



智能家居设备控制页面 XML 开发指南

版本号：V1.0

2017 年 8 月发布

中国电信股份有限公司上海研究院

目录

1 适用范围.....	3
2 XML 配置开发方式简介.....	4
3 标签分类.....	6
3.1 标签定义	6
3.2 布局和控件的通用属性列表	7
3.3 布局型标签	9
3.3.1 根标签: root.....	9
3.3.2 标签: linear	10
3.3.3 标签: frame.....	10
3.3.4 标签: block	10
3.3.5 标签: dialog	11
3.3.6 标签: commonDialog	12
3.3.7 标签: drop.....	13
3.3.8 标签: page	13
3.3.9 标签: view	14
3.4 监听器标签: listener.....	14
3.4.1 标签: listener	14
3.4.2 listener 执行条件	15
3.4.3 自定义 Action	16
3.5 控件型标签	18
3.5.1 标签: button.....	18
3.5.2 标签: image Button	18
3.5.3 标签: text.....	19
3.5.4 标签: edit.....	19
3.5.5 标签: image	19
3.5.6 标签: number.....	19
3.5.7 标签: dashboard.....	20
3.5.8 标签: Timer	21
3.5.9 标签: circleBar	21
3.5.10 标签: Roundbar	22
3.5.11 标签: SeekBar.....	23
3.5.12 标签: timeText.....	24
3.5.13 标签: list	25

3.5.14 标签: databaseOper	26
3.5.15 标签: chart 图表	26
4 修订记录.....	29

1 适用范围

本开发指南适用于中国电信智慧家庭 e-Link 开放平台架构中智能家居设备控制界面的设计及开发，是中国电信为智能家居应用开发者提供的设备控制界面开发、维护的技术依据。

本开发指南的适用者为中国电信智能家居业务、平台研发、运营人员，第三方智能家居应用开发人员。

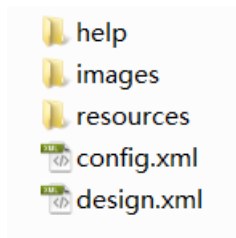
2 XML 配置开发方式简介

基于e-Link技术规范智能家居设备控制页面如果采用XML方式开发，需要遵循本开发指南定义的标签进行控制界面配置开发。本开发指南提到的所有XML配置都符合标准XML文件格式语法。

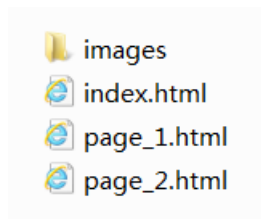
XML配置开发过程包含基础配置、界面配置、国际化配置、帮助文件配置四个大部分。

XML配置开发完成后需要对所有的配置进行打包发布，最终完成之后的文件命名和目录结构需要遵循如下规范：

目录结构：



- 1) **design.xml**: 设备的界面和功能配置文件。
- 2) **config.xml**: 设备的 LOGO、引导页、告警、通知等配置文件。
- 3) **resources 目录**: 存放所有支持的国际化文件。比如需要支持中文和英文，则需要在该目录生成两个文件，分别命名为 zh_CN.properties 和 en_US.properties，zh 和 en 是国际上通用的标准语言代码缩写，CN 和 US 是国际上通用的标准国家代码缩写。
- 4) **help 目录**: 存放设备的帮助信息界面用到的资源和文件。该部分内容以网页方式提供，提供一个 index.html 的主页面以及相应的子页面链接和用的资源。例如：



- 5) **images 目录**: 存放界面中所有用到的图片资源文件。

XML配置文件开发完成后，开发者需要对所有文件进行打包发布到智能家居云平台上。压缩包格式为标准的zip格式，命令按照如下规则：比如此时设备的厂家编号为HYL，产品是热水器编号为001，型号是H01，版本号是1.0.1，则打包的压缩包文件名称就是 HYL_001_H01_1.01.zip。登录到 <http://elink.189.cn/bm/>，将打包好的压缩包上传到相应的产品型号下面，并进行版本发布即可。

