|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
| **设备控制FA开发指导书(个人开发者)** | | |  |
| **设备控制FA开发指导书(个人开发者)** | | |
| **文档版本** | **01** | |
| **发布日期** | **2021-05-25** | |
|  | | | | |
|  | **华为终端有限公司** | |  |  |

|  |
| --- |
| 版权所有 © 华为终端有限公司2021。 保留一切权利。  本材料所载内容受著作权法的保护，著作权由华为公司或其许可人拥有，但注明引用其他方的内容除外。未经华为公司或其许可人事先书面许可，任何人不得将本材料中的任何内容以任何方式进行复制、经销、翻印、播放、以超级链路连接或传送、存储于信息检索系统或者其他任何商业目的的使用。  商标声明  、、、华为，以上为华为公司的商标（非详尽清单），未经华为公司书面事先明示许可，任何第三方不得以任何形式使用。  注意  华为会不定期对本文档的内容进行更新。  本文档仅作为使用指导，文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |
| --- | --- |
| 华为终端有限公司 | |
| 地址： | 广东省东莞市松山湖园区新城路2号 |
| 网址： | <https://consumer.huawei.com> |

目 录

[1 设备控制FA开发指导书 1](#_Toc256000000)

[1.1 开发指南 1](#_Toc256000001)

[1.2 参考信息 7](#_Toc256000002)

[1.3 JSON文件配置说明 10](#_Toc256000003)

[1.3.1 Control类型控件配置说明 11](#_Toc256000004)

[1.3.1.1 Display控件 11](#_Toc256000005)

[1.3.1.2 Enum控件 17](#_Toc256000006)

[1.3.1.3 Reversal控件 21](#_Toc256000007)

[1.3.1.4 Increase\_Decrease控件 24](#_Toc256000008)

[1.3.1.5 Progress控件 26](#_Toc256000009)

[1.3.1.6 Multifunction控件 29](#_Toc256000010)

[1.3.2 Dialog类型控件配置说明 33](#_Toc256000011)

[1.3.2.1 枚举类型Dialog 34](#_Toc256000012)

[1.3.2.2 单选类型Dialog 36](#_Toc256000013)

[1.3.2.3 展示类型Dialog 38](#_Toc256000014)

[1.3.2.4 时间选择器类型Dialog 39](#_Toc256000015)

[1.3.2.5 日期选择器类型Dialog 41](#_Toc256000016)

[1.3.2.6 日期时间选择器类型Dialog 43](#_Toc256000017)

[1.3.2.7 多列文本选择器类型Dialog 45](#_Toc256000018)

# 设备控制FA开发指导书

[1.1 开发指南](#_ZH-CN_TOPIC_0000001158604283)

[1.2 参考信息](#_ZH-CN_TOPIC_0000001111964324)

[1.3 JSON文件配置说明](#_ZH-CN_TOPIC_0000001158604281)

## 开发指南

概述

碰一碰设备控制模板（OneHop Device Feature Ability(JS) ）用于帮助开发者实现**“碰一碰-设备配网-设备控制”**完整流程。默认工程运行后为智能电风扇的样例效果，开发者可以参考[定义设备控制FA的UI界面](#section737418403116)章节修改成其他自定义设备。

