2020-10-28	3.3	Liz	新增JS提示判断例子

## ■ 支持版本

非特殊说明的功能，默认前后版本都支持

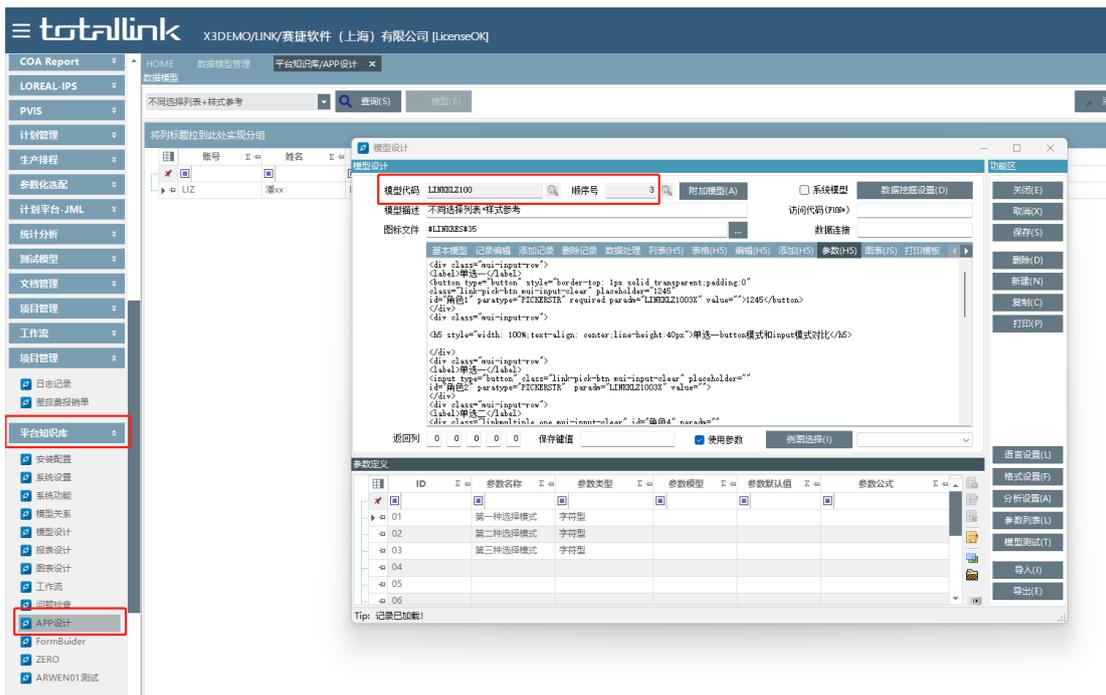
### 仅支持T20版本及以后版本的功能点

- MQTT

# 1 选择列表

## 1.1 下拉列表选项

- 在数据模型中创建一个模型，设置模型参数，便于在参数（H5）设置下拉列表，下拉列表可以在任何 H5 界面设置，实例在参数（H5），若有 TotalLINK 开发环境可参考模型 LINKKLZ100#3-不同选择列表+样式参考。





- 在参数（H5）中加入下拉框代码

参考代码：

```

<div class="mui-input-row">
<label>单选一</label>
<button type="button" style="border-top: 1px solid transparent;padding:0"
class="link-pick-btn mui-input-clear" placeholder="1245" id="角色 1" paratype="PICKERSTR"
paradm="LINKKLZ1003X" value=""></button>
</div>
<div class="mui-input-row">
<label>单选一</label>
<input type="button" class="link-pick-btn mui-input-clear" placeholder=""
id="角色 2" paratype="PICKERSTR" paradm="LINKKLZ1003X" value="">
</div>
    
```

- Paratype 属性：填写下拉框类别的代码
- Paradm 属性：填写下拉列表内容的模型代码

可使用 button 标签，也可使用 input 标签



Button 标签，选择后显示编号和描述



input 标签，选择后只显示编号

## 1.2 弹出列表选项

**模型设计**

模型代码: LINKKLZ100 顺序号: 3 附加模型(A)  系统模型 数据挖痕设置(D)

模型描述: 不同选择列表+样式参考 访问代码(FUN\*):

图标文件: #LINKRES#35 数据连接:

```

</div>
<div class="mui-input-row">
<label>单选</label>
<input type="button" class="link-pick-btn mui-input-clear" placeholder=""
id="角色2" paratype="PICKERSTR" paradm="LINKKLZ1003X" value="">
</div>
<div class="mui-input-row">
<label>单选</label>
<div class="linkmultiple one mui-input-clear" id="角色4" paradm=""
code="LINKKLZ1003X" num="10" text-value="RoleDesc-Role" reload value="PRTPLAN">Business</div>
</div>
<div class="mui-input-row">
<label>单选</label>
<div class="linkdatasearch one" type="text" class="mui-input-clear" placeholder=""
id="账号1" paratype="STR" group="部门"
code="LINKKLZ1003X" num="20" text-value="姓名,账号" para="{角色4}"
automatch="{ 部门显示: 部门, 角色显示: 角色 }" reload value=""></div>
</div>
    
```

返回列: 0 0 0 0 0 保存键值  使用参数 例图选择(I)

ID	参数名称	参数类型	参数模型	参数默认值	参数公式
01	第一种选择模式	字符型			
02	第二种选择模式	字符型			
03	第三种选择模式	字符型			
04					
05					
06					

Tip: 记录已加载!

**模型设计**

模型代码: LINKKLZ1003X 顺序号: 20 附加模型(A)  系统模型 数据挖痕设置(D)

模型描述: 选择用户 访问代码(FUN\*):

图标文件: #LINKRES#99 数据连接:

```

select
LOGINUSER AS 账号,
NAME AS 姓名,
ROLE AS 角色,
DEPT1 AS 部门]
from SYSUSER
where LOGINUSER<>NAME and NAME<>'
AND ROLE LIKE '%[0]*'
order by NAME desc
    
```

返回列: 0 0 0 0 0 保存键值  使用参数 例图选择(I)

ID	参数名称	参数类型	参数模型	参数默认值	参数公式
01	角色	字符型			
02					
03					
04					
05					
06					

Tip: 记录已加载!

- 在参数（H5）中加入下拉框代码

参考代码：

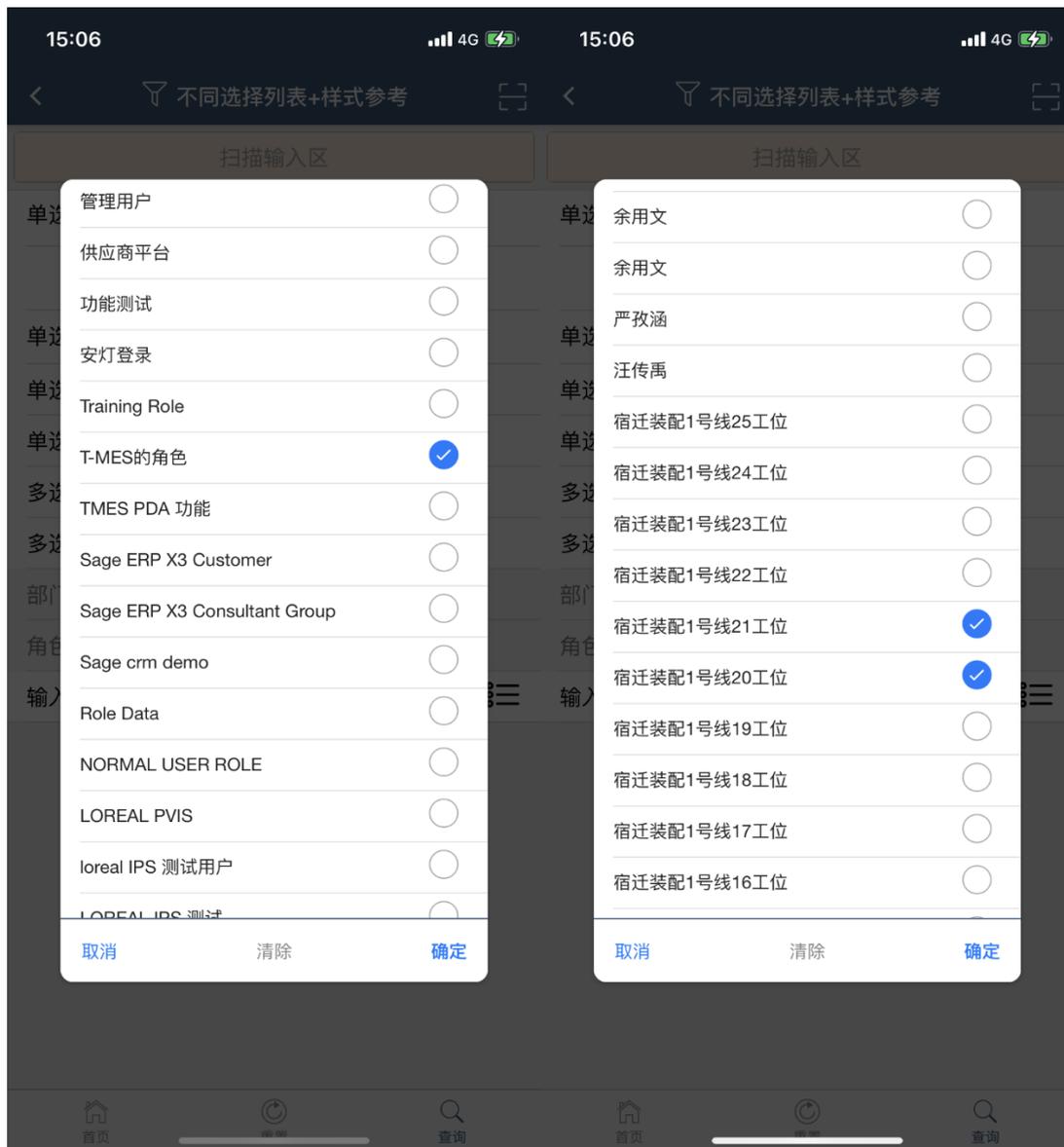
```
<div class="mui-input-row">
<label>单选二</label>
<div class="linkmultiple one mui-input-clear" id="角色 4" paradm="" code="LINKKLZ1003X"
num="10" text-value="RoleDesc-Role" reload value="" >可写默认值 </div>
</div>
<div class="mui-input-row">
<label>多选一</label>
<div class="linkmultiple mui-input-clear"
id="账号 2" paradm="" code="LINKKLZ1003X" num="20" text-value="姓名-账号" reload
value=""></div>
</div>
```

### 1. 当 paradm 属性为空时，即可触发自定义列表

- code 属性：指定模型代码，不指定默认为当前全局的模型代码
- num 属性：指定模型编号，不指定默认为当前全局的模型编号
- para 属性：指定参数，不指定默认为[]数组，para 属性指定值的时候也分两种情况：
  1. 以逗号隔开，会按顺序填充，例如：“1, 2, 3”，则会填充为[1, 2, 3]以此类推。
  2. 前后加 {1, 2, 3}，会取当前界面 id 为 1，id 为 2，id 为 3 的值进行按顺序填充以此类推。
- reload 属性：每次点击都会刷新一次列表重新获取数据。  
ps:不添加只会刷新一次数据，如果需要获取填写后的值或者需要的值是动态生成的则需要加上此属性
- text-value 属性：可以指定对应的字段名，默认是 text 和 value，可指定中文或英文的字段名。例如：text-value="RoleDesc-Role"，意思是会获取字段为"RoleDesc"的值作为显示，取字段名为"Role "的值进行操作。

### 2. 通过 class 的值等于 linkmultiple 作为标识，默认取 paradm 的值的列表，与其他列表选择一样

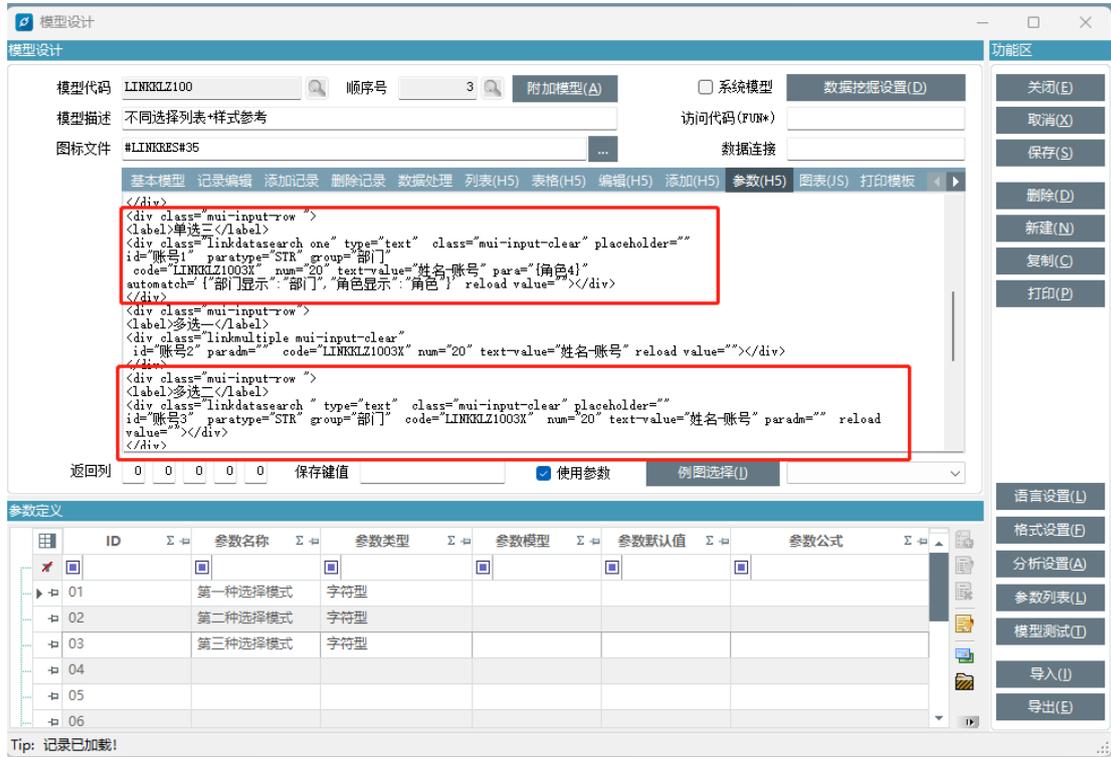
- 标签只能使用 div 标签
- 只允许选择一个 class 添加 one 则表示只允许选择一个值 class="linkmultiple one"，多选去掉 one



弹出列表单选

弹出列表多选

### 1.3 搜索列表选项



- 在参数（H5）中加入下拉框代码

参考代码：

```

<div class="mui-input-row ">
<label>单选三</label>
<div class="linkdatasearch one" type="text" class="mui-input-clear" placeholder=""
id="账号1" paratype="STR" group="部门"
code="LINKKLZ1003X" num="20" text-value="姓名-账号" para="" reload value=""></div>
</div>
<div class="mui-input-row ">
<label>多选二</label>
<div class="linkdatasearch " type="text" class="mui-input-clear" placeholder=""
id="账号3" paratype="STR" group="部门" code="LINKKLZ1003X" num="20" text-value="姓名-
账号" paradm="" reload value=""></div>
</div>
    
```

- app 列表可搜索筛选的功能，用 div 显示，通过 class 的值等于 linkdatasearch 作为标识，默认取 paradm 的值的列表，与其他列表选择一样

APP 版本 1.0.0.105 及以上

相关语法<div id="可搜索过滤的列表" class="linkdatasearch"></div>

class 添加 one 则表示只允许选择一个值 class="linkdatasearch one" ps: 自动带出值只允许单选的情况

备注：需要保证 value 或定义的 value 是唯一的，否则搜索的结果不正确

- 其他属性说明

➢ group: 可支持一级分组，group="字段名"为通过该 “字段名” 进行分组

- expression: 可支持自定义搜索匹配的表达式，默认通过定义的 text 和 value 以及定义的 group 进行模糊计算，也可以自定义模糊匹配的表达式 expression="字段名 1, 字段名 2, 字段名 3" 以此类推，逗号隔开即可。
- 上述表达式等于 expression="{row. 字段名 1.indexOf(r)!=-1||row. 字段名 2.indexOf(r)!=-1||row. 字段名 3.indexOf(r)!=-1}"，即 expression 可通过 {} 来进行自定义的筛选条件：row 表示过滤的当前行，r 为输入值；例子：expression="{row. 字段名 1==r&&row. 字段名 2==r&&row. 字段名 3==r}" 表达式的意思为 显示当字段名 1/2/3 同时等于输入值 r



## 2 选项自动带出

- 首先完成列表选项的实现,再在数据模型中的查询下拉列表内容的模型中查询出需要自动带出的数据
- 并在 H5 标签中加入 automatch 属性

参考代码:

```
<div class="mui-input-row">
<label>自定义单选</label>
<input type="button" class="link-pick-btn mui-input-clear" placeholder="请输入自定义单选"
id="自定义单选" paratype="PICKERSTR" paradm="" code="COUNTRY" num="10" text-value="
编号国家-国家"
automatch='{ "编号": "编号" }' value="">
</div>
<div class="mui-input-row">
<label>编号</label>
<input type="text" class="mui-input-clear" placeholder="请输入编号"
id="编号" paratype="STR" paradm="" value="">
</div>
```

- automatch 属性: 自动带出, 当列表选择后可自动带出 ID 为编号的值, 通过 `automatch='{"id 值": "下拉列表字段名"}'` 的方式配置, 需要严格按照格式配置否则无法识别, 即外围用单引号, 内部需要用双引号。在上面的基础上, 也可通过 JavaScript 的方法直接修改 automatch 方法回调值为当前选择的对象, 回调值 id 和 ele, 即当前点击的按钮 id 值和选取的对象



注: 只有单选才会触发这个自定义函数

**参考代码:**

```
autoMatch=function(id,ele){
    if(id=="自定义单选")
        plus.runtime.openURL(encodeURIComponent("https://www.baidu.com"))
}
```