3 标签分类

完成配置的开发涉及到XML标签、每个标签有不同的属性，本开发指南对这些标签进行分类，方便开发者对标签的理解和使用：

- 基础控件标签

此类型涵盖了常用的基本控件类型，如view、button、text、edit、image、number、progress等，这些标签作为界面上的功能展现和响应者，一般不能包含其它子控件，仅可以包含监听器控件标签。

- 布局控件标签

布局型标签可以定义界面的布局，此类标签有root、linear、frame、block、dialog、commonDialog、drop、page等。此类型标签中可以嵌套布局控件标签、控基础控件标签以及监听器控件标签。

- 监听器标签

监听器类型用于捕获控件型及布局型标签的各种用户操作事件，如点击事件、滑动事件等。标签显示为listener，将会在后边详细描述。

- 关系型标签

定义页面的值关系脚本，该标签的定义为relation，其中定义了一组dvid值，用于定义页面的值和控件的关系，比如当编号为1的dvid属性（开关机属性），其值为1时表示电源开，为0时表示电源关闭，此时需要对该dvid进行配置，使值和界面关联起来，最终达到值驱动界面的效果。

3.1 标签定义

以下列出了所有系统支持的可配置的标签，如下所示：

标签	说明	分类	使用情况
root	根布局	布局	Yes
linear	线性布局	布局	Yes
frame	帧布局	布局	No
block	块布局	布局	Yes

dialog	对话框	布局	Yes
commonDialog	通用对话框	布局	Yes
drop	下拉框显示	布局	Yes
page	子页面	布局	Yes
view	基本控件	布局	Yes
floatMenu	浮动菜单	布局	Yes
button	按钮	控件	Yes
web	网页显示	控件	Yes
text	文本显示	控件	Yes
edit	编辑框	控件	Yes
image	图片	控件	Yes
number	滑动数字选择	控件	Yes
progress	圆盘进度条	控件	Yes
imageButton	图片按钮	控件	No
listener	监听器	监听器	Yes
relation	值关系配置	关系型	Yes
dvid	属性编号	关系型	Yes

3.2 布局 and 控件的通用属性列表

每个标签都必须有一定的属性字段，才能对标签进行详细的配置，本系统定了众多的标签，每个标签又有非常多的属性，但针对布局型标签和基础控件型标签，有非常多的通用属性，下面列出的是这些属性，如下所示：

属性	说明
name	● 标签名，全局唯一，不能重复，用于标识唯一一个控件。
x	● 布局左上角的横坐标，屏幕左上角坐标为 0,0（单位：像素）。
y	● 布局左上角的竖坐标，屏幕左上角坐标为 0,0（单位：像素）。
w	● 布局宽度（单位：像素）。

h	<ul style="list-style-type: none"> ● 布局高度（单位：像素）。
size	<ul style="list-style-type: none"> ● 字体大小。
padding	<ul style="list-style-type: none"> ● 设置布局内边距 left, top, right, bottom。 ● 如设置，必须全部设置四种属性。
visiable	<ul style="list-style-type: none"> ● 设置是否可见。 ● true 可见，默认为 true。 ● false，不可见（仍占用位置）。 ● gone 不可见（不占用位置）。
enable	<ul style="list-style-type: none"> ● 设置是否可用 ● true 可用，默认为 true ● false 不可用
gravity	<ul style="list-style-type: none"> ● 布局内控件所处位置 ● 值为 center 居中显示。 ● 值为 hcenter 水平居中。 ● 值为 vcenter 垂直居中。 ● 值为 bottom 居于底部。 ● 值为 top 居于上部。 ● 值为 left 居于左部。 ● 值为 right 居于右部。
weight	<ul style="list-style-type: none"> ● 布局之间占用的比重大小
bkImg	<ul style="list-style-type: none"> ● 设置背景颜色或图片。 ● 若以#开始则识别为颜色值，例 bkImg="#FFFFFF"。 ● 若不以#开始则系统认为背景为一张图片，例 bkImg="circle_small.png"。
text	<ul style="list-style-type: none"> ● 字体设置，如含有@符号调用国际化文件，否则直接显示当前字体，例如：text="00:00"，text="@set_speed"。