该模板工程主要分为两部分：

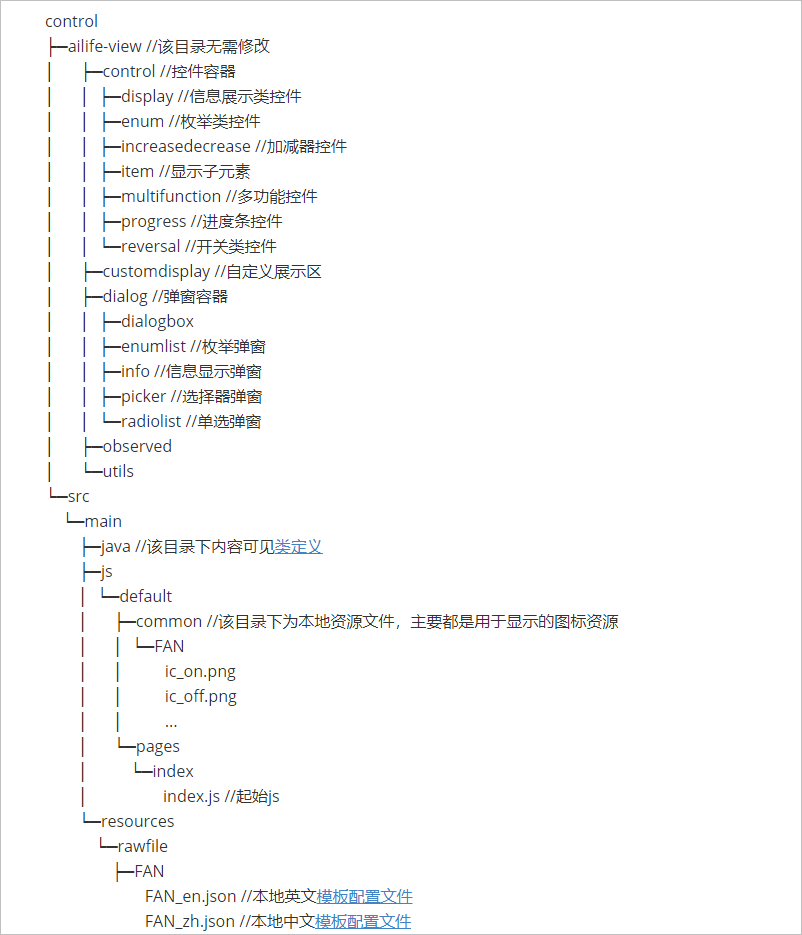
**设备配网FA （entry模块）**

**设备控制FA （control模块）**

本指导书用于指导开发者如何开发**设备控制FA的（control模块）**，关于**设备配网FA（entry模块）**的开发指导请参考设备配网FA开发指导书。

工程模板介绍

OneHop Device Feature Ability(JS)碰一碰设备控制模板的工程目录结构如下所示。



源码目录中各个类的说明如下表所示。

| Class | Description |
| --- | --- |
| **Java** | |
| MyApplication | 应用的入口类 |
| MainAbility | 拉起JS的入口类 |
| BaseDeviceDataHandler | 与设备数据交互的抽象类 |
| NetworkDeviceDataHandler | 继承 BaseDeviceDataHandler 抽象类，实现NAN无感配网通路的设备数据交互，开发者根据自己的设备进行修改 |
| SampleDeviceDataHandler | 继承 BaseDeviceDataHandler 抽象类，示例代码，开发者无需修改 |
| DataHandlerAbility | Java与JS交互消息的通道 |
| **JS** | |
| node\_modules/ailife-view | 家居控件， 可以在json配置文件声明引用 |
| index.js | 设备控制主页面，根据json配置文件加载ailife-view下的不同控件 |

向设备发送消息（NAN通道建立完成）

**JS接口：NetConfig.sendMessage(commonInfo, message, callbackFunc)**

依赖：需要import模板工程里的fa-netconfig.js进行调用

说明：commonInfo里需要填入本次配网通路的sessionId， 由配网FA获取

message为发送给设备的字符串消息

callbackFunc为消息发送成功与否的回调

示例：

sendMessage: function () {   
 let callbackFunc = function (value) {   
 //sendMessage 回调   
 };   
 let commonInfo = {   
 sessionId: getApp(this).ConfigParams.deviceInfo.sessionId   
 };   
 getApp(this).Netconfig.sendMessage(commonInfo, "111111", callbackFunc).then(   
 function (result) {   
 let ret = JSON.parse(result);   
 //sendMessage 消息下发结果   
 });   
}

**Java接口：NetConfigApi.*getInstance*().sendMessage(CommonInfo commonInfo, String message, BaseCallback<String> callback)**

依赖：在builde.gradle的dependencies中添加 implementation(group:'com.huawei.hilink',name:'ailifeability',version:'1.0.0.1',ext:'har')

说明：commonInfo里需要填入本次配网通路的sessionId， 由配网FA获取

message为发送给设备的字符串消息

callback为消息发送成功与否的回调

示例：

private void sendMessage(String message, HashMap<String, Object> tmpMap) {   
 CommonInfo commonInfo = new CommonInfo();   
 commonInfo.setSessionId(sessionId);   
 HiLog.error(LABEL\_LOG, "sessionId " + sessionId);   
 NetConfigApi.getInstance().sendMessage(commonInfo, message, (code, controlMessage, str) -> {   
 if (code == 0) {   
 HiLog.info(LABEL\_LOG, "send message success " + message);   
 dataMap.putAll(tmpMap);   
 mainHandler.postTask(() -> onDeviceDataChange(dataMap));   
 } else {   
 HiLog.error(LABEL\_LOG, "send message fail code = " +   
 code + " control message = " + controlMessage);   
 }   
 });   
}