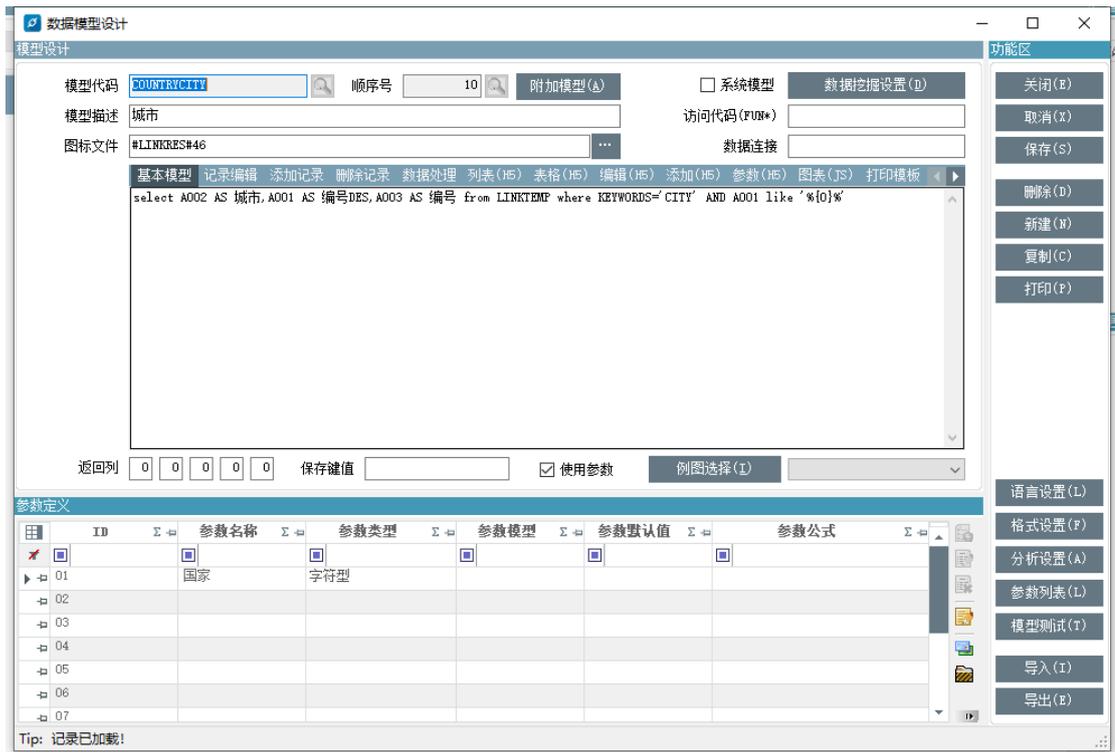
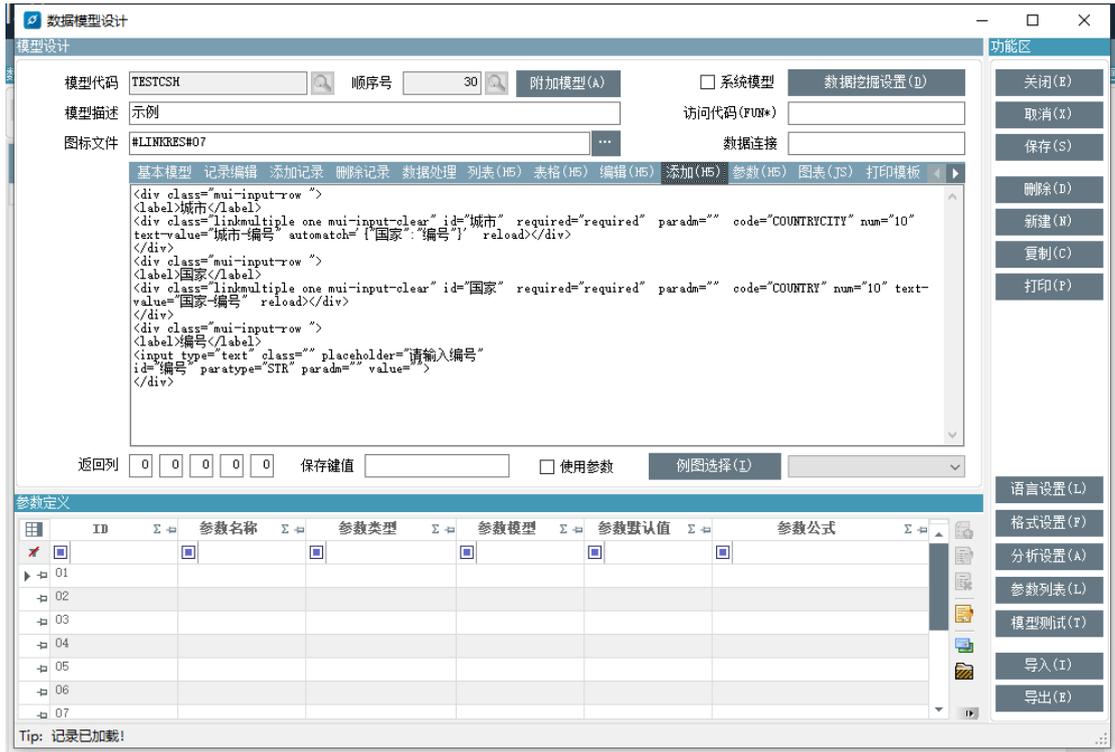
**代码解析:** 点击“自定义单选”字段后，会自动打开百度的网页

- 多个下拉框关联自动带出

当自动带出的值所在位置的控件类型为下拉框时，需要在设置自动带出的下拉框内容模型中将描述字段别名“字段”+“DES”。

例:





参考代码:

```
select A002 AS 城市,A001 AS 编号 DES,A003 AS 编号 from LINKTEMP where
KEYWORDS='CITY' AND A001 like '%{0}%'
HTML 代码:
<div class="mui-input-row ">
<label>城市</label>
<div class="linkmultiple one mui-input-clear" id="城市" required="required"
paradm="" code="COUNTRYCITY" num="10" text-value="城市-编号" automatch="{ "国家
```

```
":"编号"}' reload</div></div>
<div class="mui-input-row ">
<label>国家</label>
<div class="linkmultiple one mui-input-clear" id="国家" required="required"
paradm="" code="COUNTRY" num="10" text-value="国家-编号" reload</div>
</div>
```

**代码解析：**设置自动带出的值为最终保存值，显示值为查询模型中别名为：自动带出字段名+DES

比如：城市下拉框自动带出编号，在 COUNTRYCITY 模型代码中设置一个别名为“编号DES”字段为 APP 界面中显示字段。

## 3 重置

- `mui-input-clear` 默认清空对应区域内的所有可填框和相关按钮，隐藏的按钮，或设置 `not-reset` 属性不清空，例如 `<input type="hidden">` 以及 `<input not-reset type="button">`，则不会被清空对应的值。
- 重置后的默认值设置
  1. `default-text` 属性可以配置显示描述
  2. `default-value` 属性可以配置传入值

## 4 打开第三方应用

- 打开第三方应用，**安卓使用的是包名 com. xxx. xxx，可带额外参数，IOS 只能使用 scheme 值即浏览器方式，且不可带额外参数，IOS 不支持带有额外参数。**

通过在 Javascript 里调用 `launchApp(pname, action, extra)`

➤ `Pname`: (String 类型) 第三程序包名

仅 Android 平台支持，标识程序包名，其他平台忽略此属性值。

平台支持：

1. Android-2.2+(支持)：第三程序的包名，如 “io.dcloud.HelloH5”
2. Ios-ALL(不支持)：忽略此属性

➤ `Action`: (String 类型) 程序的操作行为

Android 平台上与系统的 `action` 值一致；iOS 平台为要调用程序的 URLScheme 格式字符串。

平台支持：

1. Android-2.2+(支持)：第三程序的操作行为，与系统的 `action` 值完全一致。
2. iOS-4.3+(支持)：要调用程序的 URLScheme 格式字符串。

➤ `Extra`: (JSON 类型) 调用程序的参数

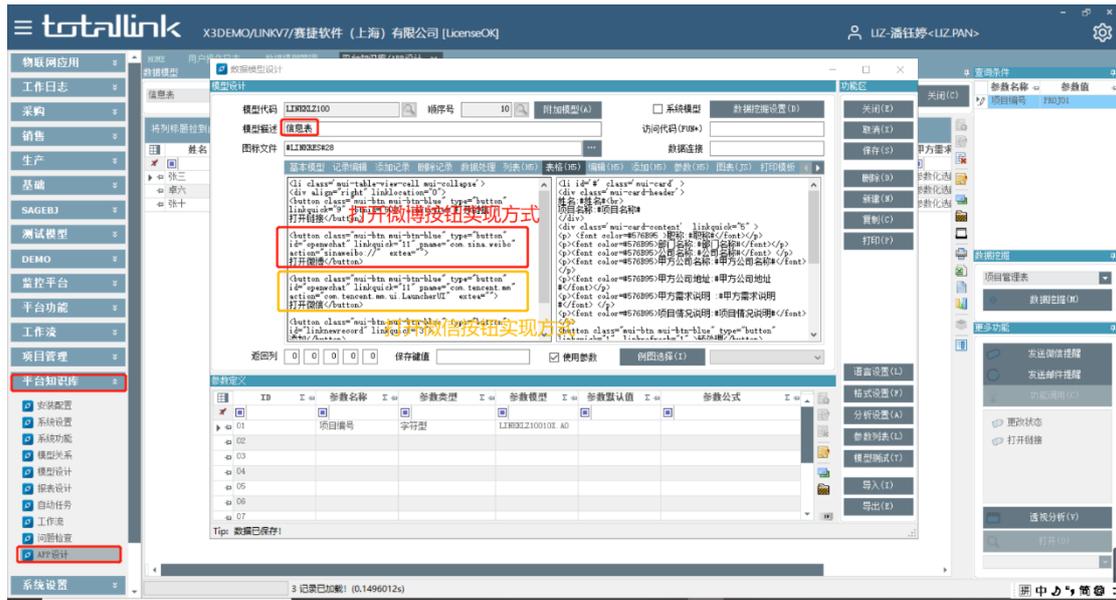
仅 Android 平台支持，为 JSON 格式，用于传递给要调用程序的参数，如 `extra: {url: "http://www.html5plus.org"}`。

平台支持：

1. Android-2.2+(支持)：传递给第三程序的参数，必须是 JSON 格式数据，如 “{ ‘name’ : ‘value’ , ‘name2’ : ‘value2’ }”。
2. iOS-ALL(不支持)：忽略此属性值。

参考代码：

```
Android: pname="com.tencent.mm"//打开微信
         pname="com.sina.weibo"//打开微博
IOS:     action="sinaweibo://" //打开微博
```



参考代码:

```
<button class="mui-btn mui-btn-blue" type="button" id="openwechat" linkquick="11"
pname="com.sina.weibo" action="sinaweibo://" extea="">
打开微博</button>
```

注意: 以上方式是通过 linkquick=11 的方式实现。

打开第三方的参考属性值:

序号	应用	pname	action
1	微信	com.tencent.mm	weixin://
2	微博	com.sina.weibo	sinaweibo://
3	手机 QQ	com.tencent.mobileqq	mqqapi://
4	钉钉	com.alibaba.android.rimet	dingtalk://

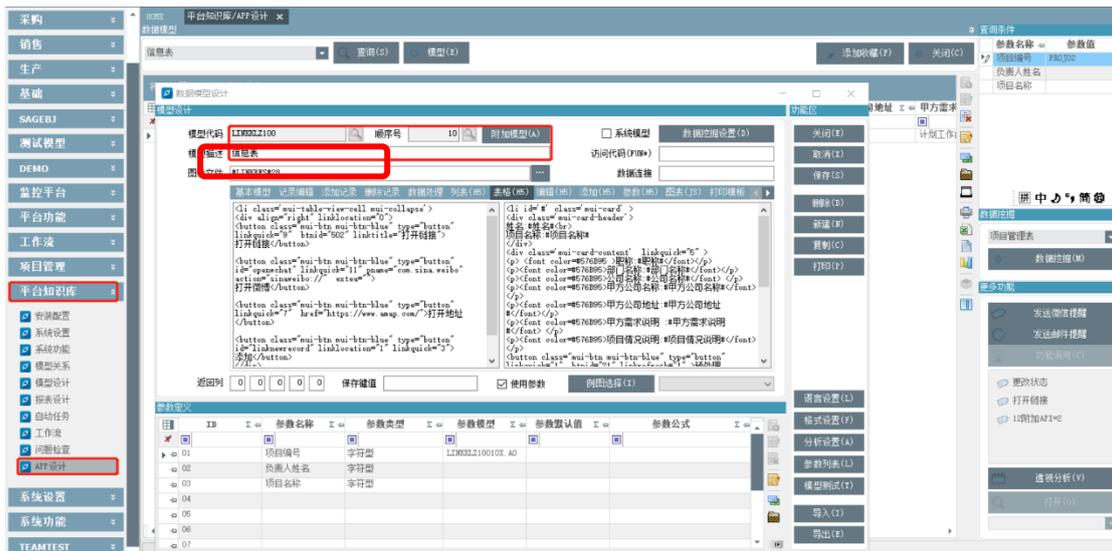
## 5 自定义快捷功能

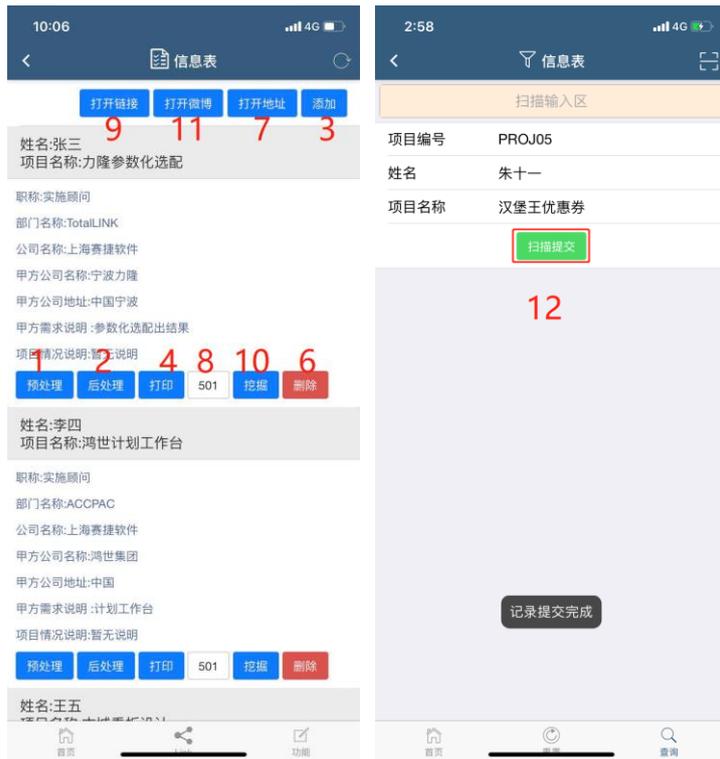
- 在内容列表 H5 中任意元素任意位置添加 linkquick 属性可以自定义相关快捷功能，具体指以及对对应功能，如：linkquick="X"，X 的取值范围为 1-11，以下具体属性的作用可参考具体功能的文档说明：
  - 1:预处理功能 linkpreprocess，全局界面可加上 para 属性
  - 2:后处理功能 linkpostprocess，全局界面可加上 para 属性
  - 3:添加功能 recNew 附带位置信息，可定义 linklocation 属性，用于获取当前的地理位置
    - linklocation=" 0" :不启动获取当前的地理位置
    - linklocation=" 1" :启动获取当前的地理位置，以#LINKLOCATION#的方式
    - linklocation=" 2" :启动获取当前的地理位置，以#LINKLOCATIONJSON#的方式
  - 4:打印功能 linktagprint
  - 5:编辑功能 linkedit 附带位置信息，可定义 linklocation 属性，全局界面可加上 para 属性
  - 6:删除功能 linkremove，全局界面可加上 para 属性
  - 7:打开网址功能，需要定义 href 属性填写网址
  - 8:长按文本菜单功能contextmenulist需要定义btnid,linkparafill,linkrefresh,linktitle属性;可定义linkimage,linklocation属性
    - btnid: btnid 与 id 的性质一样
    - linkparafill: 是否有传递参数, 0: 否; 1: 是
    - linkrefresh: 是否刷新, 0: 否; 1: 是
    - linktitle: 定义标题名称
    - linkimage: 调用获取图片的功能, 0: 否; 1: 是
  - 9:链接功能 linkurllist 需要定义 btnid,linktitle 属性，用于模型中的链接
  - 10:数据挖掘功能 linkcontextdatamine，需要定义 btnid，linkparafill，linkrefresh，linktitle 属性;可定义 echartflag 属性
  - 11:打开第三方 APP 需要定义 pname,action,extea 属性,具体属性详情请查看 6 打开第三方应用
  - 12: 调用 API 接口实现复杂的快捷功能，需定义 code 属性（模型代码），num 属性（模型编号），para 属性（模型参数），不填则提交界面上所有数据，{}花括号代表取界面上的 id，其他格式为静态数据，多个参数可用逗号隔开，api 属性（接口类型）
    - /api/DataModel/linkDMDatasetResult 获取模型数据 api=1,需搭配 formula

使用，才可显示获取到的数据

- /api/DataModel/linkDMGridRowSubmit 附加功能执行 api=2, 可调用附加模型的基本模型、记录编辑、添加记录、删除记录、数据处理、功能调用的脚本
  - /api/DataModel/linkDMContextFunBack 数据处理单行 api=3, 调用附加模型 501-520 的数据处理脚本、功能调用的脚本, script=“501-520”, code=“基础模型的模型代码”。
  - /api/DataModel/linkDMGridDataSubmit 多行记录数据处理 api=4  
目前 APP 端支持单行处理
  - /api/LINK/linkFormula 获取公式结果 api=5
  - /api/DataModel/linkDMFormData 载入模板/跳转刷新 api=6
  - /api/DataModel/linkDMDataSubmit 全部数据一起提交接口 api=7  
api= 7 接口比较复杂, app 端无法测试, 暂时适用 formbuilder
- 13: 低功耗蓝牙, 通过蓝牙设备读写数据, 同时支持公式赋值功能
- service 属性: 服务 id, 未指定则取第一个
  - key 属性: 特征值 id, 未指定则取第一个
  - notify 属性: 订阅特征值, 是否启用订阅功能
  - write 属性: 写入数据, 格式为 write=“写入的内容”
- 14: 表示链接 MQTT 协议, 具体请参考 [《16.2 APP 端》](#)

注: 使用 linkquick 属性自定义快捷功能, 设置其属性, 可参照 H5DateList 模型中的属性设置 (必须要在可调试模式下, 才可看到具体信息)。





在平台知识库>>APP 设置中搭建自定义快捷功能，首先我们要知道自定义快捷键的格式，具体如下：

```
<button class="mui-btn" type="button" linkquick="X" (linkquick="编号", 编号不同, 其设置属性不同) code="模型编号" num="模型顺序号" para="模型参数"><span class="mui-icon mui-icon-flag"> (图标, 根据 class 不同, 显示不同图标) </span> 显示内容</button>
```

- 根据下列要求在信息表定义快捷方式。

序号	要求	分析
1	建立信息表 A 和项目管理表 B，关联 A. 项目编码=b. 项目编码，针对这两张表做添加、编辑、挖掘、删除的操作	编辑：linkquick=5; 添加：linkquick=3; 删除：linkquick=6; 挖掘：linkquick=10
2	建立 21 模型（预处理），邮件提醒功能，发送‘项目情况说明’内容	预处理：linkquick=1code="模型编号" num="模型顺序号" para="模型参数"
3	建立 41 模型（后处理），邮件发送功能，发送‘项目情况说明’内容	后处理：linkquick=2code="模型编号" num="模型顺序号" para="模型参数"
4	打印功能实现，打印相关内容	打印：linkquick=4
5	在查询界面加上‘打开地址’的按钮测试，打开链接地址	直接打开链接：linkquick=7
6	在查询界面加上‘打开微博’的按钮测试，打开第三方 APP	打开第三方应用：linkquick=11
7	在查询界面加上‘打开链接’的按钮测试，通过模型打开 formbuilder 的链接	模型打开链接：linkquick=9
8	定义 501 按钮，用于长按文本菜单功能	长按文本菜单功能：linkquick=8

9	扫描解析后，执行附加模型的数据处理，并调用 callfunction	附加功能执行:linkquick=12, api=2
10	通过 LINKQUICK=3 添加调用自动获取地址功能	LINKLOCATION 引用地址实例