color	● 设置颜色属性。
style	● 设置字体样式，例如：style="bold"。
img	● 设置图片。
value	● 设置值，以@时支持国际化。例如 value="@windNone": 。
group	● 设置组，button 更改自身颜色时，在这个组中的其他 button 控件默认组的颜色

3.3 布局型标签

3.3.1 根标签：root

root型布局充当整个XML布局文件的最外环布，是布局的基础。布局的最外环要放置一个root标签，作为根标签，所有的标签都要嵌套进这个布局文件中。

其用法如下：

```
<root bkImg="#FFFFFF" designWidth="720" designHeight="1280">
  <!-- 顶部设置 -->
  <block h="75" bkImg="#f87f6a">
    .....
  </block>
  .....
</root>
```

属性如下：

属性	解释	是否必需
designWidth	<ul style="list-style-type: none"> ● 总布局宽度，像素为单位。 ● 具体值与 UI 提供的交互图宽度(像素)一致。 ● 仅限根布局，且必须有。 	是
designHeight	<ul style="list-style-type: none"> ● 总布局高度，像素为单位。 ● 具体值与 UI 提供的交互图的高度(像素)一致。 ● 仅限根布局，且必须有。 	是

bkImg	<ul style="list-style-type: none"> ● 设置背景颜色或图片。 ● 若以#开始则识别为颜色值,例 bkImg="#FFFFFF"。 ● 若不以#开始则系统认为背景为一张图片, 例 bkImg="circle_small.png"。 ● 定义与否均可。 	否
-------	---	---

3.3.2 标签: linear

linear型布局作为重要的内部布局使用,是布局的基础。可以嵌套其它标签使用。

其用法如下:

```
<linear direction="h" gravity="hcenter">
    <number name="water_number" width="200" margin="10 10
    10 10" min="0" max="24" value="5"/>
</linear>
```

下表为linear充当内部布局时的常用属性,同时支持通用属性:

属性名称	解释	是否必需
direction	<ul style="list-style-type: none"> ● 设置 linear 布局中控件的排放方向。 ● 没有设置则默认为垂直排放,值为 v 表示垂直排放,其它值表示水平排放。 	是
gravity	<ul style="list-style-type: none"> ● 参考前边通用属性配置。 	否

3.3.3 标签: frame

frame是本XML文件的支持的布局类型,但作为保留布局,并未运用。

3.3.4 标签: block

除去根标签,整个布局的排版是由紧密排布的 block 标签构成,然后 block 中填充控件型、监听器型布局。

用法如下:

```
<linear
```

```
name="root" fullWidth="720" fullHeight="1280" bkImg="#FFFFFF">
    <block h="75" bkImg="#f87f6a">
        .....
    </block>
    <block h="500" bkImg="#f87f6a">
        .....
    </block>
</linear>
```

下表为 block 布局的常用属性，同时支持通用属性：

属性名称	解释	必需
name	● 标签的识别名称，定义与否均可。	否
h	<ul style="list-style-type: none"> ● 标签布局高度，像素为单位。 ● 具体值与 UI 提供的交互图的高度(像素)一致。 ● 由屏幕左上角向下延伸高度 h。 	是

3.3.5 标签：dialog

此标签源于标签dialog，可在该布局下放置自定义布局，以弹出框的形式展现给用户。此标签只需放置在根标签下，dialog标签下再嵌套其它标签，与位置排放顺序无关，建议放在末尾。