定义设备控制FA的UI界面

**本章节用于指导开发者使用预设的组件快速构建出自定义设备控制页面，您也可以跳过json配置完全重写界面代码，并且使用上一节的设备控制接口给设备发送消息。**

该模板工程在control/node\_modules下提供了一系列的UI控件，同时定义了一套json文件规则来引用自定义UI控件来生成自定义的设备控制页面。

在control/src/main/js/default/pages/index/index.hml中相关控件引用代码如下：( 控件样式请参考 1.3JSON文件配置说明 章节 )



**使用模板的JSON配置方式自定义设备控制页面需要执行以下几个步骤：**

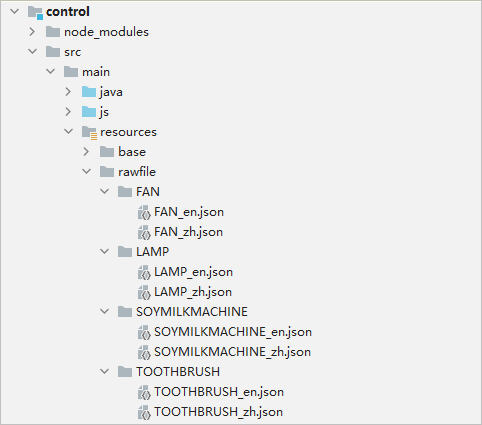
声明用于描述界面的json文件, 放置到 **control/src/main/resources/rawfile/<设备名>** 的目录下

json配置文件的详细字段定义见配置文件字段说明， 文件名必须为 <设备名>\_<语言>.json



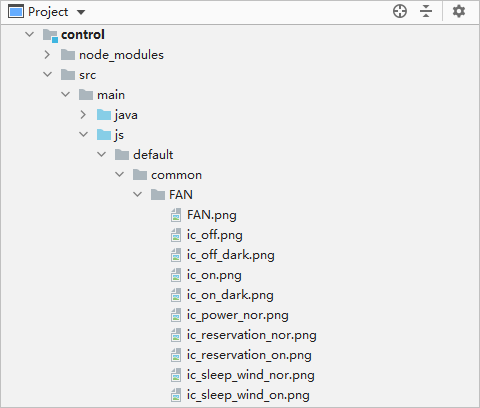
模板工程默认提供了台灯、牙刷、豆浆机 和 智能电风扇 4个样例，对应的配置文件为LAMP.json, TOOSHBRUSH.json，SOYMILKMACHINE.json，FAN.json

默认运行使用FAN.json



提供UX资源图，可本地打包也可云化。

使用本地资源需要统一打包到**control/src/main/js/default/common/<设备名>**的目录下，在json配置文件里可以直接引用。



使用云端资源需要在代码中添加资源的网络路径目录，具体修改到DataHandlerAbility.java中的getTemplate方法，将iconUrl填入您的网络路径， json配置文件中引用的资源是该网络路径下的相对路径。

if (result != null) {   
 // iconUrl is icon's prefix path. Here we use /common/productName in js module. You can change it into   
 // your cloud url. All icons file must be under iconUrl.   
 result.put("iconUrl", SampleDeviceDataHandler.EXAMPLE\_RESOURCE\_DIR + "/" + productName);   
 // deviceIcon is the product icon's file name. It must exist under iconUrl.   
 result.put("deviceIcon", "/" + productName + ".png");   
 // logoIcon is the product logo's file name. It must exist under iconUrl.   
 result.put("logoIcon", "/logo.png");   
}

将DataHandlerAbility.java中的DEVICE\_DATA\_MODE设为DEVICE\_DATA\_MODE\_NETWORK\_DEVICE，然后在NetworkDeviceDataHandler.java中添加数据处理逻辑。

参考NetworkDeviceDataHandler.java中的fanDataModel，模板中已经实现了一个智能电风扇的数据处理逻辑，您可以参照该电风扇模型添加您的自定义设备处理逻辑。