Linkquicik:1-10 的快捷功能，序号 1-8



具体参考代码如下：

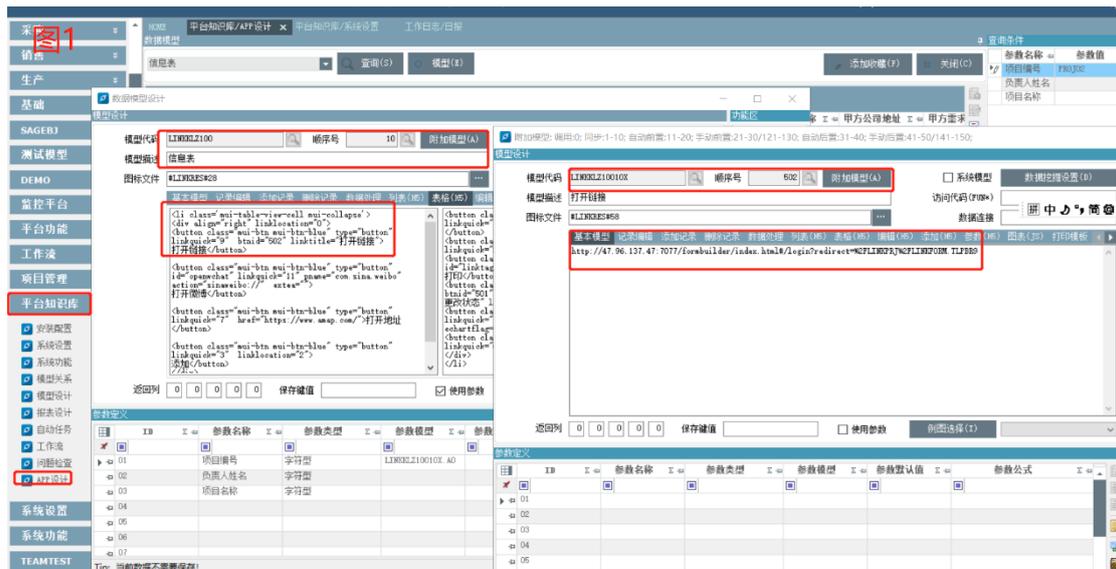
```

/*表格左侧的代码*/
<li class='mui-table-view-cell mui-collapse'>
<div align="right" >
<button class="mui-btn mui-btn-blue" type="button" linkquick="9" btnid="502"
linktitle="打开链接">打开链接</button>
/*通过模型打开链接，定义一个附加 502 模型基本模型中填写一个网址，具体请查看图 1*/

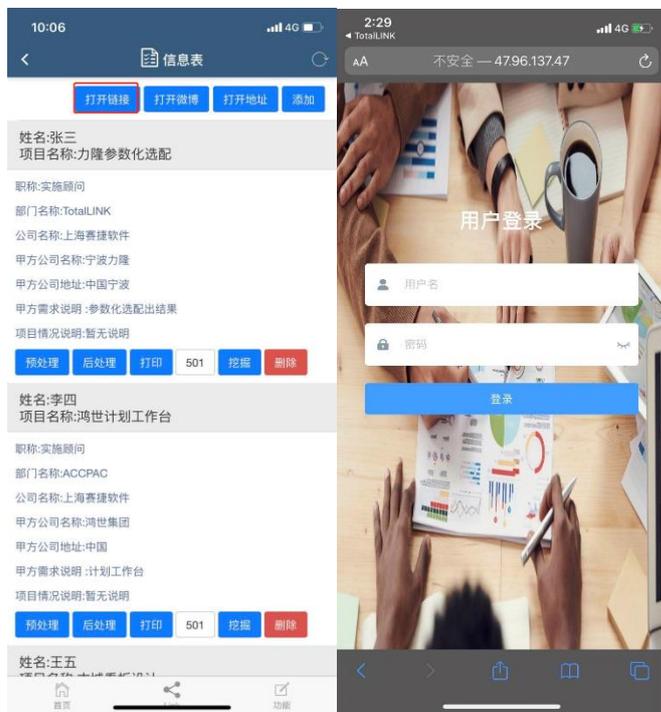
<button class="mui-btn mui-btn-blue" type="button" id="openwechat" linkquick="11"
pname="com.sina.weibo" action="sinaweibo://" extea="">
打开微博</button>
/*打开第三方软件*/

<button class="mui-btn mui-btn-blue" type="button" linkquick="7"
href="https://www.amap.com/">打开地址</button>
/*直接打开链接地址*/

<button class="mui-btn mui-btn-blue" type="button" linkquick="3"
linklocation="2">添加</button>
/*点击添加按钮，跳转添加界面，并获取地理位置*/
</div>
</li>
#LINKTABLEROWS# /*在表头的结尾处需要加上#LINKTABLEROWS#，连接表内容*/
    
```



例如：打开链接按钮结果：



表格右侧参考代码具体如下：

```

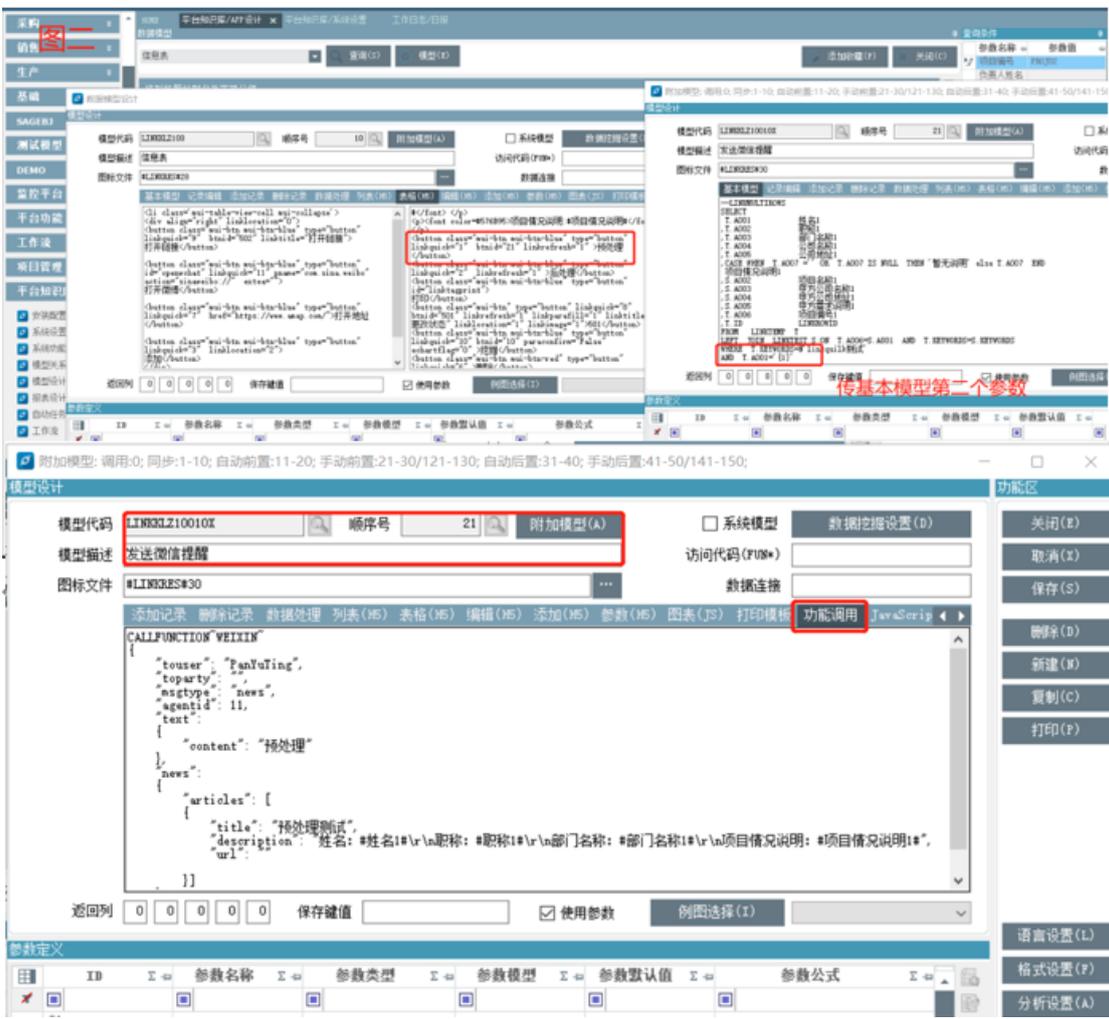
/*表格右侧参考代码*/
<li id='#' class='mui-card' >
<div class='mui-card-header'>
姓名:#姓名#<br>
项目名称:#项目名称#
</div>
<div class='mui-card-content' linkquick="5" linklocation="2">
/*在 div 中将编辑的快捷键，点击 div 所在的区域就会*/
<p> <font color=#576B95 >职称:#职称#</font></p>
<p><font color=#576B95>部门名称:#部门名称#</font></p>
<p><font color=#576B95>公司名称:#公司名称#</font> </p>
<p><font color=#576B95>甲方公司名称:#甲方公司名称#</font> </p>
<p><font color=#576B95>甲方公司地址:#甲方公司地址#</font></p>
<p><font color=#576B95>甲方需求说明 :#甲方需求说明#</font> </p>
    
```

```

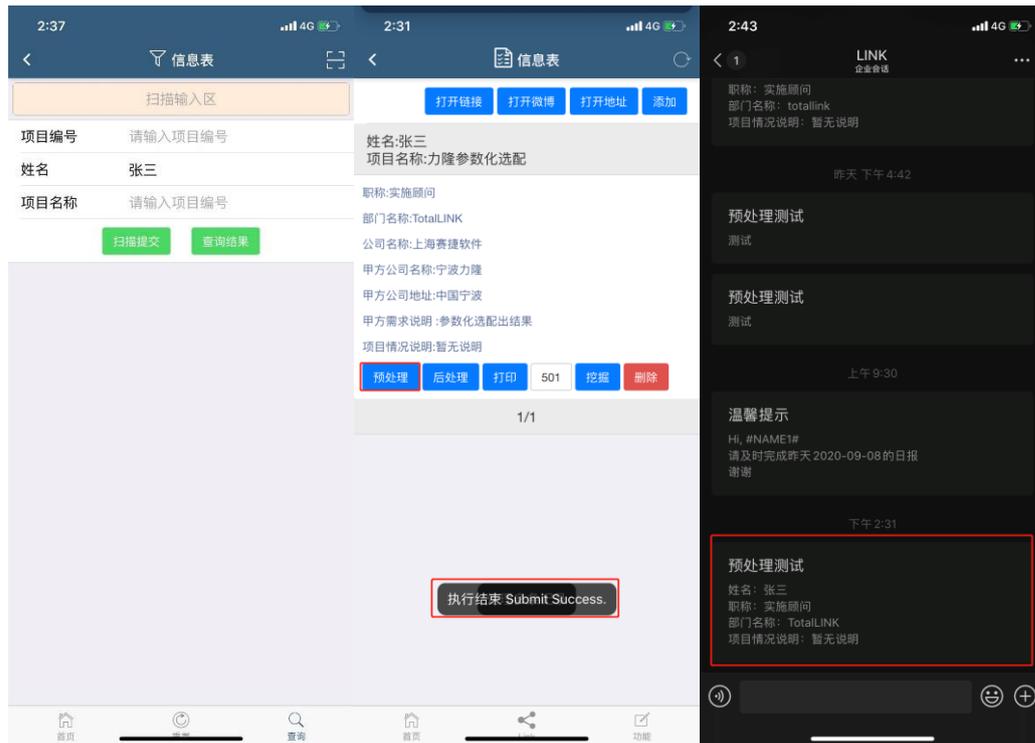
<p><font color=#576B95>项目情况说明:#项目情况说明#</font> </p>
<button class="mui-btn mui-btn-blue" type="button" linkquick="1" btnid="21"
linkrefresh="1" code="LINKKLZ10010X" num="21" para="">预处理</button>
/*微信发送, 需要将姓名做参数传递到附加 21 模型当中, 具体实现方式, 图 2*/
<button class="mui-btn mui-btn-blue" type="button" linkquick="2" btnid="41"
linkrefresh="1" code="LINKKLZ10010X" num="41" para="{0}">后处理</button>
/*邮件发送, 需要将姓名做参数传递到附加 41 模型当中, 同预处理方式一致 */
<button class="mui-btn mui-btn-blue" type="button" linkquick="4">
打印</button>

<button class="mui-btn" type="button" linkquick="8" btnid="501" linkrefresh="1"
linkparafill="1" linktitle="更改状态" linklocation="1" linkimage="1">501</button>

<button class="mui-btn mui-btn-blue" type="button" linkquick="10" btnid="10"
paraconfirm="False" echartflag="0" >挖掘</button>
/*删除按钮*/
<button class="mui-btn mui-btn-red" type="button" linkquick="6" >删除</button>
/*删除按钮*/
</div></li>
    
```



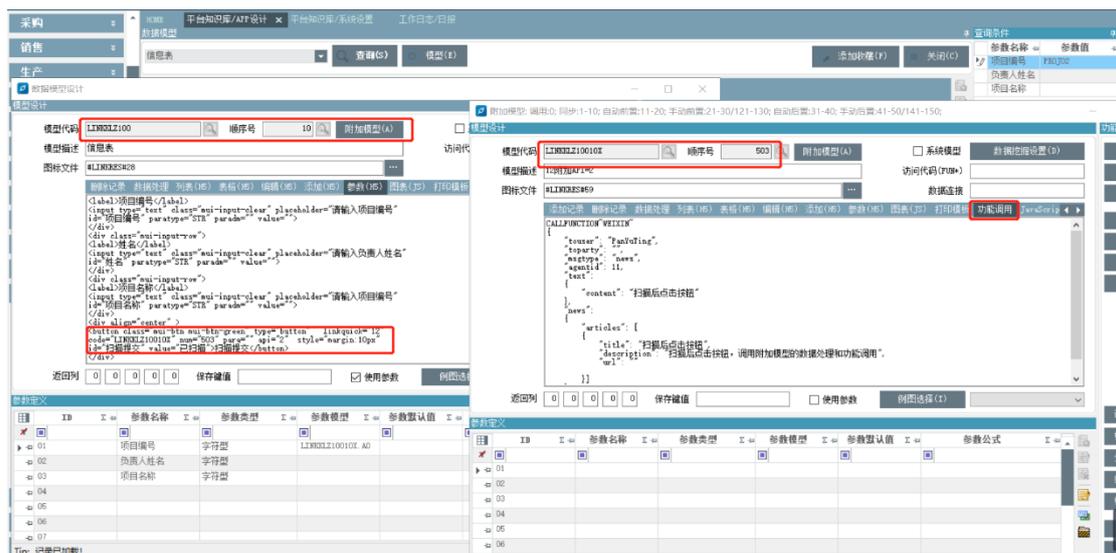
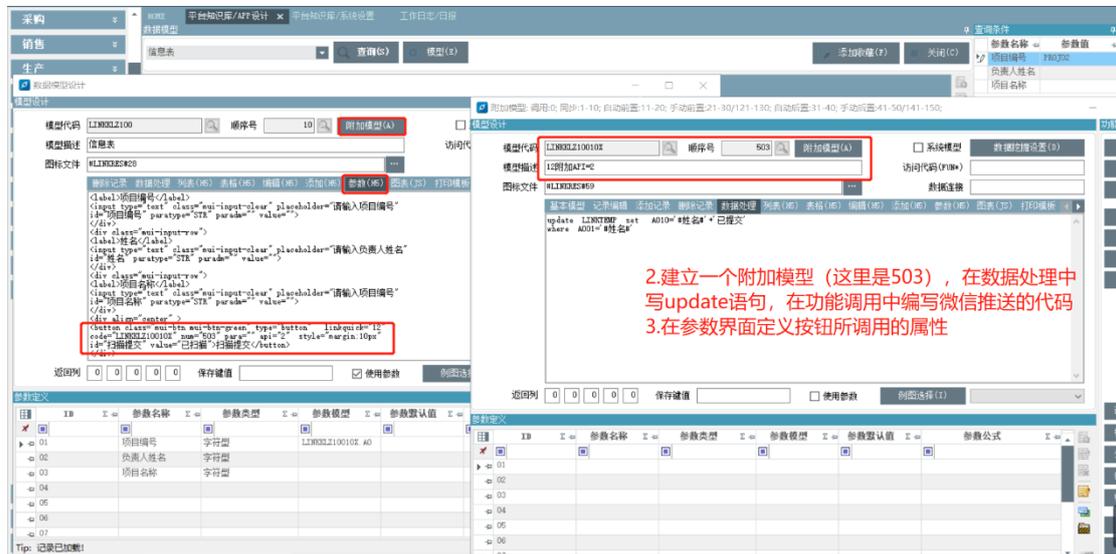
例如 预处理按钮操作结果:



长按文本菜单功能的结果:



linkquick=12 的快捷功能, 序号 9



具体参考代码如下:

```
<button class="mui-btn mui-btn-green" type="button" linkquick="12"
code="LINKKLZ10010X" num="503" para="" api="2" style="margin:10px"
```

```
id="扫描提交" value="已扫描">扫描提交</button>
```

实现效果：



```
1 | select * from LINKTEMP WHERE KEYWORDS='linkquick测试' and A001='朱十一'
```

	XDS	OPEDATE	A001	A002	A003	A004	A005	A006	A007	A008	A009	A010	A011	A012	A013	A014	A015	A016	A017
1	uilk测试	09 4 2020 2:05PM	朱十一	实施顾问	totalink	上海赛捷软件	上海普陀	PROJ05	NULL	NULL	朱十一已提交	NULL							