用法如下：

```
<dialog name="speed_dialog">
    <linear bkImg="#FFFFFF" height="wrap" w="@450" padding="0 10
    0 0">
        <text text="@set_speed" height="150" size="36"
        style="bold" gravity="vcenter" padding="20 0 0 10"/>
        <view h="1" bkImg="#F8B7AC"/>
        <linear direction="h" gravity="hcenter">
            <number name="speed_number" width="200" margin="10
            10 10 10" min="0" max="24" value="5"/>
        </linear>
    </linear>
</dialog>
```

```

.....

</linear>

</linear>

</dialog>

```

下表为dialog的属性，同时支持通用属性：

属性名称	解释	必需
name	● 标签的识别名称。	是

3.3.6 标签：commonDialog

此标签为一种dialog，用户需要对点击事件及标题进行定义，可在该布局下放置自定义布局，以弹出框的形式展现给用户。此标签只需放置在根标签下，与位置排放顺序无关，建议放在末尾。

用法如下：

```

<!-- 水量设定对话框 -->
<commonDialog name="water_dialog" w="@450" height="wrap"
  title="@set_water">
  <linear direction="h" gravity="hcenter">
    <number name="water_number" width="200" margin="10 10 10
      10" min="0" max="24" value="5"/>
  </linear>
  <listener type="click">
    <if>
      <close name="water_dialog"/>
      <send cmd="opt" dvid="5" wait="@waitInfo"
        value="{water_number}"/>
    </if>
  </listener>
</commonDialog>

```

下表为其属性，同时支持通用属性：

属性名称	解释	必需
name	● 标签的识别名称。	是
title	● 设置标题。	否

3.3.7 标签：drop

此标签为自定义下拉页面，可在该布局下放置自定义布局，以下拉框的形式展现给用户。此标签只需放置在根标签下，与位置排放顺序无关，建议放在末尾。

用法如下：

```
<drop name="drop1" x="0" y="0" w="300">
    <item type="help" text="帮助信息"/>
    <item type="history" text="历史记录"/>
    <item type="feedback" text="意见反馈"/>
    <item type="define" text="自定义"/>
</drop>
```

下表为其属性，同时支持通用属性：

属性名称	解释	必需
name	● 标签的识别名称。	是
x	● 起始横坐标。	是
y	● 起始竖坐标。	是

3.3.8 标签：page

此标签为自定义二级界面，可在该布局下放置自定义布局，可以认为是二级界面的根布局。

用法如下：

```
<page name="test1">
    <view h="0" weight="1"/>
    <!-- 信号、火焰、水泵、风量 -->
    <block h="135">
        .....
```

```

</block>

<view h="0" weight="2"/>

</page>

```

下表为其属性，同时支持通用属性：

属性名称	解释	必需
name	● 标签的识别名称。	是

3.3.9 标签：view

view 标签基于 View，可以用于线条的布局等。

用法如下：

```

<view x="360" y="280" w="@450" h="1" bkImg="#fff8b7ac"/>

<view weight="1" h="0"/>

```

下表为其属性

属性名称	解释	必需
weight	● 所占比重 如果仅有一块 view 区的 可以设置 weight="1" 。 两块的话 比重多的设置为 weight="2" 。且后面必需紧跟 h="0"。	否

3.4 监听器标签：listener

3.4.1 标签：listener

listener 标签为定义的监听事件，嵌套在其它标签下并对其事件进行监听，同时 listener 标签下也嵌套了一系列标签，用于声明执行的事件，和执行的条件。

用法如下：

```

<listener type="click">
  <if>
    <close name="temp_dialog"/>
    <send cmd="opt" dvid="2" wait="@waitInfo"
      value="{tmp_number}"/>
  </if>

```

```
</listener>
```

属性名称	解释
click	● 设置为点击类型。
longClick	● 设置为长按类型。
down	● 点下时执行。
up	● 弹起时候执行。
move	● 移动时执行。