/\*\*   
 \* NetworkDeviceDataHandler is used to control Harmony devices. An example of a fan is provided as a reference.   
 \* Implement two methods, recoverData is used to initialize data, modifyProperty is used to respond to interface   
 \* operations. {@link NetworkDeviceDataHandler#sendMessage} is used to send messages to the device, After   
 \* sending successfully, {@link NetworkDeviceDataHandler#onDeviceDataChange(Map)} is called to refresh the UI interface   
 \*/   
public class NetworkDeviceDataHandler extends BaseDeviceDataHandler {   
 private static final HiLogLabel LABEL\_LOG = new HiLogLabel(HiLog.DEBUG, 0, "MessageDataHandler");   
 private static String sessionId = null;   
 private final DeviceDataModel dataModel;   
 private final EventHandler mainHandler;   
 private final Map<String, Object> dataMap = new HashMap<>();   
 private final DeviceDataModel fanDataModel = new DeviceDataModel() {   
 @Override   
 public Map<String, Object> recoverData() {   
 dataMap.put("switch/on", 0);   
 dataMap.put("windMode/mode", 0);   
 dataMap.put("windGear/gear", 0);   
 dataMap.put("shutdown/mode", 0);   
 return dataMap;   
 }   
 }   
}

想了解更多代码处理流程, 参考 [核心代码说明](#section17354153553419)

----结束

## 参考信息

核心代码说明

核心逻辑都在DataHandlerAbility.java中, 代码流程如下:

1. JS页面拉起后，会向Java侧请求模板配置数据。

JS代码：

async requestTemplate() {   
 // This function requests the JSON configuration used for display from Java   
 let action = utils.makeAction(ACTION\_MESSAGE\_CODE\_GET\_TEMPLATE, {});   
 let result = await FeatureAbility.callAbility(action);   
 let resultJson = JSON.parse(result);   
 if (resultJson.code == 0) {   
 let template = JSON.parse(resultJson.data);   
 this.parseJson(template.template);   
 }   
}

JAVA代码：

private boolean onRemoteRequest(int code, MessageParcel data, MessageParcel reply) {   
 switch (code) {   
 case ACTION\_MESSAGE\_CODE\_GET\_TEMPLATE: {   
 reply.writeString(getWriteTemplateString());   
 break;   
 }   
 default: {   
 reply.writeString("service not defined");   
 return false;   
 }   
 }   
 return true;   
}

1. Java侧读取模板配置文件：配置文件的路径在DataHandlerAbility的getTemplate中指定，文件来源可以是本地资源，也可以是从网络路径读取。

private ZSONObject getTemplate() {   
 // source indicates whether json config file is from the local file or cloud file.   
 final String source = "local";   
 // configDir indicates the directory path of json config file.   
 final String configDir = SampleDeviceDataHandler.EXAMPLE\_TEMPLATE\_FILE\_DIR + "/" + productName;   
 // configPath indicates the url path of json config file. You can change it into different url corresponding   
 // to your product name or device id.   
 final String configPath = configDir + "/" + productName + "\_" + language + ".json";   
 ZSONObject result = null;   
 if (LOCAL.equals(source)) {   
 try {   
 Resource resource = abilityContext.getResourceManager().getRawFileEntry(configPath).openRawFile();   
 byte[] tmp = new byte[1024 \* 1024];   
 int reads = resource.read(tmp);   
 if (reads != -1) {   
 String jsonString = new String(tmp, 0, reads, StandardCharsets.UTF\_8);   
 result = ZSONObject.stringToZSON(jsonString);   
 }   
 } catch (IOException e) {   
 HiLog.error(LABEL\_LOG, "failed to open template from local");   
 }   
 } else {   
 try {   
 URL url = new URL(configPath);   
 URLConnection urlc = url.openConnection();   
 urlc.connect();   
 try (InputStream is = urlc.getInputStream();   
 InputStreamReader isReader = new InputStreamReader(is, StandardCharsets.UTF\_8);   
 BufferedReader reader = new BufferedReader(isReader)) {   
 StringBuilder sb = new StringBuilder();   
 String str;   
 while ((str = reader.readLine()) != null) {   
 sb.append(str);   
 }   
 result = ZSONObject.stringToZSON(sb.toString());   
 }   
 } catch (IOException e) {   
 HiLog.error(LABEL\_LOG, "failed to open template from cloud");   
 }   
 }   
 if (result != null) {   
 // iconUrl is icon's prefix path. Here we use /common/productName in js module. You can change it into your   
 // cloud url. All icons file must be under iconUrl.   
 result.put("iconUrl", SampleDeviceDataHandler.EXAMPLE\_RESOURCE\_DIR + "/" + productName);   
 // deviceIcon is the product icon's file name. It must exist under iconUrl.   
 result.put("deviceIcon", "/" + productName + ".png");   
 // logoIcon is the product logo's file name. It must exist under iconUrl.   
 result.put("logoIcon", "/logo.png");   
 }   
 return result;   
}