附加503模型数据处理修改的结果

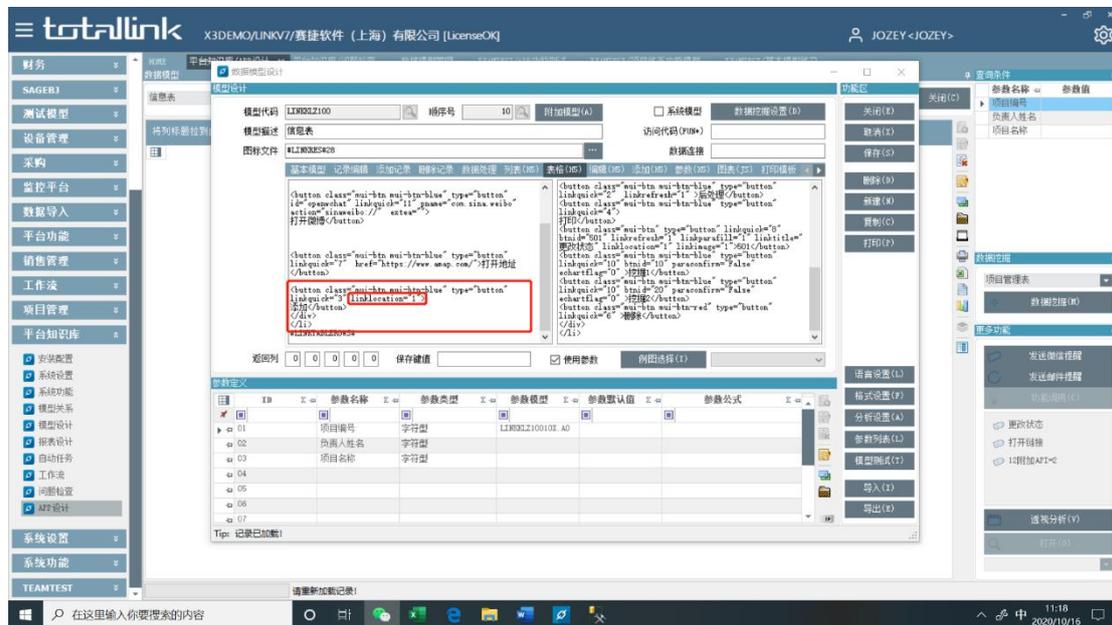
注意：1. 根据需求具体问题具体分析自定义快捷功能。

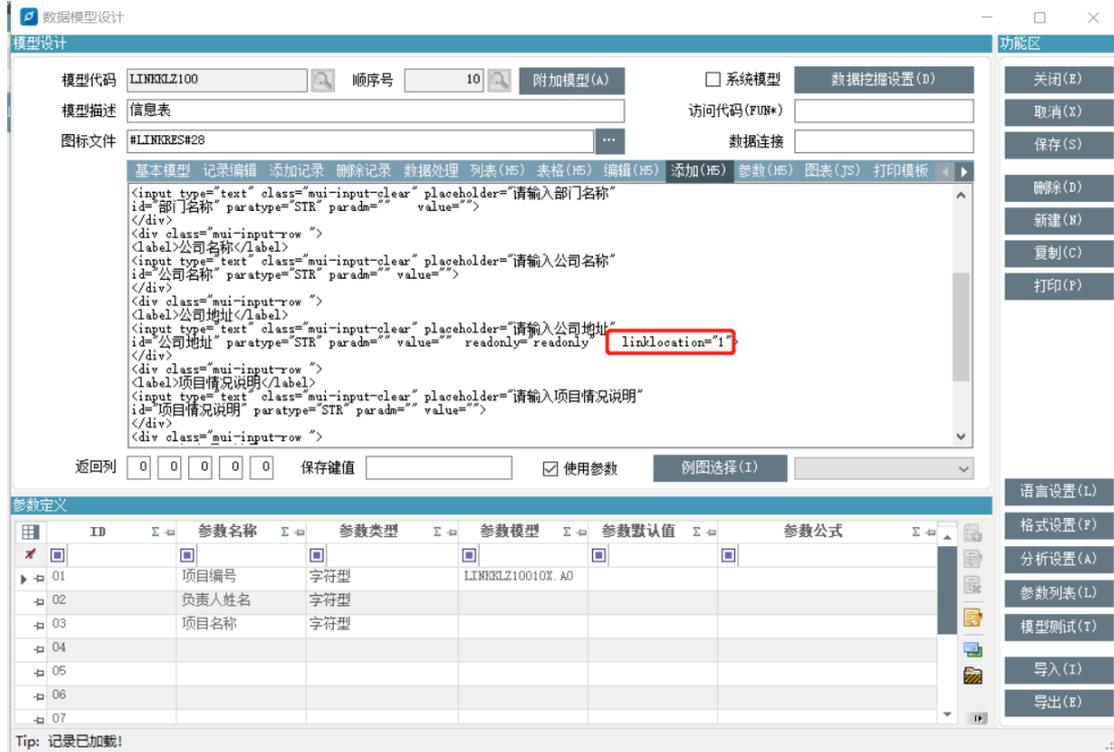
2. linkquick 属性相当于特殊 id，当 H5 代码中写 id 属性，就不用写 linkquick 属性，两者写其一，否则在调用时会调两次。

- 调用地址实例
- 确保手机定位打开



在添加按钮及调用地址的标签加 linklocation="1"调用本地地址





参考代码:

```

添加按钮:
<button class="mui-btn mui-btn-blue" type="button" linkquick="3" linklocation="1">
添加</button>
</div>

地址标签:
<div class="mui-input-row ">
<label>公司地址</label>
<input type="text" class="mui-input-clear" placeholder="请输入公司地址"
id=" 公 司 地 址 " paratype="STR" paradm="" value="" readonly="readonly"
linklocation="1">
</div>
    
```

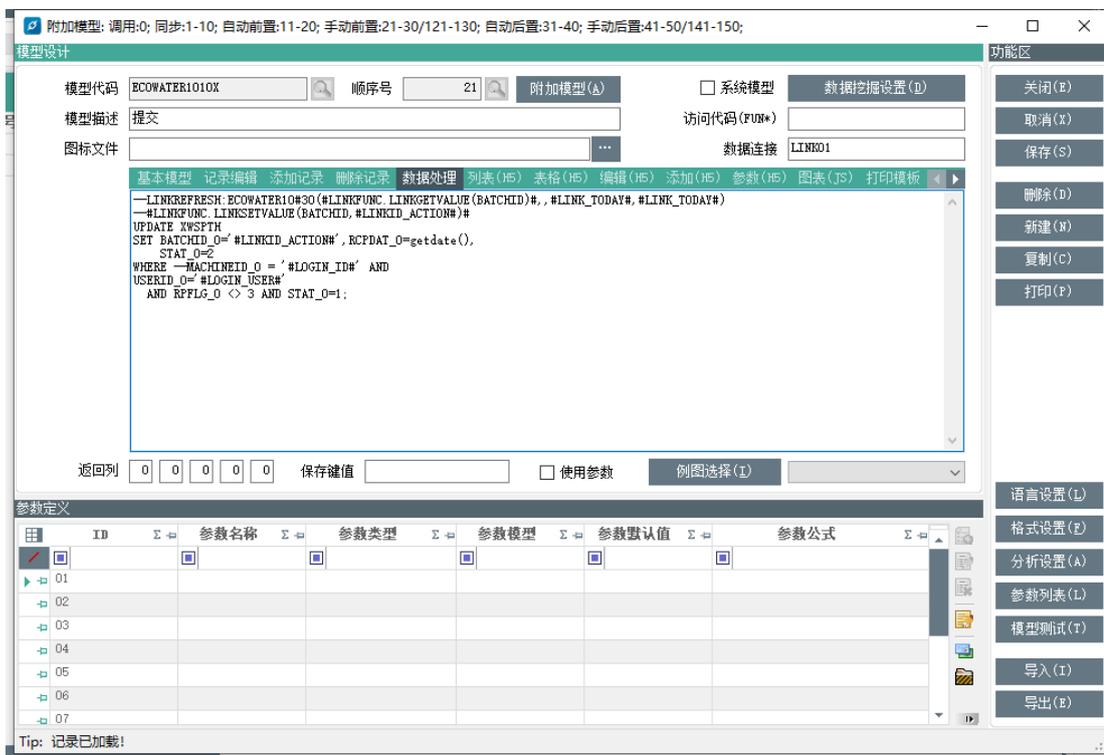
调用结果:

当添加完一条数据后, 系统自动获取到本地地址

The screenshot shows a SQL query result in Microsoft SQL Server Enterprise Manager. The query is: `SELECT * FROM LINKBIHQ WHERE KEYWORDS='linkbi测试'`. The result table has the following columns: SYS, COMPTR, KEYWORDS, OPERATE, A001, A002, A003, A004, A005, A006. The data includes various system identifiers and keywords, with one row containing a Chinese address: '中国上海市普陀区靖远路南660号'. A red box highlights this row, and a red arrow points to it.

SYS	COMPTR	KEYWORDS	OPERATE	A001	A002	A003	A004	A005	A006
1	JOZYE1	8820983-7190-4402-0344-C77628FC172	linkbi测试	10 15 2020	2:14PM	曹丫丫	安装顾问	Totalink	总维护
2	JOZYE1	860973C-8389-48C8-B081-E09ADC2932B7	linkbi测试	10 15 2020	5:43PM	梁磊	升级	Link	总维护
3	JOZYE1	860973C-8389-48C8-B081-E09ADC2932B7	linkbi测试	10 15 2020	5:44PM	刘六			中国上海市普陀区靖远路南660号
4	JOZYE1	860973C-8389-48C8-B081-E09ADC2932B7	linkbi测试	10 15 2020	5:50PM	胡翰	技术		#LINKLOCATION#
5	JOZYE1	860973C-8389-48C8-B081-E09ADC2932B7	linkbi测试	10 15 2020	5:55PM	陆强			#LINKLOCATION#
6	JOZYE1	860973C-8389-48C8-B081-E09ADC2932B7	linkbi测试	10 15 2020	5:58PM	梁磊			[{"coordinate": "ec102", "address": {"country": "中国..."
7	LIZ FAX	98478D1E-60CA-4405-88E8-2005B7A9C44	linkbi测试	09 2 2020	11:05AM	王五	安装顾问	IS	上海赛捷软件
8	LIZ FAX	98478D1E-60CA-4405-88E8-2005B7A9C44	linkbi测试	09 2 2020	11:06AM	梁六	安装顾问	Totalink	上海赛捷软件
9	LIZ FAX	98478D1E-60CA-4405-88E8-2005B7A9C44	linkbi测试	09 2 2020	5:41PM	梁六	安装顾问	Totalink	上海赛捷软件
10	LIZ FAX1	018840C2-05E9-406C-8090-533281705740	linkbi测试	09 4 2020	2:03PM	朱十一	安装顾问	totalink	上海赛捷软件
11	LIZ FAX1	018840C2-05E9-406C-8090-533281705740	linkbi测试	09 8 2020	10:02AM	胡翰	安装顾问	totalink	上海赛捷软件
12	LIZ FAX1	018840C2-05E9-406C-8090-533281705740	linkbi测试	09 9 2020	10:14AM	李十二	安装顾问	IS	上海赛捷软件
13	LIZ FAX1	A155A702-05E9-406C-8090-533281705740	linkbi测试	09 9 2020	10:15AM	梁十三	安装顾问	totalink	上海赛捷软件

## 6 提交后直接跳转界面



参考代码:

```
--
LINKREFRESH:ECOWATER10#30(#LINKFUNC.LINKGETVALUE(BATCHID)#,,#LINK_TODAY#,#LINK_TODAY#)
```

代码格式: LINKREFRESH:模型代码#模型序号(#参数 1#, #参数 2#, #参数 2 #)

模型代码、模型序号、参数: 填写需要跳转到的界面的数据信息

## 7 JS 控制提示判断

- 全局的状态列表 BEFORE\_STA\_ARRAY 与 AFTER\_STA\_ARRAY

取值-数组, 默认["para", "new", "attach", "memo", "linkquick"], 例: BEFORE\_STA\_ARRAY = ["para", "new", "attach", "memo"]

BEFORE\_STA\_ARRAY: 执行前校验的状态列表:

AFTER\_STA\_ARRAY: 执行后触发的状态列表

para 为模型参数界面

attach 为参数更新界面

new 为添加编辑数据界面

memo 为添加编辑的多功能界面 (比如附加 501、502 等)

- 执行前校验

函数名: `beforeFunction(sta) { return false }`

默认 `return false` 则验证通过

`return true` 则验证不通过

sta 中返回 BEFORE\_STA\_ARRAY 全局状态

- 执行后触发

函数名: `afterFunction(sta) { }`

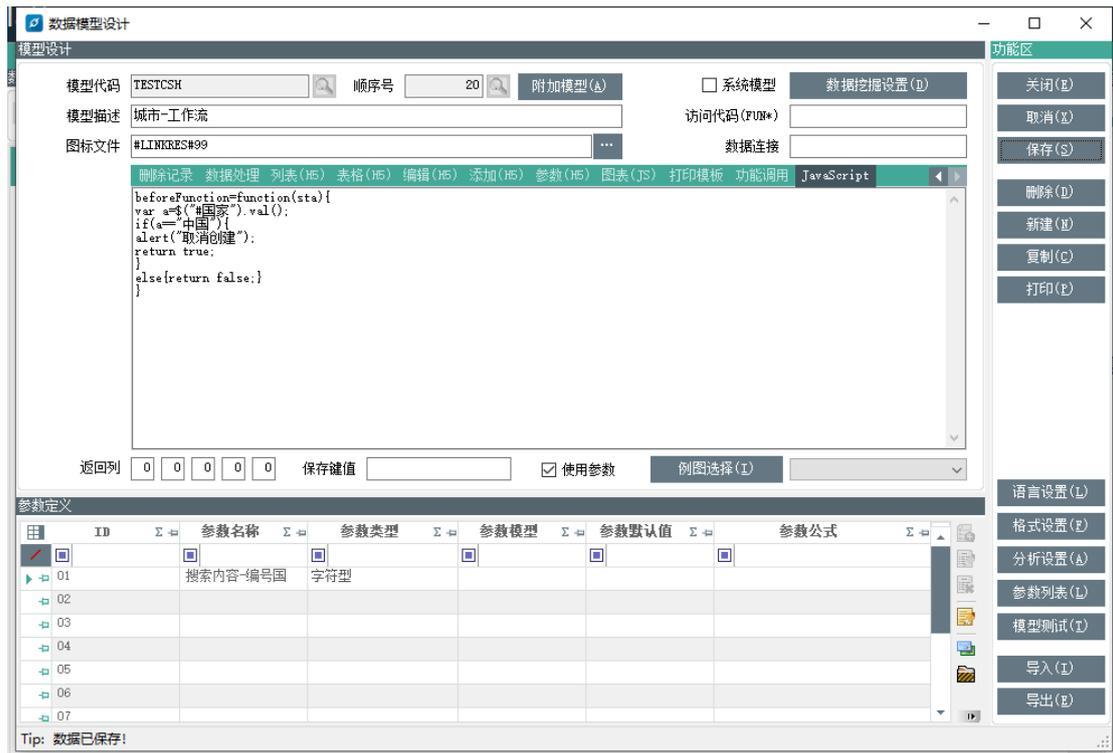
sta 中返回 AFTER\_STA\_ARRAY 全局状态

异步执行不区分, 只在当前代码执行后的最后触发, 过程中涉及异步操作不予处理

**注: js 代码中判断结束后都需要 `return` 返回 `true` 或 `false`**

### 执行前校验:

例: 在点击保存时, 根据编写的判断条件进行判断, 并提示编写的内容, 可控制是否继续进行保存。



参考代码:

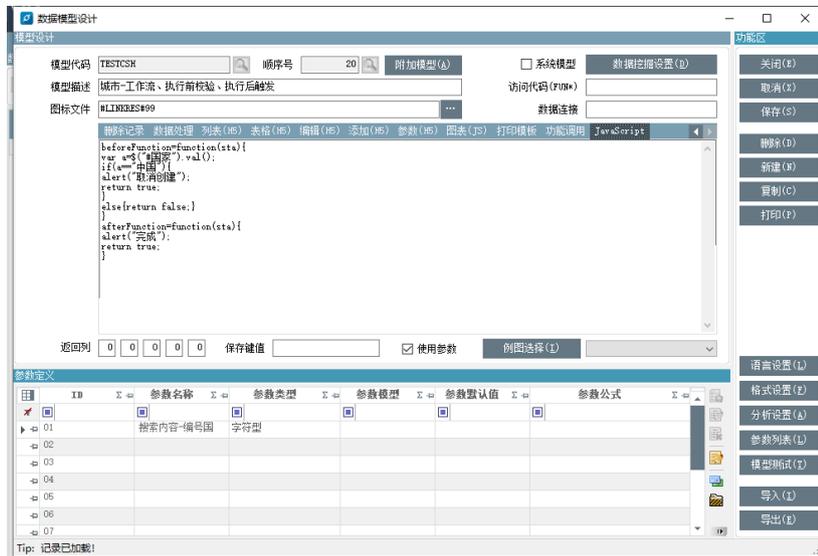
```
beforeFunction=function(sta){
var a=$("#国家").val();
if(a=="中国"){
alert("取消创建");
return true;
}
else
```

```
{return false;}}
```

上面的 JS 默认在执行“para”, "new", "attach", "memo"四种状态动作时弹出提示框

执行后校验:

在每次按钮执行完触发提示等事件



参考代码:

```
afterFunction=function(sta){
alert("完成");
return true;
}
```

默认下支持["para", "new", "attach", "memo", "linkquick"]五种模式

如果只希望在执行固定页面时提示，有两种写法：

写法一：

```
var BEFORE_STA_ARRAY=["para", "new"]
beforeFunction=function(sta){
var a=$("#国家").val();
if(a=="中国"){
alert("取消创建");
return true;
}
else
{return false;}}
```

写法二：

```
beforeFunction=function(sta){
var a=$("#国家").val();
if((sta="para" || sta=="new")&&a=="中国"&&sta=="linkquick1"&&sta=="linkquick12"){
alert("取消创建");
return true;
}
else
{return false;}}
```

案例：

扫描解析出产品编号、库位、超出秒数、数量，点击提交按钮，提交解析出来的数据。

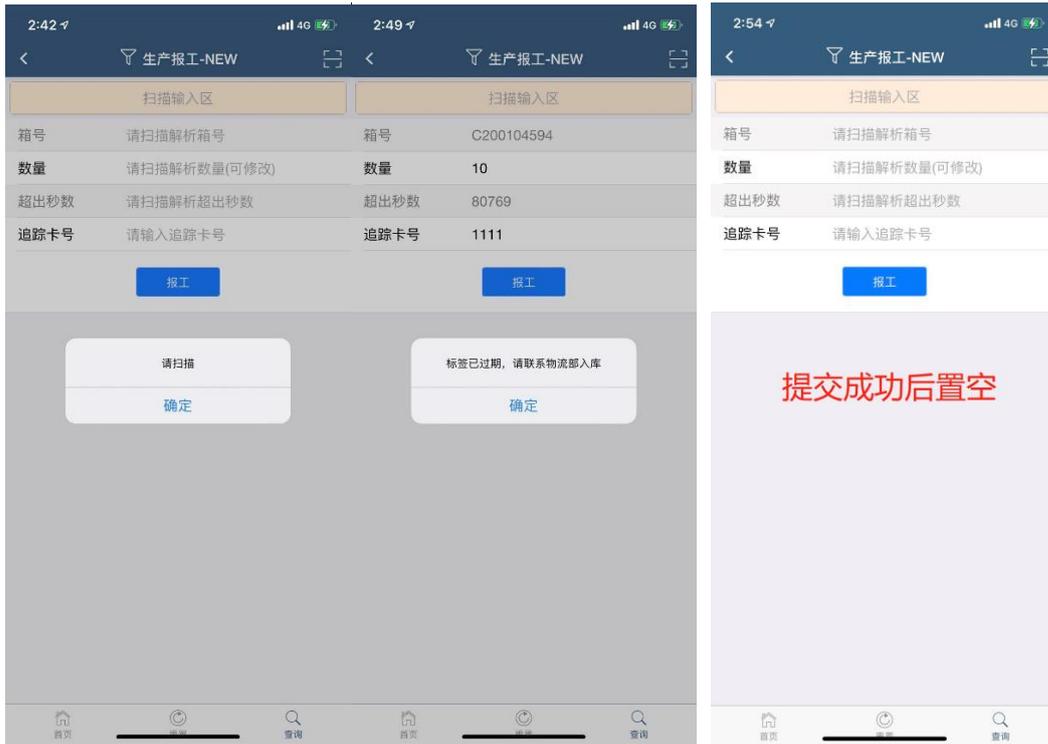
要求：1.操作失误提示，未扫描就点击提交，需提示：请扫描后提交

2.扫描出超出秒数>43200，需提示标签已过期，请联系物流部入库

3.提交成功后，将产品编号、库位、超出秒数、数量置为空



效果：



参考代码:

```

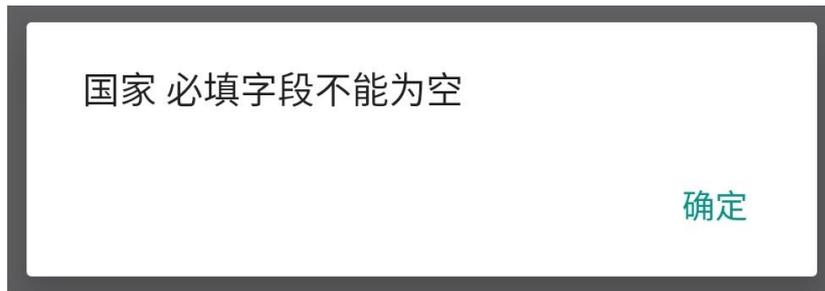
beforeFunction=function(sta){//提交前
var a=$("#箱号").val();
if(a==""){
alert("请扫描");
return true;
}else if($("#超出秒数").val()>43200){
alert("标签已过期, 请联系物流部入库");
return true;
}else if($("#追踪卡号").val()==''){
alert("请输入追踪卡号");
return true;
}
else
{return false;}
}
afterFunction=function(sta){//提交后
if(sta=="linkquick12"){
$("#箱号,#数量,#超出秒数,#追踪卡号").val("");
}
}
    
```