3.4.2 listener 执行条件

listener 执行时首先要做一个判断, 判断是否符合当前监听器执行的条件, 如符合则继续往下进行, 否则跳过此次监听。

listener 支持 if 标签判断, 并可以在 if 中加入 condition 判断是否符合条件, 用法如下:

```
<if condition="{3} eq 0">
    <tip tip="@speed_condition"/>
</if>
<else>
    <dialog name="speed_dialog"/>
</else>
```

if 标签中 condition 是判断是否符合条件的载体。

在上例中, 如果符合 if 中 condition 的条件, 则执行 tip 标签的语句, 否则执行 dialog 标签, 即调用定义好的 dialog, 将其显示出来。如果未定义 if 语句则直接执行事件。

condition 属性如下:

字段名称	解释
eq	● 等于。
lt	● 小于。
gt	● 大于。

le	● 小于等于。
ge	● 大于等于。
ne	● 不等于。
and	● 与。
or	● 或。
{}	● 取控件或 dvid 的值。

3.4.3 自定义 Action

监听器执行过程中对 condition 进行符合判断（如果有的话），然后执行相应的动作，这些动作就由 action 来定义。本 XML 中暂时支持的 action 有 tip、dialog、close、send、change。如有必要，可在后续可做相应扩展。

3.4.3.1 自定义 action: tip

tip 用于向用户弹出一条提示语，显示错误或者其他提示信息。

用法如下：

```
<tip tip="@speed_condition"/>
```

属性	说明
tip	● 弹出的提示语。

3.4.3.2 自定义 action: dialog

dialog 用于向用户弹出一个 dialog 框，这个 dialog 是在其它部位定义好的，通过其名字实现调用。

用法如下：

```
<dialog name="speed_dialog"/>
```

3.4.3.3 自定义 action: close

close 用于关闭用户打开的 dialog 框，通过其名字实现调用。

用法如下：

```
<close name="temp_dialog"/>
```

3.4.3.4 自定义 action: send

send 用于向用户弹出一条提示语，显示错误或者其他提示信息。

用法如下：

```
<send cmd="opt" dvid="5" wait="@waitInfo" value="{water_number}"/>
```

Send 标签定义了一些属性：

属性	说明
wait	● 等待时的提示语。
cmd	● 消息的类型。
value	● 消息的内容。
dvid	● 消息的字段编号。
user	● 消息的接收人。
actions	● 命令执行成功后的动作。

3.4.3.5 自定义 action: change

change 主要实现改变其它控件状态或者值的功能。

change 标签定义了一些属性：

属性	说明
targetView	● 目标控件。
gray	● 是否变灰。
visiable	● 是否可见。
value	● 值。
img	● 定义图片。
color	● 定义颜色。
size	● 定义字体大小。
style	● 定义字体样式。

3.4.3.6 自定义 action: drop

drop 用于打开下拉页面，通过名字实现调用。

用法如下：

```
<drop name="drop1" from="more"/>
```

3.4.3.7 自定义 action: page

page 用于打开二级页面，通过名字实现调用。

用法如下：

```
<page name="test1"/>
```

3.5 控件型标签

3.5.1 标签: button

标签button是本XML的重要组成部分之一，此标签基于控件Button。在button标签下，一般要定义一个name供其它标签识别，同时也可以通用属性。此标签下通常要定义一个相应的listener监听事件。

用法如下：。

```
<button name="powerSwitch" x="360" y="52" w="157" h="104"
  bkImg="main_power_on.png">
  <listener type="click">
    <if condition="{1} eq 1">
      <send cmd="opt" dvid="1" value="0" wait="@waitInfo"/>
    </if>
    <else>
      <send cmd="opt" dvid="1" value="1" wait="@waitInfo"/>
    </else>
  </listener>
</button>
```

3.5.2 标签: image Button

标签imageButton是本XML的重要组成部分之一，此标签基于控件ImageButton。在imageButton标签下，一般要定义一个name供其它标签识别，同时也可以通用属性。此标签下通常要定义一个相应的listener监听事件。