1. JS读取到页面配置后，会向Java侧请求设备数据。

JS代码：

async getData() {   
 let action = utils.makeAction(ACTION\_MESSAGE\_CODE\_GET\_DATA, {});   
 await FeatureAbility.callAbility(action);   
}

JAVA代码：

private boolean onRemoteRequest(int code, MessageParcel data, MessageParcel reply) {   
 switch (code) {   
 case ACTION\_MESSAGE\_CODE\_GET\_DATA: {   
 deviceDataHandler.getData();   
 break;   
 }   
 default: {   
 reply.writeString("service not defined");   
 return false;   
 }   
 }   
 return true;   
}

1. Java侧进行初始化设备数据，调用BaseDeviceDataHandler#initProfileData返回设备所有profile数据。

NetworkDeviceDataHandler(String deviceId, DeviceDataCallback deviceDataCallback) {   
 super(deviceId, deviceDataCallback);   
 dataModel = fanDataModel;   
 mainHandler = new EventHandler(EventRunner.getMainEventRunner());   
 mainHandler.postTask(() -> initProfileData(0, "", dataModel.recoverData()));   
}

例如：模板样例智能电风扇的数据初始化如下。

private final DeviceDataModel fanDataModel = new DeviceDataModel() {   
 @Override   
 public Map<String, Object> recoverData() {   
 dataMap.put("switch/on", 0);   
 dataMap.put("windMode/mode", 0);   
 dataMap.put("windGear/gear", 0);   
 dataMap.put("shutdown/mode", 0);   
 return dataMap;   
 }   
}

1. JS界面加载完毕后，事件响应如下：事件触发后会调用到BaseDeviceDataHandler#modifyDeviceProperty，发送到设备端，此时UI不会有任何变化。等设备产生响应后，再触发BaseDeviceDataHandler#onDeviceDataChange回调变化后的数据到UI层，界面会自动刷新。

例如：模板样例智能电风扇发送消息给设备端的逻辑如下。

private void sendMessage(String message, HashMap<String, Object> tmpMap) {   
 CommonInfo commonInfo = new CommonInfo();   
 commonInfo.setSessionId(sessionId);   
 HiLog.error(LABEL\_LOG, "sessionId " + sessionId);   
 NetConfigApi.getInstance().sendMessage(commonInfo, message, (code, controlMessage, str) -> {   
 if (code == 0) {   
 HiLog.info(LABEL\_LOG, "send message success " + message);   
 dataMap.putAll(tmpMap);   
 mainHandler.postTask(() -> onDeviceDataChange(dataMap));   
 } else {   
 HiLog.error(LABEL\_LOG, "send message fail code = " +   
 code + " control message = " + controlMessage);   
 }   
 });   
}



* 当前模板只提供了从手机发送消息到设备侧的功能， 设备侧发送消息到手机暂不支持，相关能力后续逐步更新。

## JSON文件配置说明

模板中json配置文件为设备控制界面的主要依据，智能电风扇页面json文件路径为control/src/main/resources/rawfile/FAN，配置如下：

{   
 "devName": "智能电风扇",   
 "devType": "0",   
 "prodId": "FAN", -- prodId指定了控制界面上的产品图名称   
 "templateUIInfo": [ -- templateUIInfo下面可以堆砌模板提供的Control类型控件，以行为单位。   
 {"name": "电源"...},   
 {"name": "模式"...},   
 {"name": "档位"...},   
 {"name": "预约关机"...}   
 ],   
 "dialogUIInfo": [ -- dialogUIInfo下面可以声明Dialog类型控件，默认不显示， 由点击事件触发弹框   
 {"name": "关机时间"...}   
 ]   
}

目前模板里面提供了两种类型的控件（Control类型控件直接显示在界面上， Dialog类型控件需要触发后以弹窗形式展示）， 在json配置文件里声明不同控件即可。

**修改json配置文件后重新安装运行FA就能显示修改后的效果。**

### Control类型控件配置说明

Control类型控件一共有六个，分别是Display类型控件、Enum类型控件、Reversal类型控件、Progress类型控件、IncreaseDecrease类型控件和Multifunction类型控件。