## 8 必填追加提示信息

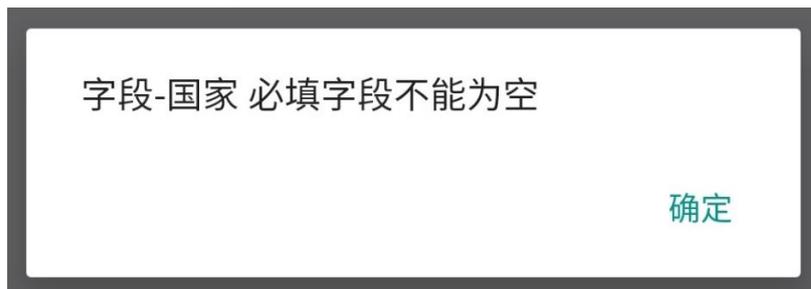
- 必填追加提示信息加入 tip 属性

在 H5 中，标签设置为 required（必填）时，控制其必填提示信息，需要追加 tip 属性，在原有提示信息中追加自定义信息。

原提示信息



追加提示信息后

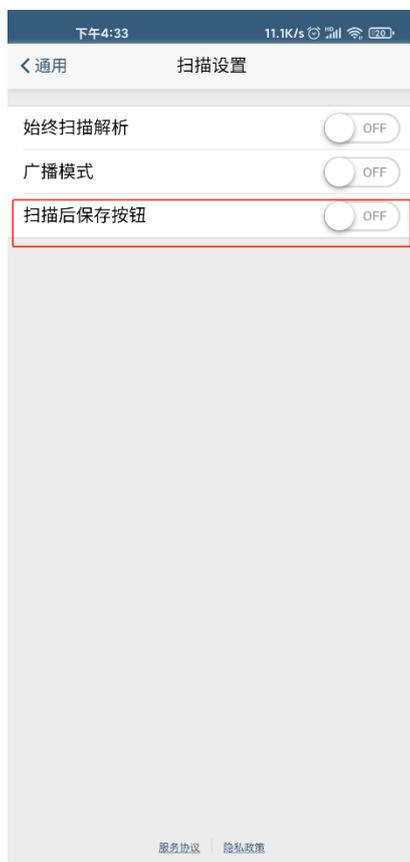


参考代码：

```
<div class="mui-input-row">
<label>国家</label>
<input type='text' required tip=" 字段 - 国家 " class=" mui-input-clear "
placeholder="请输入国家"
id="国家" paradm="" value="">
</div>
```

## 9 扫描后自动保存--LINK\_SCAN\_AFTER

- 通过 APP “我的-通用-扫描设置” 的 “扫描后保存” 按钮，可以全局定义是否显示扫描后是直接保存还是手动保存。1 标识开启，0 不开启，默认不开启。



- 如果想单独修改模型的扫描模式，可以在模型 JS 模块中写入以下代码，表示开启扫描后出现 “自动保存” 按钮，默认是关闭，打开即开启

```
LINK_SCAN_AFTER="1"
```



- 如果想通过某个字段控制扫描后自动保存功能，可以在 js 加入下面的代码

```
reScanFlag=function(id){return true;}
```

解析：id 为各个控件的 id 值，默认 return true 即执行，return false 则不执行  
比如，下面的代码表示，当国家字段值为中国时，sta 返回 true，即 1，则开启扫描后自动保存功能

```
reScanFlag=function(sta){  
  var a=$("#国家").val();  
  if(a=="中国"){  
    alert("取消创建");  
    return true;  
  }  
  else  
  {return false;}}
```

## 10 formula 公式赋值

formula 公式赋值指在相对应的字段标签内嵌入“data.Table.x”或 js 格式代码，与 linkquick 方法搭配使用，达到给当前字段或其它字段赋值的效果。

**基本写法：**

- data.Table.x(x 为所调用查询模型的第几行的数据，0 为第一行)有三种调用方法，var()、text()、attr()，分别对应不同的使用场景

- formula=" id.val=data.Table.x " —常用
- formula=" id.text=data.Table.x "
- formula=" id.attr=data.Table.x "

- js 也有三种调用方法，var()、text()、attr()，分别对应不同的使用场景。{}花括号也支持编写 js 代码。（注意 formula 里的代码涉及到双引号的地方都改为单引号）

- formula="{\$('#id').val('值')}" —常用
- formula="{\$('#id').text('值')}"
- formula="{\$('#id').attr('属性','值')}"

**使用对比：**

- data.Table.x 与 linkquick=12, api=1 搭配使用，可以将页面输入的参数传至另外一个模型，并将返回值填到相对应的字段
- formula-js 与 linkquick=12, api=1 搭配使用，可以将页面输入的参数传至另外一个模型，并将返回值与页面上的信息做计算填到相对应的字段。Js 的这种方式甚至支持更复杂的用法，仅需要把相对应的 js 代码放到花括号即可。

**var()与 text()：**

- var() 取的是对应标签里的实际值，比如 input 标签里实际的值，<input type="text" id="input1" value="123">里的 123
- text() 指的是标签里的值，比如<div>我是一段文本</div>这里的“我是一段文本”

**attr()方法：**

- attr()：formula 此处的 attr 还可以直接写成 formula="{\$('#数据6').css('color','blue')}"
- attr 可以是 value、css

可以通过下面的方法更好的了解 var() 和 text() 方法的区别，大家根据兴趣阅读：

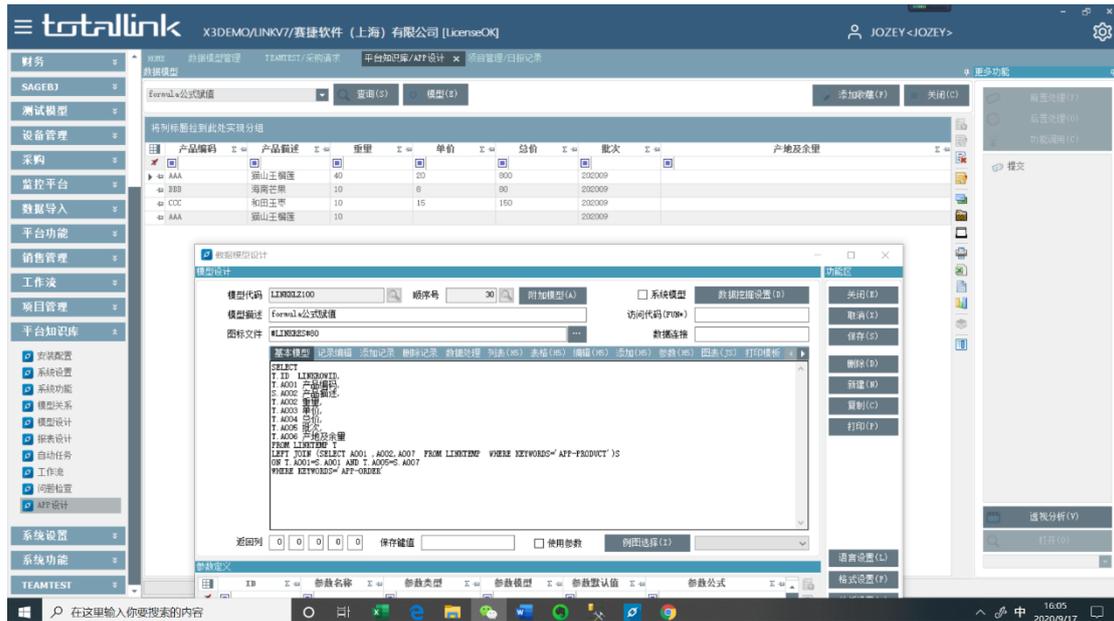
<https://www.cnblogs.com/WhiteM/p/7470105.html>

## 12.1 案例 1 -val() (常用)

id.val=data.Table.x 与 formula="{\${#id}'.val('值')}" 的应用

案例需求描述：新增一条产品订单记录时，需要输入产品编码、产品批次、重量的信息，然后点击按钮获取产品单价并计算总价填写到页面上

- 如下图，查询模型关联了订单和产品列表

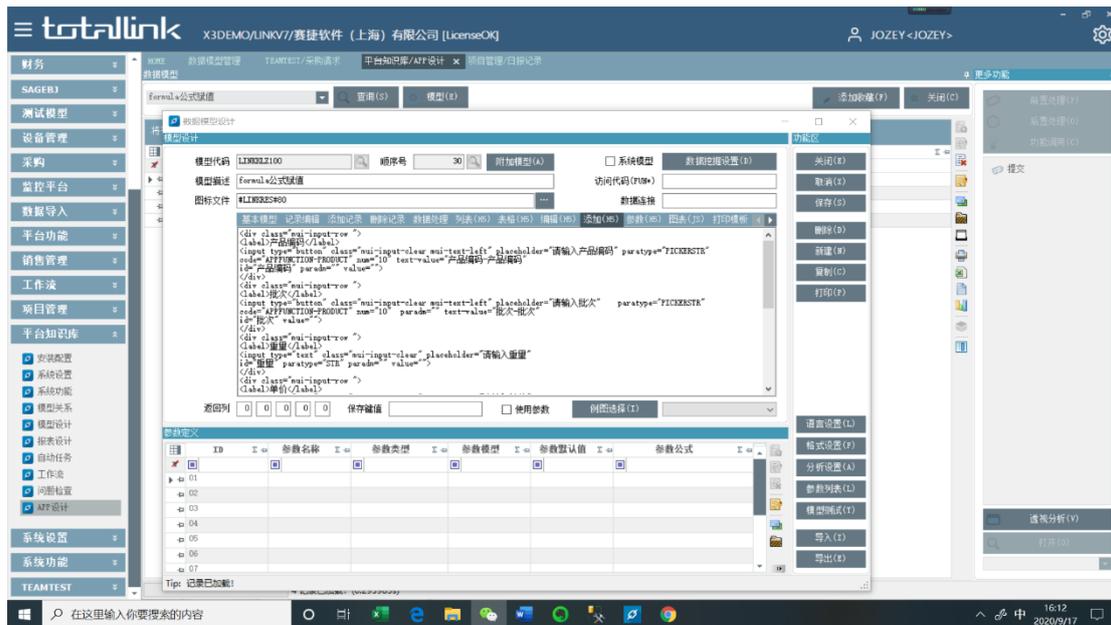


参考代码：

```

SELECT
T.ID LINKROWID,
T.A001 产品编码,
S.A002 产品描述,
T.A002 重量,
T.A003 单价,
T.A004 总价,
T.A005 批次,
T.A006 产地及余量
FROM LINKTEMP T
LEFT JOIN (SELECT A001 ,A002,A007 FROM LINKTEMP WHERE KEYWORDS='APP-PRODUCT')S
ON T.A001=S.A001 AND T.A005=S.A007
WHERE KEYWORDS='APP-ORDER'
    
```

- 添加 (H5) 模块加入以下代码



参考代码:

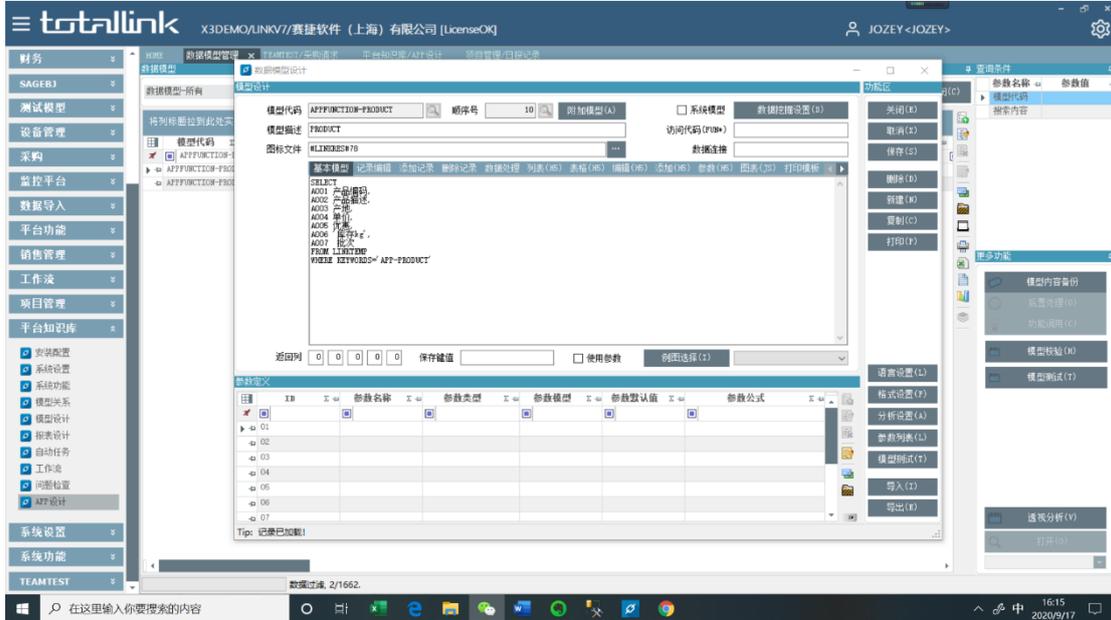
```

<div class="mui-input-row ">
<label>产品编码</label>
<input type="button" class="mui-input-clear mui-text-left" placeholder="请输入产品
编码" paratype="PICKERSTR" code="APPFUNCTION-PRODUCT" num="10" text-value="产品编码
-产品编码"
id="产品编码" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>批次</label>
<input type="button" class="mui-input-clear mui-text-left" placeholder="请输入批次
" paratype="PICKERSTR" code="APPFUNCTION-PRODUCT" num="10" paradm="" text-
value="批次-批次"
id="批次" value="">
</div>
<div class="mui-input-row ">
<label>重量</label>
<input type="text" class="mui-input-clear" placeholder="请输入重量"
id="重量" paratype="STR" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>单价</label>
<input type="text" class="mui-input-clear" placeholder="请输入单价"
id="单价" paratype="STR" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>总价</label>
<input type="text" class="mui-input-clear" placeholder="请输入总价"
id="总价" paratype="STR" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>产地及余量</label>
<input type="text" class="mui-input-clear" placeholder="请输入产地及余量"
id="产地及余量" paratype="STR" paradm="" value="">
</div>

<input type="button" linkquick="12" api="1" code="APPFUNCTION-PRODUCT-PARA"
num="10" para="{产品编码,批次}" formula="单价.val=data.Table.0.单价" id="BLUE1"
    
```

```
value="计算 1-(.val)" >
<input type="button" linkquick="12" api="1" code="APPFUNCTION-PRODUCT-PARA"
num="10" para="{ 产品 编码 , 批次}" formula="{var a=${'# 重量 ').val()*1;var
b=result.data.Table[0].单价*1;${'#单价 ').val(b);${'#总价 ').val(b*a);${'#产地及余量
').val(result.data.Table[0].产地)}" id="BLUE2" value="计算 2-(js/val)" >
```

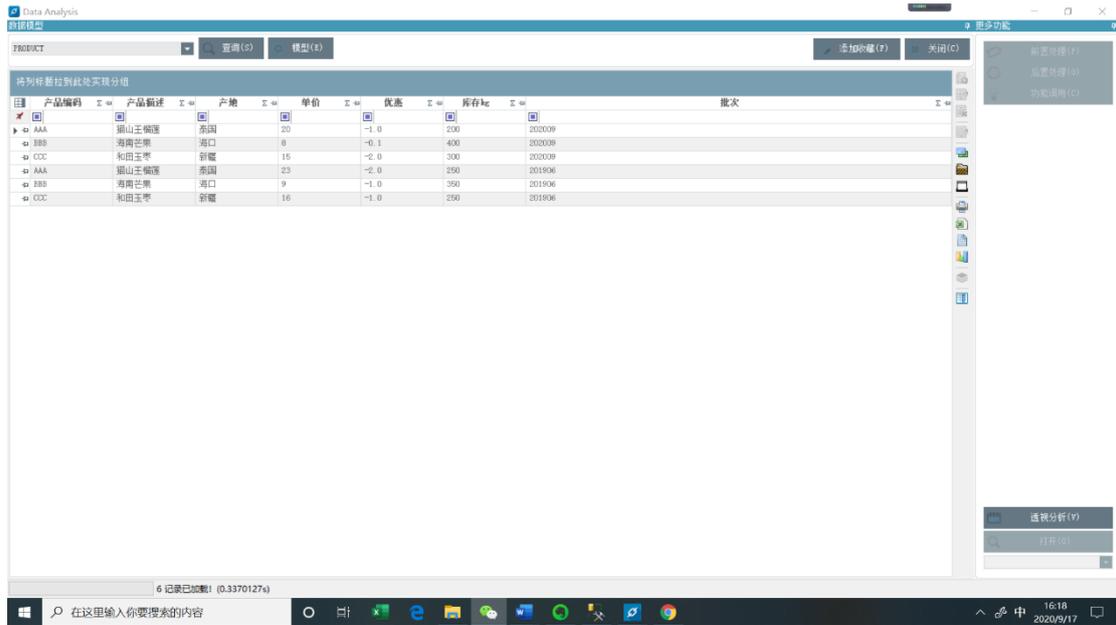
- 在数据模型新增查询模型做下拉列表：APPFUNCTION-PRODUCT



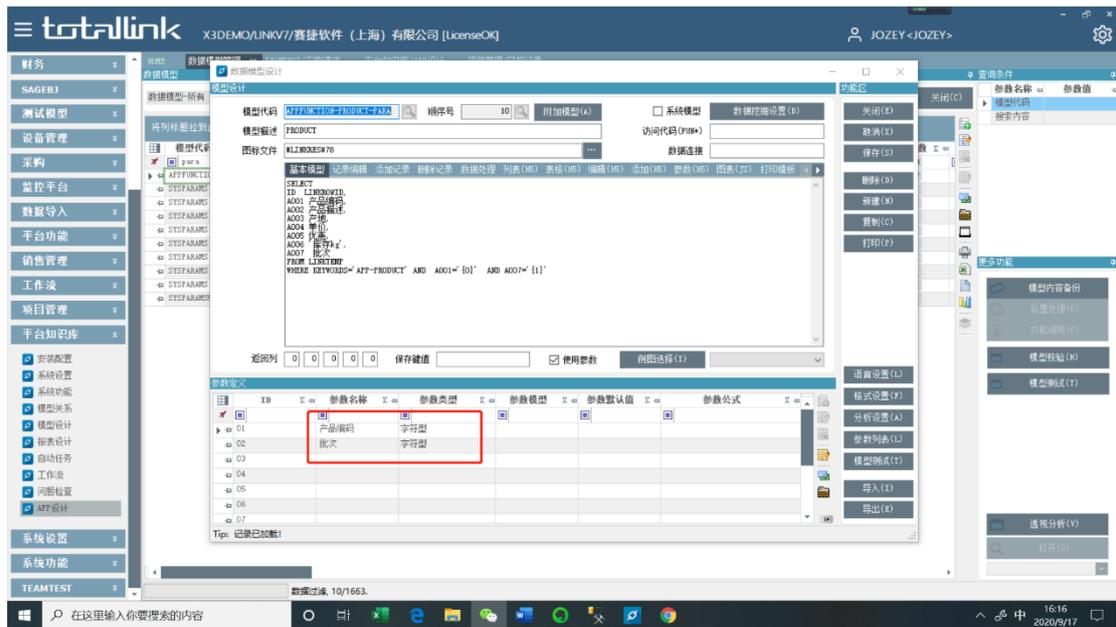
参考代码：

```
SELECT
A001 产品编码,
A002 产品描述,
A003 产地,
A004 单价,
A005 优惠,
A006 '库存 kg',
A007 批次
FROM LINKTEMP
WHERE KEYWORDS='APP-PRODUCT'
```