此标签用法类似于button标签，用法同上。

3.5.3 标签：text

标签text是本XML的重要组成部分之一，此标签基于控件TextView。text标签可以用通用属性。

用法如下：

```
<text x="455" y="147" w="110" h="113" text="°C" color="#ececec"
      size="100"/>
```

3.5.4 标签：edit

标签edit是本XML的重要组成部分之一，此标签基于控件EditText。edit标签可以用通用属性。

用法类似于text标签。

3.5.5 标签：image

标签image是本XML的重要组成部分之一，此标签基于控件ImageView。image标签可以用通用属性。

用法如下：

```
<image x="530" y="374" w="100" h="100" img="circle_small.png"/>
```

3.5.6 标签：number

标签number是本XML的重要组成部分之一，此标签基于控件NumberPicker。number标签常用属性如下表，可以用通用属性。此标签主要用来显示数字滑动表。

用法如下：

```
<number name="tmp_number" width="200" margin="10 10 10 10" min="0"
        max="100" value="5"/>
```

```
<number name="freeze_tmp_number" x="290" y="150"
w="200" h="320" centerSize="70" size="50" focus="true"
dividerColor="#c8c8c8" textColor="#555555"
dispArray="-26,-25,-24,-23,-22,-21,-20,-19,-18,-17,-16"
max="10"/>

<text text="°C" x="370" y="130" w="80" h="53"
size="50"/>
```

效果图:

空调将在



属性	说明
dividerColor	● 分割线颜色
centerSize	● 中间字体大小
dispArray	● 数组

3.5.7 标签: dashboard

标签dashboard用于仪表盘的显示，使用通用属性。

用法如下:

```
<dashboard name="airBoard" x="360" y="265" w="530" h="530"
value="90" dial="dial1.png" hand="hand1.png"
```

```
markColor="#ccffffff"/>
```

3.5.8 标签：Timer

标签 Timer 用于定时器控件，

用法如下：

```
<timer name="timer_remaind" interval="1" var="{9}" step="-1"
    end="0">
</timer>
```

其属性如下：

属性	说明
interval	● 最小值
var	● 最大值，当前值
step	● 步幅
end	● 结束条件
callback	● 结束后状态更改

3.5.9 标签：circleBar

标签 circleBar 用于圆形动态进度条，

用法如下：

```
<circleBar name="power" x="360" y="300" w="524" h="524" value="0"
    max="2400" startAngle="135" sweepAngle="270" outArcWidth="5"
    outArcColor="#4cffffff" onColor="#ffffff"
    offColor="#33ffffff"/>
```

效果图如下：



其属性表如下：

属性	说明
startAngle	● 代表初始角度
sweepAngle	● 代表扫描角度
outArcWidth	● 圆形外圈的宽度
outArcColor	● 圆形外圈的颜色
onColor	● 进度亮起时颜色
offColor	● 进度变暗时颜色

3.5.10 标签：Roundbar

标签 Roundbar 用于环形动态进度条。

用法如下：

```
<roundBar name="wash_time_roundBar" x="360" y="240" w="380"
  h="380" startAngle="-90" sweepAngle="360" value="100"
  touchable="false" inRadius="175" outRadius="180"
  pointer="count_down_thumb" pointerWidth="20"
  pointerHeight="20" bottomColor="#dddddd" color="#e96451"/>
```

效果图：



其属性如下：

属性	说明
startAngle	● 代表初始角度
sweepAngle	● 代表扫描角度
inRadius	● 内圈的半径
outRadius	● 外圈的半径
pointer	● 指针
touchable	● 可触摸 移动的
bottomColor	● 背景颜色 底部颜色