- 该模型查询结果如下：



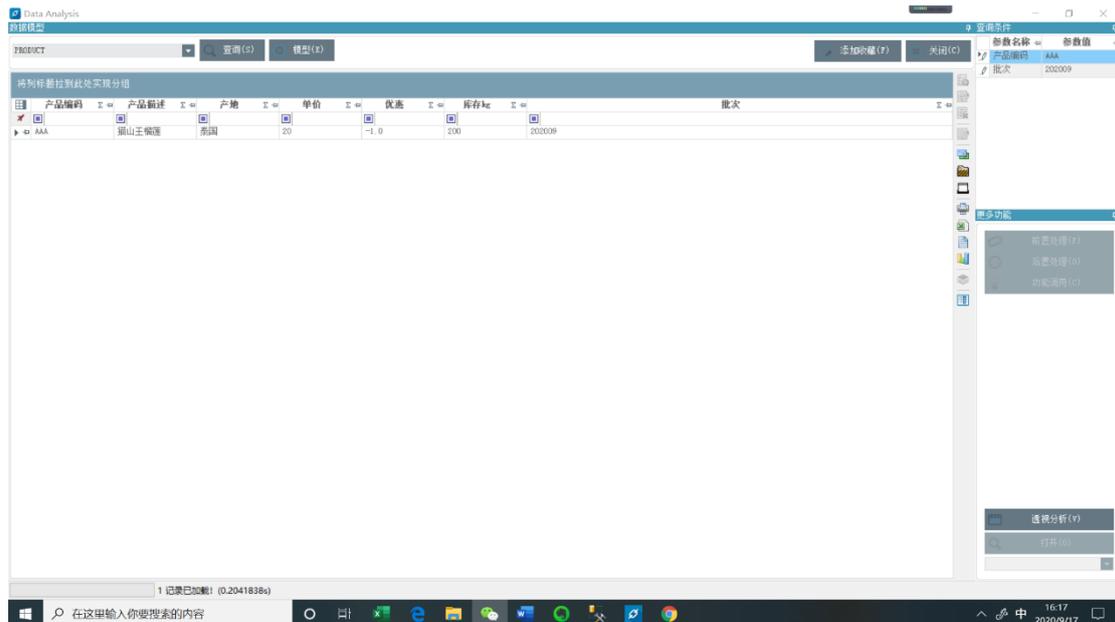
- 在数据模型新增查询模型做产品单价的来源：APPFUNCTION-PRODUCT-PARA



参考代码：

```
SELECT
ID LINKROWID,
A001 产品编码,
A002 产品描述,
A003 产地,
A004 单价,
A005 优惠,
A006 '库存 kg',
A007 批次
FROM LINKTEMP
WHERE KEYWORDS='APP-PRODUCT' AND A001='{0}' AND A007='{1}'
```

- 该模型输入参数后查询结果如下



- APP 测试
- 如下图，选择产品编码、批次、填入重量后点击第一个按钮，系统返回查询到的单价值



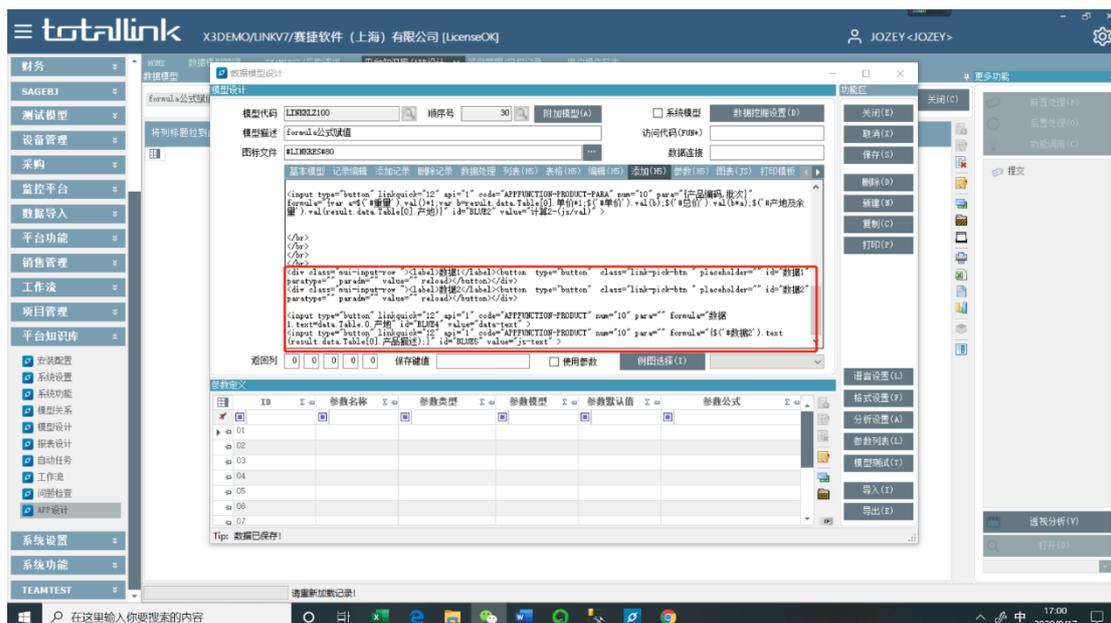
- 如下图，选择产品编码、批次、填入重量后点击第二个按钮，系统返回查询到的单价值、产地及余量、以及计算当前的总价



## 12.2 案例 2-text()

id.text=data.Table.x 与 formula="{\${(#id)}.text('值')}" 的应用

- 如图，添加（H5）页面添加代码



参考代码:

```

<div class="mui-input-row ">
<label>数据 1</label>
<button type="button" class="link-pick-btn " placeholder=""
id="数据 1" paratype="" paradm="" value="" reload></button>
</div>
<div class="mui-input-row ">
<label>数据 2</label>
<button type="button" class="link-pick-btn " placeholder=""
id="数据 2" paratype="" paradm="" value="" reload></button>
</div>

<input type="button" linkquick="12" api="1" code="APPFUNCTION-PRODUCT" num="10"
para="" formula="数据 1.text=data.Table.0.产地" id="BLUE4" value="data-text" >
<input type="button" linkquick="12" api="1" code="APPFUNCTION-PRODUCT" num="10"
para="" formula="{$('#数据 2').text(result.data.Table[0].产品描述);}" id="BLUE5"
value="js-text" >
    
```

- 调试结果：两种方式的调用结果相同
- 当点击第一个按钮时，系统获取查询模型第一行的产地字段值输出到“数据 1”



- 当点击第二个按钮时，系统获取查询模型第一行的产品描述字段值输出到“数据 1”

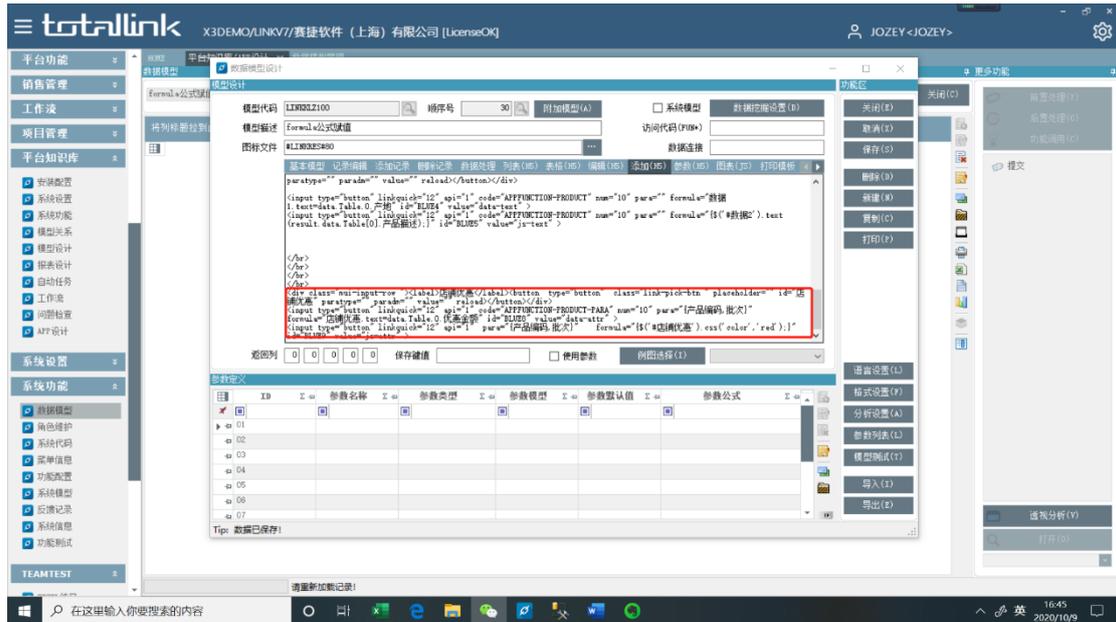
The screenshot shows the 'formula公式赋值' (Formula Assignment) screen in the TotalLINK mobile app. At the top, there is a '扫描输入区' (Scan Input Area). Below it are several input fields: '产品编码' (Product Code), '批次' (Batch), '重量' (Weight) with a placeholder '请输入重量' (Please enter weight), '单价' (Unit Price) with a placeholder '请输入单价' (Please enter unit price), '总价' (Total Price) with a placeholder '请输入总价' (Please enter total price), and '产地及余量' (Origin and Quantity) with a placeholder '请输入产地及余量' (Please enter origin and quantity). There are also two calculation buttons: '计算1-(val)' and '计算2-(js/val)'. Below these are two data entry fields: '数据1' (Data 1) with the value '泰国' (Thailand) and '数据2' (Data 2) with the value '猫山王榴莲' (Mao Shan Wang Durian), which is highlighted with a red box. Below these is a 'data-text' field with the value 'js-text', also highlighted with a red box. A '连续添加' (Continuous Add) toggle switch is set to 'ON'. At the bottom, there is a modal dialog box titled 'TotalLINK: Data Result from Data Model'. The bottom navigation bar contains three icons: '取消' (Cancel), '重置' (Reset), and '保存' (Save).

## 12.3 案例 3-attr()

`id.attr=data.Table.x` 与 `formula="{${'#id'}.attr('属性:值')}`" 的应用

场景：点击按钮 1，根据产品编码和批次，得到优惠价格；点击按钮 2，优惠字段显示红色

如图，添加（H5）页面添加代码



参考代码:

```

<div class="mui-input-row ">
<label>店铺优惠</label>
<button type="button" class="link-pick-btn " placeholder=""
id="店铺优惠" paratype="" paradm="" value="" reload></button>
</div>

<input type="button" linkquick="12" api="1" code="APPFUNCTION-PRODUCT-PARA"
num="10" para="{ 产品编码, 批次}" formula=" 店铺优惠.text=data.Table.0. 优惠金额"
id="BLUE8" value="data-attr" >
<input type="button" linkquick="12" api="1" para="{ 产品编码, 批次}"
formula="{ $('#店铺优惠').css('color','red');}" id="BLUE9" value="js-attr" >
    
```

- 调试结果: 输入产品编码和批次后, 点击第一个按钮, 店铺优惠获取到值为 1.2



下午4:47 1.7K/s 15%

formula公式赋值

扫描输入区

产品编码 AAA

批次 202009

重量 请输入重量

单价 请输入单价

总价 请输入总价

产地及余量 请输入产地及余量

计算1-(.val) 计算2-(js/val)

数据1

数据2

data-text js-text

店铺优惠 1.2

data-attr js-attr

连续添加 ON

取消 重置 保存

- 点击第二个按钮，店铺优惠值显示红色。

下午4:52 2.8K/s 4G 14

< formula公式赋值 >

扫描输入区

产品编码	AAA
批次	202009
重量	请输入重量
单价	请输入单价
总价	请输入总价
产地及余量	请输入产地及余量

计算1-(val) 计算2-(js/val)

数据1

数据2

data-text js-text

店铺优惠	1.2
------	-----

data-attr js-attr

连续添加  ON

取消 重置 保存

# 11 MQTT

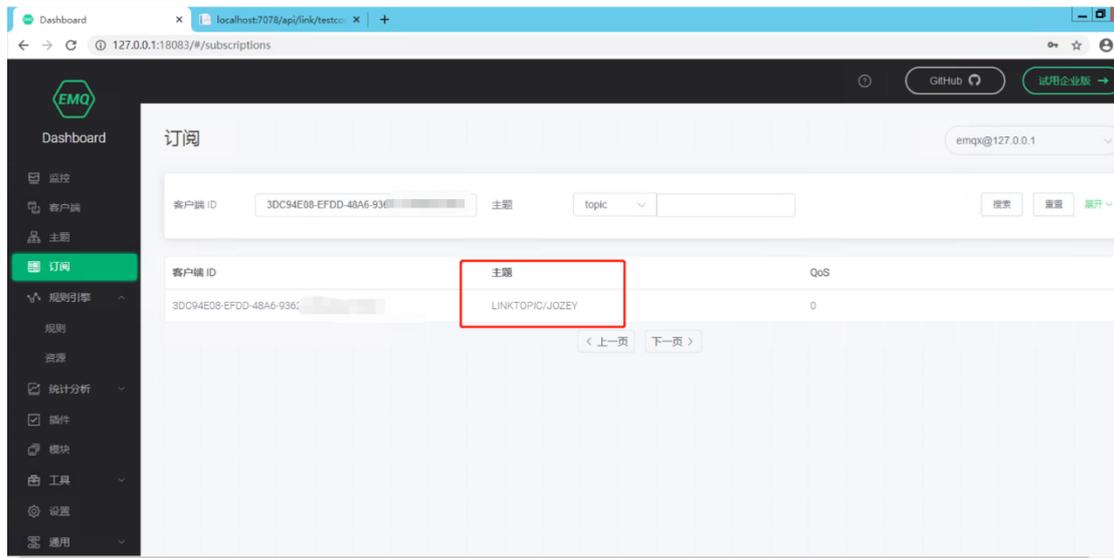
TotalLINK 新增 MQTT 功能，支持用户之间或人机之间实现数据通讯。

## 13.1 PC 端

目前 MQTT 支持两种模式，一种是发送纯文本消息，一种是发送指令和消息。用户每次登录客户端时支持订阅主题，订阅的主题仅在本次登录有效，若退出则失效。

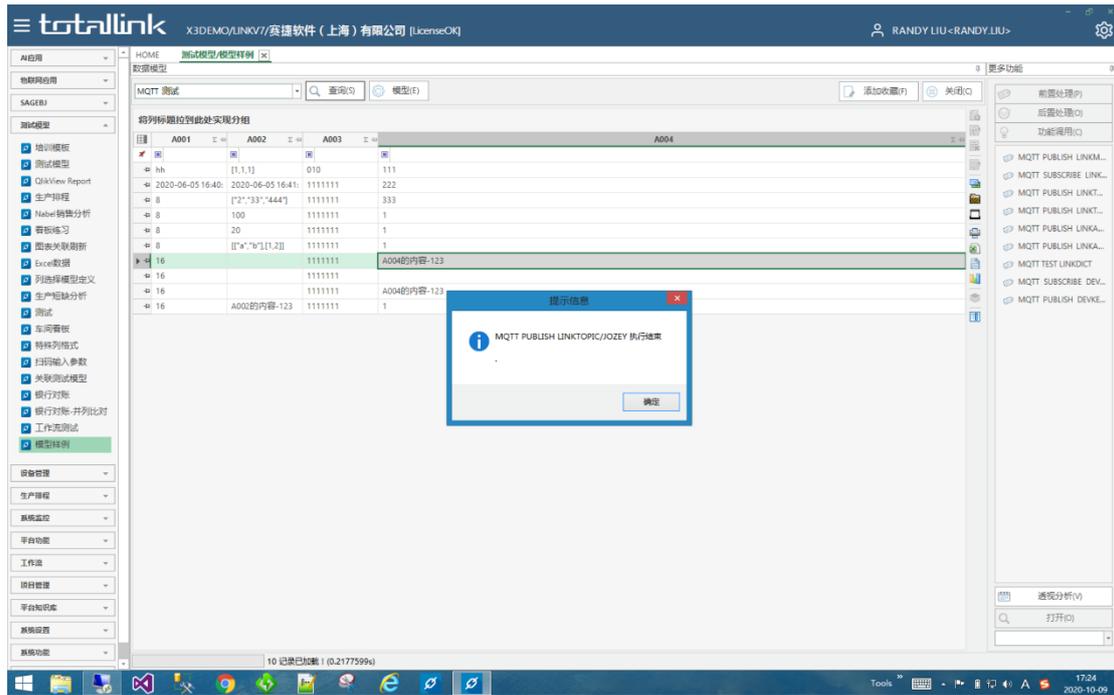
### 13.1.1 案例 1—给订阅默认主题的用户发送消息

无论哪个用户，在每次登录系统时，系统默认订阅一个该登录用户的主题。比如 jozey 登录了 TotalLINK PC 端，则默认订阅了 LINKTOPIC/JOZEY 的主题