3.5.11 标签：SeekBar

标签 SeekBar 用于线性进度条。

用法如下：

```
<seekbar name="water_temperature_seekbar_six" x="360" y="20"
    w="660" h="36" min="0" max="100" pointer="thumb"
    pointerWidth="36" pointerHeight="36" progressHeight="6"
    color="#e96451" bkImg="#eeeeee" mark="mark_grey"
    markHeight="14" markWight="30" step="20">
```

效果图：



其属性如下：

属性	说明
pointer	● 指针图片
pointerWidth	● 指针宽度
pointerHeight	● 指针高度
progressHeight	● 线条的高度
mark	● 刻度图片
markHeight	● 图片高度
markWight	● 图片宽度
Step	● 移动步幅
tag	● tag 上的文字数字
tagImg	● tag 背景图 即数字显示部分的背景图
tagWidth	● tag 宽度
tagHeight	● tag 高度
mode	● tag 的出现方式 0 滑动时出现， 1 始终出现 ，一旦设置自行将 showTag 置为 true
showTag	● 为 true 时，显示方式根据 mode 改变；为 false 时不显示，优先级大于 mode
Color	● 进度条颜色值
bkImg	● 进度条背景

3.5.12 标签：timeText

标签 timeText 用于时间文本显示框。

用法如下：

```
<timeText name="time_text" x="360" y="190" w="200" h="64"
    gravity="center" text="--:--:--" color="#e8374a" size="50"
    format="h:m:s" fill="true"/>
```

效果图：

00:15:00

其属性表如下：

属性	说明
format	● 时间显示格式
fill	● 不足两位是否补零 例：01
fontFace	● 字体格式
filter	● 取值：false ， true 例： 00:00:01 隐藏时间为 0 的时间：如 "0 小时 3 分钟"，显示为"3 分钟"
gravity	● 对齐方式
prefix	● 前缀
suffix	● 后缀

3.5.13 标签：list

用法如下：

```
<list name="power_statistic_list" x="360" y="254" w="720" h="508"
    itemHeight="122" type="{_DID_}-PowerStatistic"
    sortColumn="DATA2" sortType="desc" bkImg="#ffffff,#f6f6f6"/>
```

效果图：



属性表：

属性	说明
itemHeight	● List 中单个模块的大小
sortColumn	● 排序字段

sortType	● 排序规则: asc,desc
type	● 当在数据库中取数据时, 数据类型标识
url	● 连接接口

3.5.14 标签: databaseOper

数据库操作 action, 返回 json 格式数据, 完成配置文件中对数据库的增、删、改、查操作。用法如下:

```
<databaseOper name="update_power_statistic" method="update"
  params="ID,DATA1"
  values="{statistic_id},{statistic_rename}"/>
```

属性表:

属性	说明
table	● 数据存储表
id	● 数据表主键
method	● 数据库操作方式, 四种: add,delete,update,query (查)
params	● 参数名称
values	● 参数值

3.5.15 标签: chart 图表

```
<chart name="statistic_chart" x="360" y="254" w="720" h="508"
  moveLimits="-0.05,1441,0,0" showAxes="true"
  pointStyle="TRIANGLE" xLabelNum="12" yLabelNum="6"
  seriesColor="#029ae5" type="line" margin="15 40 10 0"
  showXGrid="true" gridColor="#fefefe" fillBelowLine="true"
  fillBelowLineColor="#33029ae5" showChartValues="false"
  yLabelsAlign="left" marginColor="#ffffff" xTitle="(min) "
  yTitle="(W) " axesColor="#a00000" xLabelsColor="#999999"
  yLabelsColor="#00ff00" labelsColor="#0000ff"
  labelsTextSize="18" bkImg="#ffffff" xAxisMax="12"
```

```
yAxisMax="3" moveVertical="false" yAxisMin="0.1"/>
```

Chart 的效果图:

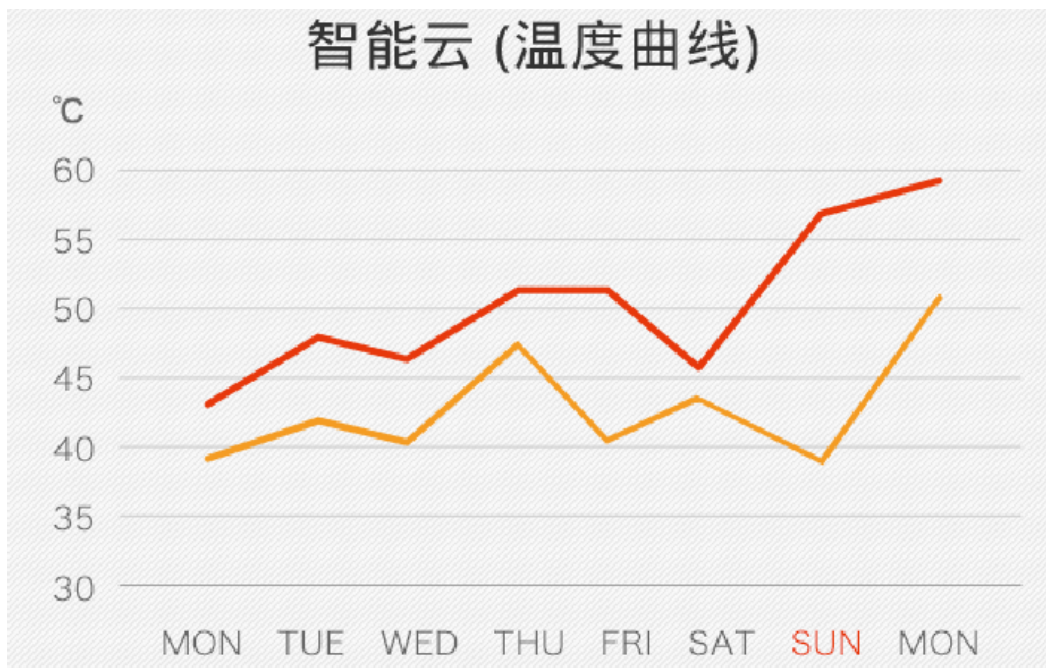


Chart 的属性表:

属性	说明
moveLimits	● 轴拖动的范围, 数组格式 : 10,1,2,20,代表[panMinimumX, panMaximumX, panMinimumY, panMaximumY]
showAxes	● 是否显示坐标轴
pointStyle	● type=line 时, 点的形状, 一共 6 种: X,CIRCLE,TRIANGLE,SQUARE,DIAMOND,POINT
xLabelNum	● x 轴显示刻度数
yLabelNum	● y 轴显示刻度数
seriesColor	● 每个系列的颜色
Type	● 图表类型: line-线型, bar-柱状, pie-饼型, 默认为柱状
margin	● 图表边距
showXGrid	● 显示 x 轴网格线
gridColor	● 网格线颜色
fillBelowLine	● 是否系列下端线填充
fillBelowLineColor	● 下端线填充颜色
yLabelsAlign	● y 轴刻度对齐方式
xLabelsAlign	● x 轴刻度对齐方式

marginColor	● 图表周边颜色
xTitle	● X 轴名称（单位）
axesColor	● 坐标轴颜色
xLabelsColor	● X 轴刻度颜色
yLabelsColor	● Y 轴刻度颜色
labelsColor	● X, y 轴名称颜色
labelsTextSize	● X, y 轴名称大小
xAxisMax	● X 轴显示刻度最大值
yAxisMax	● Y 轴显示刻度最大值
yAxisMin	● Y 轴显示刻度最小值
moveVertical	● x、y 轴拖动设置 是否可以垂直移动
moveHorizontal	● x、y 轴拖动设置 是否可以水平移动
zoomEnabledX	● x 是否可以缩放
zoomEnabledY	● y 是否可以缩放
xTextLabels	● x 所有刻度
space	● 柱间隔

4 修订记录

日期	修改描述	章节	修改类型 (增、删、改)	负责人