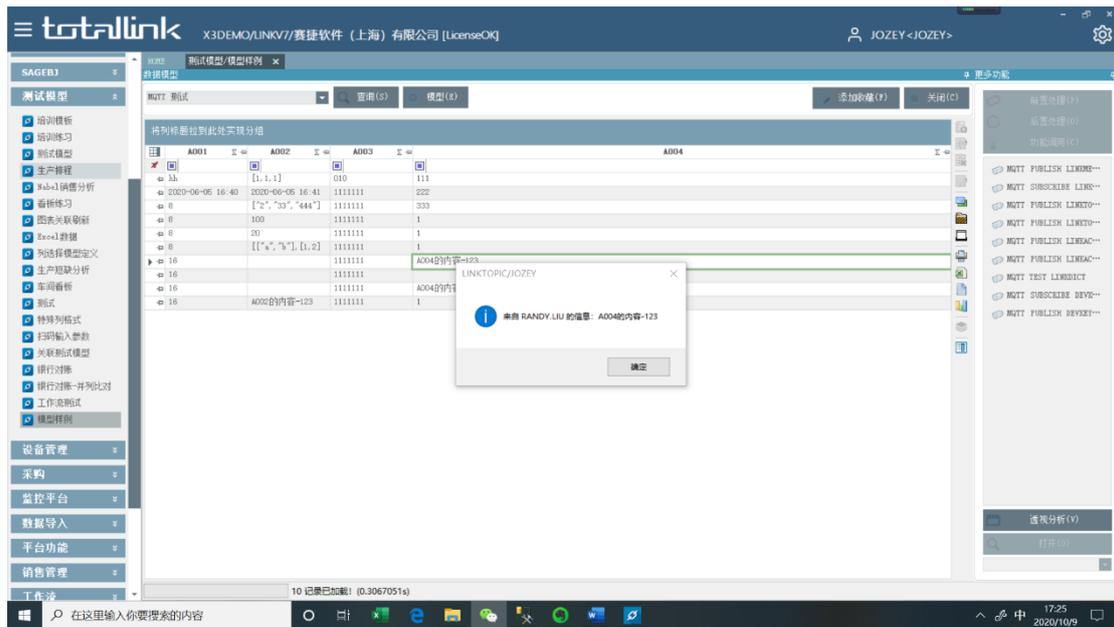


- 添加附加 504，在功能调用模块写入以下代码



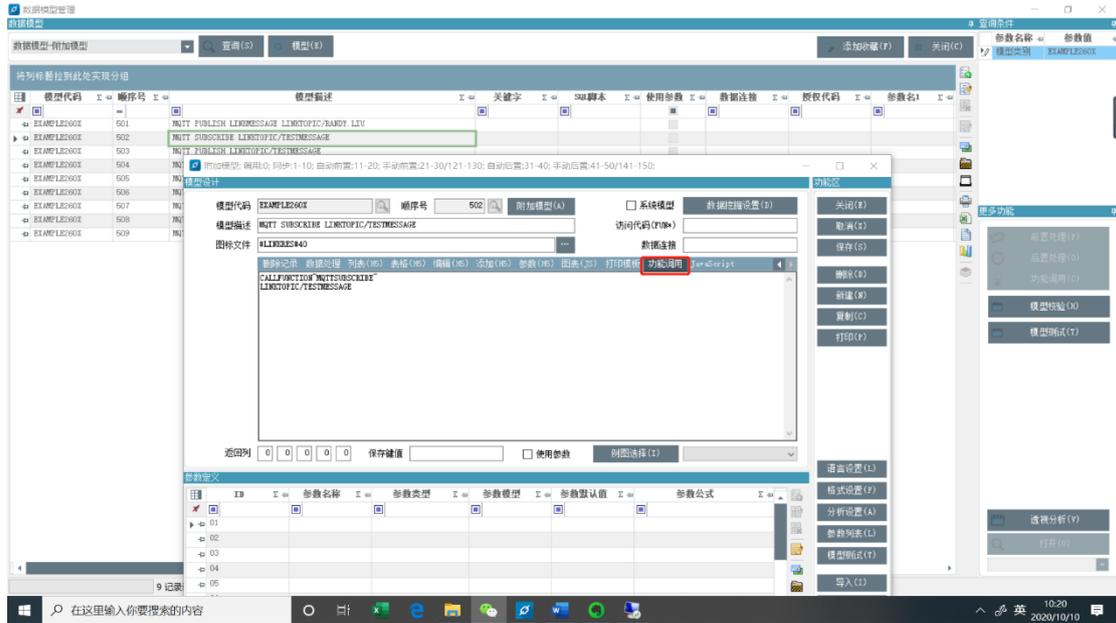


- Jozey 的客户端跳出提示



### 13.1.2 案例 2—订阅主题并发送消息

- 添加 502 附加模型，在功能调用模块加入以下代码



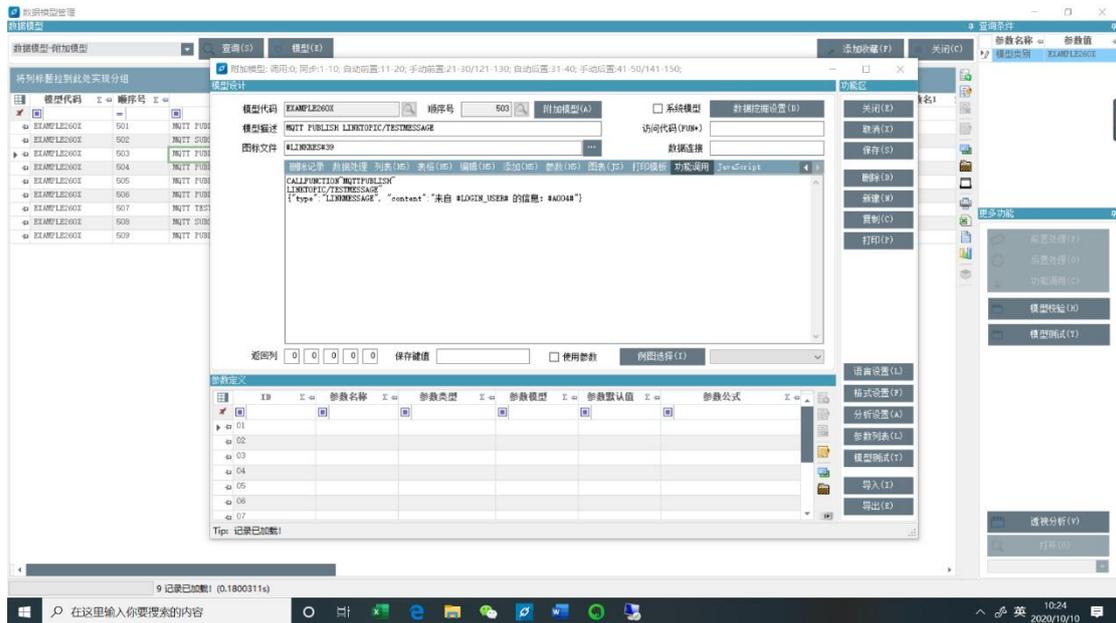
参考代码:

```
CALLFUNCTION~MQTTSUBSCRIBE~
LINKTOPIC/TESTMESSAGE
```

代码解析:

上面代码表示订阅 LINKTOPIC/TESTMESSAGE 主题

- 添加附加 503, 功能调用模块添加以下代码



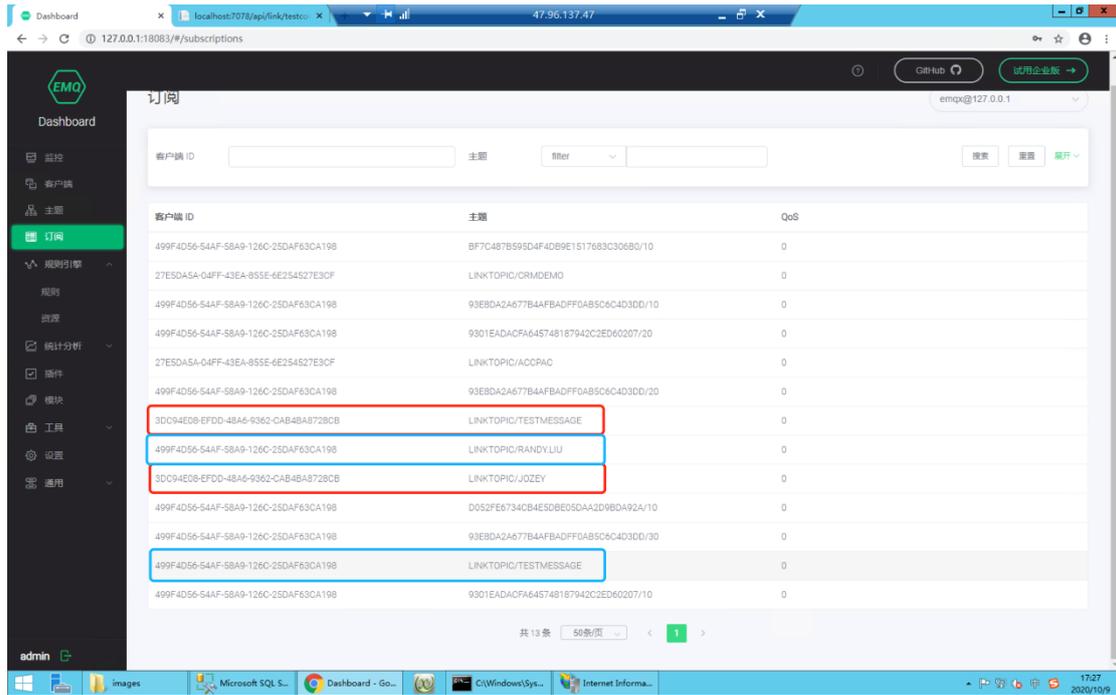
参考代码:

```
CALLFUNCTION~MQTTPUBLISH~
LINKTOPIC/TESTMESSAGE~
{"type": "LINKMESSAGE", "content": "来自 #LOGIN_USER# 的信息: #A004#"}

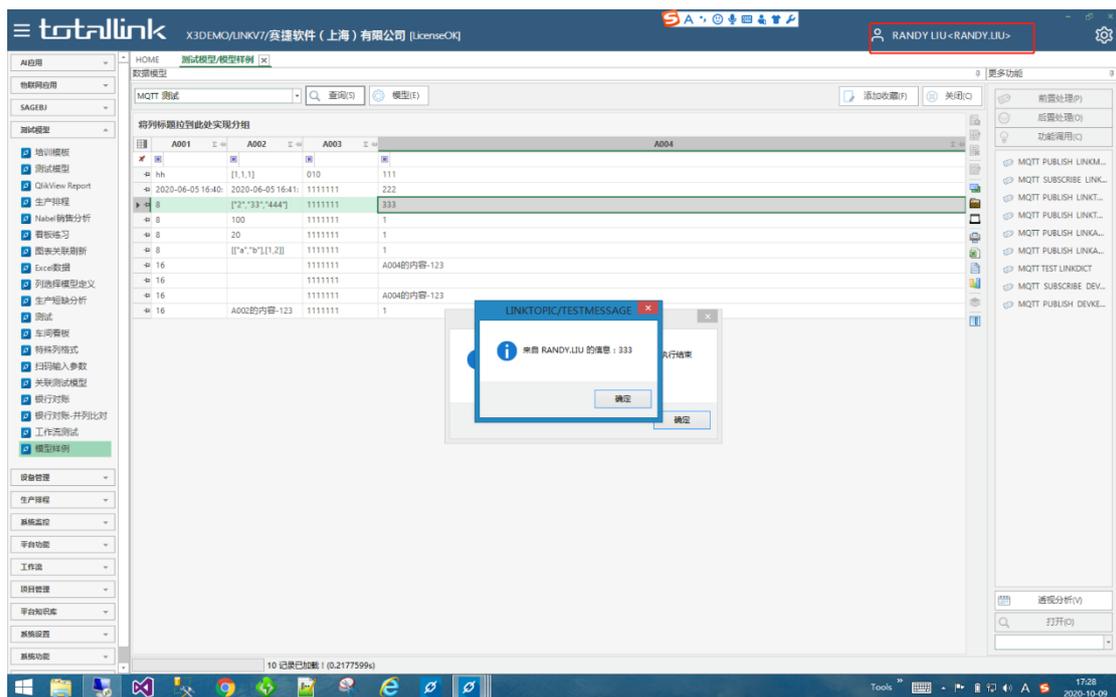
```

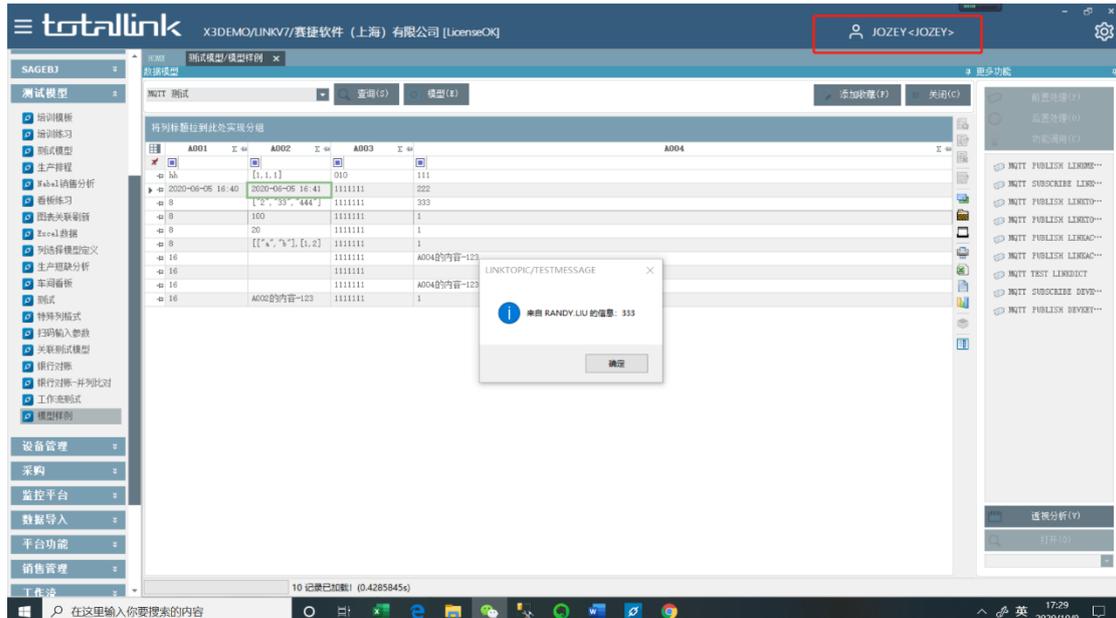
代码解析: 上面的代码表示给订阅 LINKTOPIC/TESTMESSAGE 主题的用户发送消息

- 用户 randyliu 和 jozey 分别在客户端执行附加 502, 订阅 LINKTOPIC/TESTMESSAGE 主题, 在 MQTT 链接网站可以查到订阅的情况



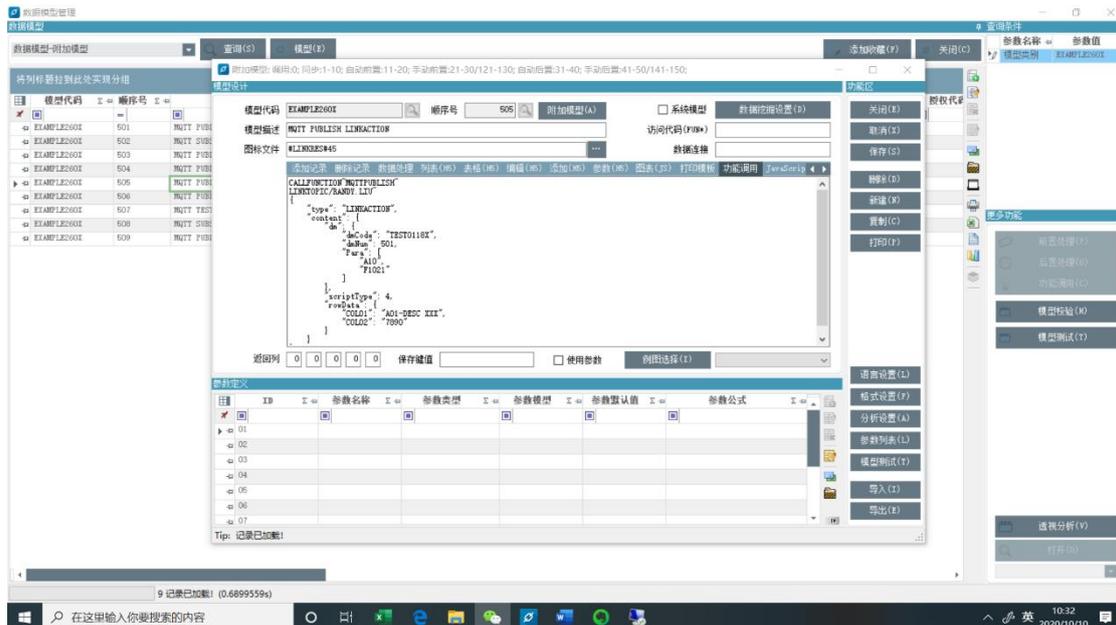
- 当 randyliu 用户执行附加 503, 给订阅 LINKTOPIC/TESTMESSAGE 主题的用户发送消息, 则 randyliu 和 jozey 的客户端会同时收到消息





### 13.1.3 案例 3—LINKADCTION 发送指令写法一

- 添加附加 505，在功能调用模块加入以下代码



参考代码:

```
CALLFUNCTION~MQTTPUBLISH~
LINKTOPIC/RANDY.LIU~
{
  "type": "LINKACTION",
  "content": {
    "dm": {
      "dmCode": "TEST0118X",
      "dmNum": 501,
      "Para": [
        "A10",
        "F1021"
      ]
    }
  }
}
```

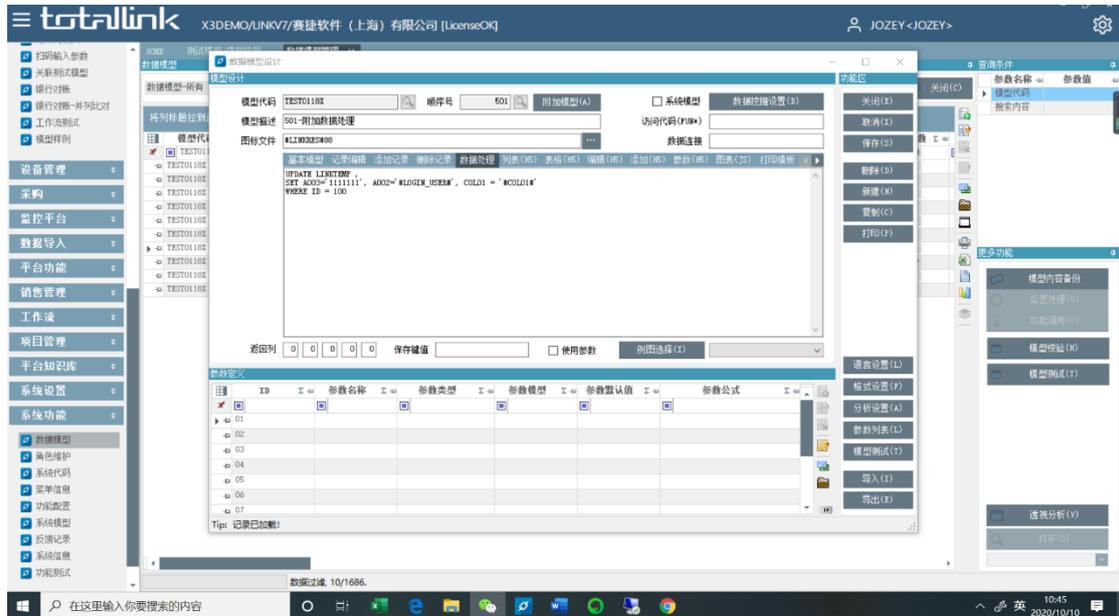
```

    }
  },
  "scriptType": 4,
  "rowData": {
    "COL01": "A01-DESC XXX",
    "COL02": "7890"
  }
}
}
}

```

代码解析:

- Type: 表示调用类型
  - dmCode: 模型代码
  - dmNum: 模型编号
  - Para: 传过去的模型参数
  - scriptType: 调用模型的模块, 0 基本模型, 1 记录编辑, 2 添加记录以此类推
  - rowData: 传过去的变量
- randyliu 执行上面的代码, 订阅 LINKTOPIC/RANDY.LIU 主题的用户客户端都会执行代码中模型数据处理的代码
  - 上面调用模型的代码



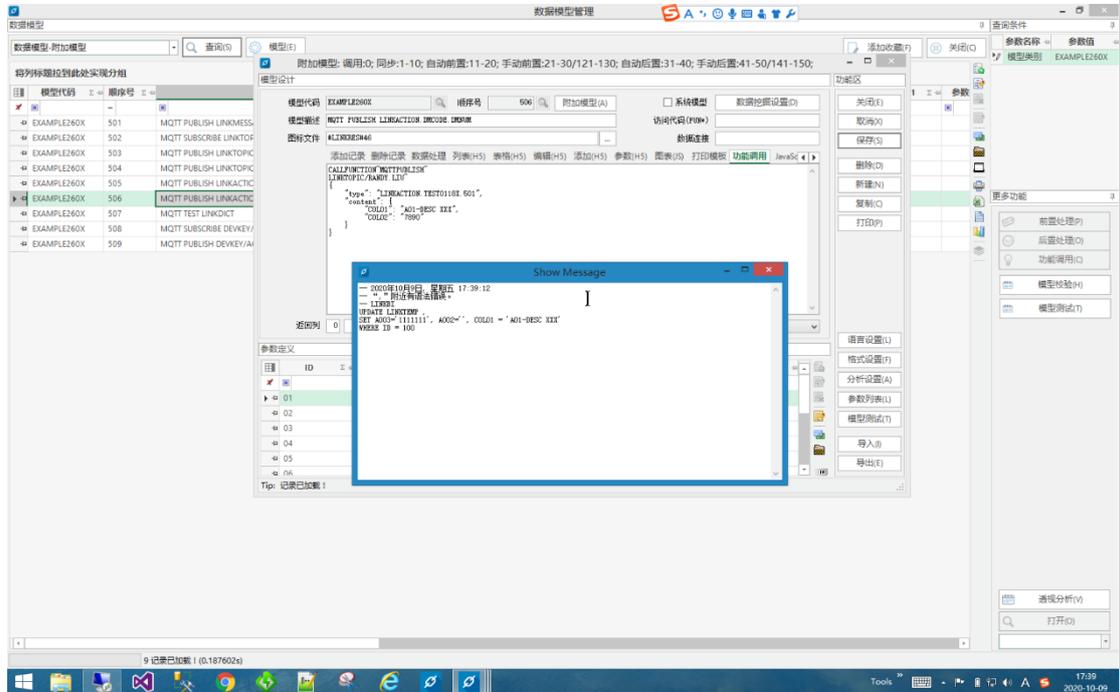
- 调用结果: 可以看到, 订阅 LINKTOPIC/RANDY.LIU 主题的用户收到了消息并执行了相应模型的代码, 自定义变量 COL01 的值也传到执行模型



```

"content": {
  "COL01": "A01-DESC XXX",
  "COL02": "7890"
}
    
```

- 调用结果：可以看到，订阅 LINKTOPIC/RANDY.LIU 主题的用户收到了消息并执行了相应模型的代码，自定义变量 COL01 的值也传到执行模型



## 13.2 APP 端

Linkquick=14, 表示链接 MQTT 协议

以下属性均为非必填

- publish 属性；publish 属性的值为空的时候不会发布消息
- topic 属性；订阅的主题
- reload 属性；是否每次点击都重新连接
- formula 属性；支持公式赋值功能，具体规则参考 formula 相关定义
- para 属性；可自定义相关参数，格式为数组，优先级比 publish 属性高，即当设置了 para 属性则发送 para 属性相关规则的内容
- port 属性；端口；默认 8083
- user 属性；用户名；默认为当前用户
- url 属性；服务地址；默认为当前服务地址
- off 属性；不订阅主题

### 13.2.1 案例 1--订阅主题

- 在模型中添加 H5 代码



参考代码:

```
<input type="text" class="mui-input-clear" placeholder="MQTT2 获取到的数据"
id="MQTT2" paratype="STR" paradm="" value="">

<input type="button" id="sdsdsb2" linkquick="14" publish=""
topic="LINKTOPIC/TESTMESSAGE" formula="MQTT2.val=result" value="MQTT 订阅">
```

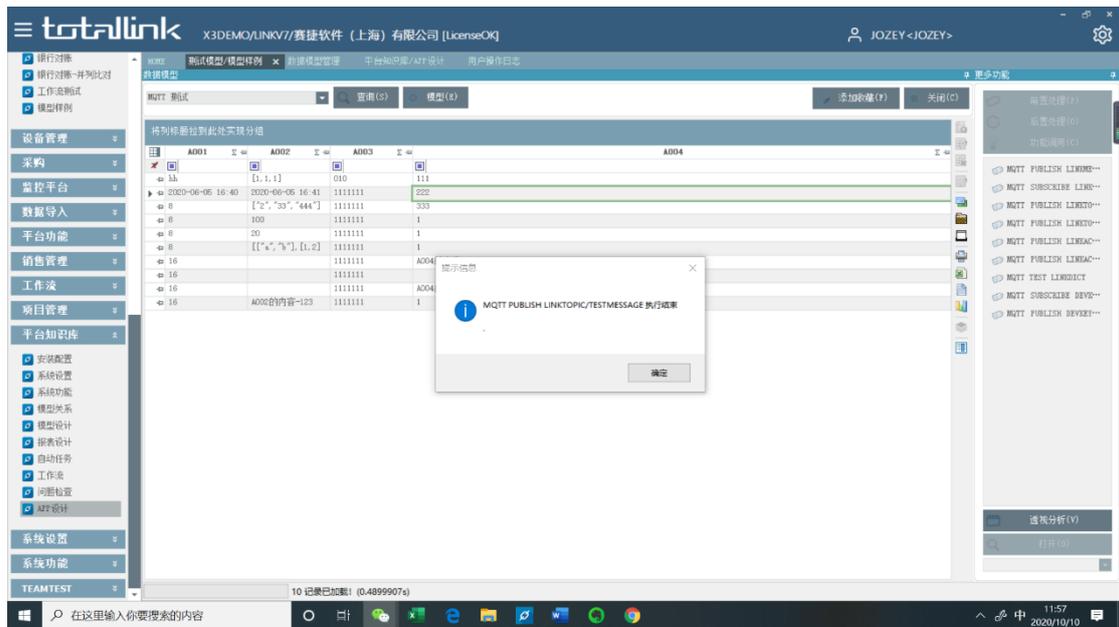
代码解析:

上面的"MQTT 订阅"按钮,红色部分表示订阅 LINKTOPIC/TESTMESSAGE 主题

- 在 APP 点击订阅按钮,可以看到 MQTT 链接网站新增一条订阅的数据

客户端 ID	用户名	IP 地址	心跳 (秒)	会话过期间隔 (秒)	当前订阅数量	连接状态	备注
APP_DINA	undefined	117.136.120.142.4...	60	0	1	● 已连接	202
APP_LORRYZHENG_...	undefined	120.33.148.173.46...	60	0	1	● 已连接	202
3DC094E08-EFDD-48...	JOZEY	180.166.150.27.99...	60	7200	1	● 已连接	202

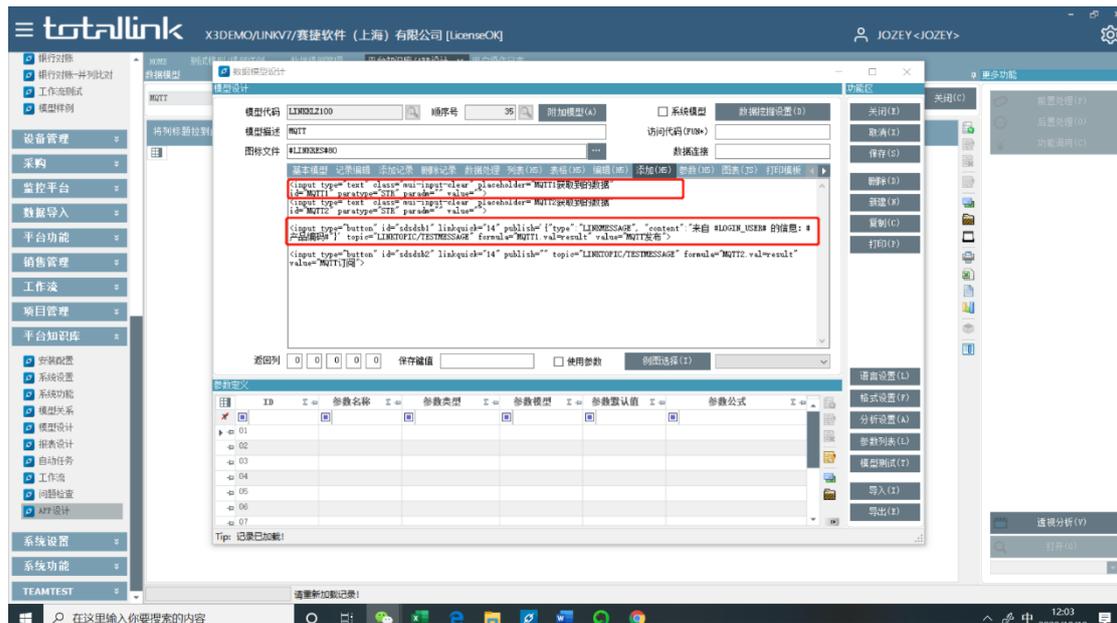
- 在客户端给 LINKTOPIC/TESTMESSAGE 主题发送消息,消息会显示在 app 端 id="MQTT2"的文本框





### 13.2.2 案例 2--给固定主题发送消息

- 在模型中添加 H5 代码





参考代码:

```
<input type="text" class="mui-input-clear" placeholder="MQTT1 获取到的数据"
id="MQTT1" paratype="STR" paradm="" value="">
<input type="button" id="sdsdsb1" linkquick="14" publish='{"type":"LINKMESSAGE",
"content":"来自 #LOGIN_USER# 的信息: #产品编码#}" topic="LINKTOPIC/TESTMESSAGE"
formula="MQTT1.val=result" value="MQTT 发布">
```

代码解析:

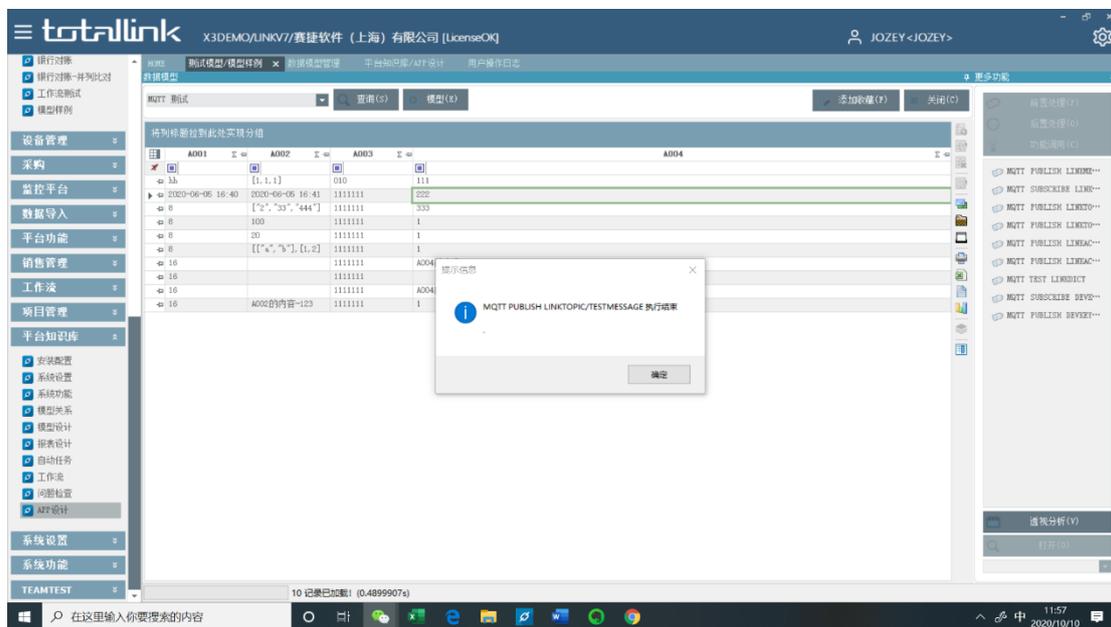
上面的"MQTT 发布"按钮,红色部分表示订阅 LINKTOPIC/TESTMESSAGE 主题并发送消息

- 在 dina 用户的 APP 点发布按钮,可以看到 MQTT 链接网站新增一条订阅的数据,并且文本框出现该按钮设定的消息

客户端 ID	用户名	IP 地址	心跳 (秒)	会话过期间隔 (秒)	当前订阅数量	连接状态	会话
APP_DINA	undefined	117.136.120.142.4...	60	0	1	● 已连接	202
APP_LORRYZHENG...	undefined	120.33.148.173.46...	60	0	1	● 已连接	202
3DC94E08-EFDD-48...	JOZEY	180.166.150.27.99...	60	7200	1	● 已连接	202



- 清空该文本框后，在客户端用户 jozey 给 LINKTOPIC/TESTMESSAGE 主题发送消息，消息会显示在 dina 的 app 端 id="MQTT1"的文本框





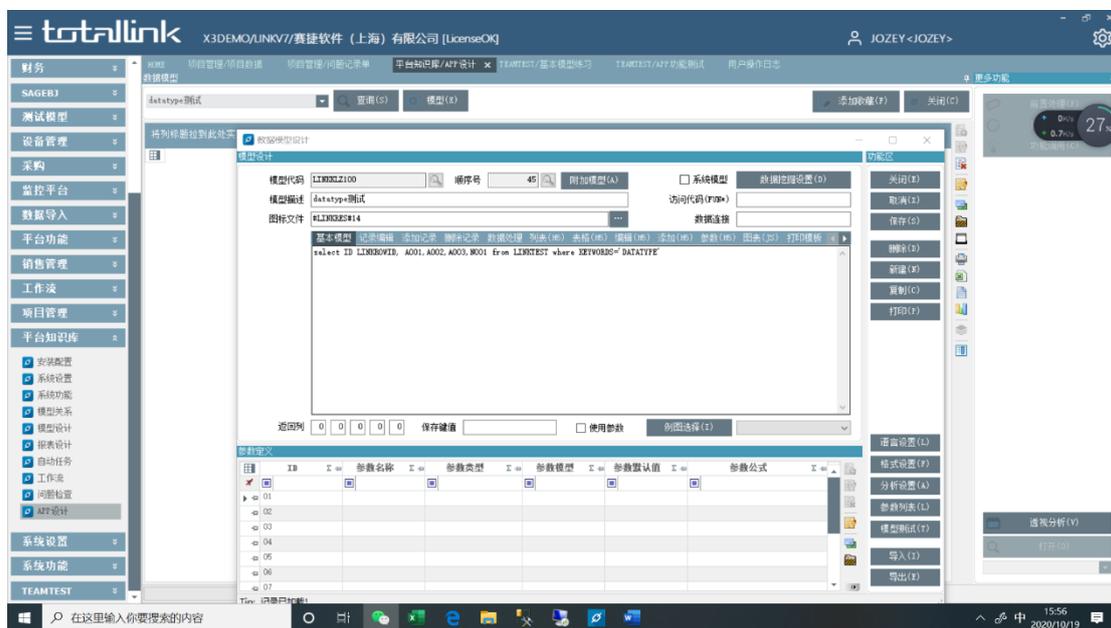
# 12 APP 数据类型控制

APP2.0.0.2 及之后版本新增数据类型自定义功能，通过在控件中加入 dataType 属性（部分类型可配合 other 属性使用），将用户在 APP 输入的值进行判断转换，然后再传进数据库表里。

- dataType="bool" bool 类型：''（空字符串） null undefined NaN 会转换成 false 其它都会转换成 true
- dataType="number-bool" number-bool 类型：先转 number 类型再转 bool 类型
- dataType="string" string 类型：字符串
- dataType="number" number 类型：数字
- dataType="float" float 类型：浮点型数字，默认保留两位小数，ps：可配合 other 属性使用，自定义保留几位小数，默认 other="2" 即保留两位小数
- dataType="array" array 类型：数组 ps：可配合 other 属性使用，自定义分隔的符号，默认 other="," 即逗号分隔的数组
- dataType="object" object 类型：对象

下面介绍 dataType 分别为 bool、number-bool、float 时 TotalLINK APP 给后台传的数据类型，其它类型待完善后实施人员后续补充例子

- 在数据库中，LINKTEST 表字段 A001, A002, A003 的数据类型是 nvarchar, N001 的数据类型是 decimal(16, 2)



- 在添加（H5）加入以下代码

模型代码: LINKKLZ100 顺序号: 45 附加模型(A)  系统模型 数据挖掘设置(D)

模型描述: datatype测试 访问代码(FUN\*)

图标文件: #LINKRES#14 数据连接

```

<div class="mui-input-row ">
<label>A001</label>
<input type="text" class="" placeholder="请输入 A001" dataType="bool"
id="A001" paratype="STR" paradm="" value="" />
</div>
<div class="mui-input-row ">
<label>A002</label>
<input type="text" class="" placeholder="请输入 A002" dataType="number-bool"
id="A002" paratype="STR" paradm="" value="" />
</div>
<div class="mui-input-row ">
<label>A003</label>
<input type="text" class="" placeholder="请输入 A003" dataType="float"
id="A003" paratype="STR" paradm="" value="" />
</div>
<div class="mui-input-row ">
<label>N001</label>
<input type="text" class="" placeholder="请输入 N001" dataType="number"
id="N001" paratype="STR" paradm="" value="" />
</div>
    
```

返回列: 0 0 0 0 0 保存键值  使用参数 例图选择(I)

ID	参数名称	参数类型	参数模型	参数默认值	参数公式
01					
02					
03					
04					
05					
06					
07					

参考代码:

```

<div class="mui-input-row ">
<label>A001</label>
<input type="text" class="" placeholder="请输入 A001" dataType="bool"
id="A001" paratype="STR" paradm="" value="" />
</div>
<div class="mui-input-row ">
<label>A002</label>
<input type="text" class="" placeholder="请输入 A002" dataType="number-bool"
id="A002" paratype="STR" paradm="" value="" />
</div>
<div class="mui-input-row ">
<label>A003</label>
<input type="text" class="" placeholder="请输入 A003" dataType="float"
id="A003" paratype="STR" paradm="" value="" />
</div>
<div class="mui-input-row ">
<label>N001</label>
<input type="text" class="" placeholder="请输入 N001" dataType="number"
id="N001" paratype="STR" paradm="" value="" />
</div>
    
```

- 在 APP 添加一条数据，APP 输入的值及 APP 传给数据库的值经过转换如下



```

"RDS": [
  {
    "Key": "A001",
    "Type": "text",
    "Value": false
  },
  {
    "Key": "A002",
    "Type": "text",
    "Value": false
  },
  {
    "Key": "A003",
    "Type": "text",
    "Value": "3.67"
  },
  {
    "Key": "N001",
    "Type": "text",
    "Value": "1.789"
  }
]
    
```

测试结果:

可以看到, A001 不输入值时给数据库传了 false; A002 输入值 0 时给数据库传了 false; A003 给数据库传了字符, 默认保留两位小数; N001 给数据库传了一个数字类型的数据

- 我们修改下 A003 和 N001 的代码

```

<div class="mui-input-row ">
<label>A003</label>
<input type="text" class="" placeholder="请输入 A003" dataType="float" other="1"
id="A003" paratype="STR" paradm="" value="">
</div>
<div class="mui-input-row ">
<label>N001</label>
<input type="text" class="" placeholder="请输入 N001"
id="N001" paratype="STR" paradm="" value="">
</div>
    
```

- 在 APP 再次输入数据, 并查看日志传递的结果



```

"RDs": [
  {
    "Key": "A001",
    "Type": "text",
    "Value": false
  },
  {
    "Key": "A002",
    "Type": "text",
    "Value": false
  },
  {
    "Key": "A003",
    "Type": "text",
    "Value": "1.7"
  },
  {
    "Key": "N001",
    "Type": "text",
    "Value": "456.96"
  }
],
    
```

测试结果:

可以看到, A003 保留了一位小数, N001 给数据库传递的值带引号的字符数据