#### Display控件

Display控件用于信息展示，支持显示两个或者三个区域内容，支持左右对齐（不带分割线） 或者 居中（带分割线）。

每个区域内容支持文字， 或者文字加图片（图片显示大小为 24vp \*24vp），文字颜色支持自定义，颜色和图片支持根据数据范围来选择。

示例图：



对应样例：

{   
 "gravity": "default",   
 "heightType": "default",   
 "name": "电量",   
 "span": 1,   
 "uiList": [   
 {   
 "subType": "text",   
 "textDisplay": {   
 "down": {},   
 "upLeft": {   
 "description": {   
 "path": "bluetoothConnectStatus/bluetoothConnectStatus",   
 "value": [   
 {   
 "scope": "0",   
 "target": "未连接"   
 },   
 {   
 "scope": "1",   
 "target": "已连接"   
 }   
 ]   
 },   
 "type": "dynamicText",   
 "value": "${bluetoothConnectStatus/bluetoothConnectStatus}"   
 },   
 "upRight": {}   
 }   
 },   
 {   
 "icon": {   
 "path": "infoDisplay/battery",   
 "value": [   
 {   
 "defaultTarget ": "/template/pic/battery\_10\_default.png",   
 "disableTarget ": "/template/pic/battery\_10\_default.png",   
 "scope": "-10",   
 "target": "/template/pic/battery\_10\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_20\_default.png",   
 "disableTarget ": "/template/pic/battery\_20\_default.png",   
 "scope": "10-20",   
 "target": "/template/pic/battery\_20\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_30\_default.png",   
 "disableTarget ": "/template/pic/battery\_30\_default.png",   
 "scope": "20-30",   
 "target": "/template/pic/battery\_30\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_40\_default.png",   
 "disableTarget ": "/template/pic/battery\_40\_default.png",   
 "scope": "30-40",   
 "target": "/template/pic/battery\_40\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_50\_default.png",   
 "disableTarget ": "/template/pic/battery\_50\_default.png",   
 "scope": "40-50",   
 "target": "/template/pic/battery\_50\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_60\_default.png",   
 "disableTarget ": "/template/pic/battery\_60\_default.png",   
 "scope": "50-60",   
 "target": "/template/pic/battery\_60\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_70\_default.png",   
 "disableTarget ": "/template/pic/battery\_70\_default.png",   
 "scope": "60-70",   
 "target": "/template/pic/battery\_70\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_80\_default.png",   
 "disableTarget ": "/template/pic/battery\_80\_default.png",   
 "scope": "70-80",   
 "target": "/template/pic/battery\_80\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_90\_default.png",   
 "disableTarget ": "/template/pic/battery\_90\_default.png",   
 "scope": "80-90",   
 "target": "/template/pic/battery\_90\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_99\_default.png",   
 "disableTarget ": "/template/pic/battery\_99\_default.png",   
 "scope": "90-100",   
 "target": "/template/pic/battery\_99\_default.png"   
 },   
 {   
 "defaultTarget ": "/template/pic/battery\_100\_default.png",   
 "disableTarget ": "/template/pic/battery\_100\_default.png",   
 "scope": "100-",   
 "target": "/template/pic/battery\_100\_default.png"   
 }   
 ]   
 },   
 "iconText": {   
 "type": "dynamicText",   
 "value": "${infoDisplay/battery}%",   
 "range": "100-0"   
 },   
 "subType": "iconTextHorizontal"   
 }   
 ],   
 "uiType": "DISPLAY"   
}

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String (1-128) | 当前控件显示名称 | 例如：“电量” |
| span | 必选 | float | 跨度 | 根据普通手机设置控件的宽度占比，为1表示占满普通手机的宽。 |
| uiType | 必选 | String(1-128) | 控件风格 | 控件风格为：“DISPLAY” ，“REVERSAL”,“INCREASE\_DECREASE”，“ENUM” ，“MULTIFUNCTION” |
| heightType | 可选 | String(1-128) | 高度类型 | “default”，“low”；默认为“default” |
| gravity | 可选 | String(1-128) | 子视图位置样式 | 样式分为：”default”,” “centerDivider”。默认为“default”，位置样式为俩边靠，无分隔线。” centerDivider”为等分居中且带分隔线。 |
| uiList | 必选 | DisplayItem[] | 子视图列表 | 子视图列表，具体根据实际显示需要配置对应个数的显示控件 |

子属性DisplayItem

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| subType | 必选 | String(1-128) | 子类型 | 纯文本，图标，图标+文本“text”，“icon”，“iconTextVertical”, “iconTextHorizontal” |
| icon | 可选 | DynamicContent | 动态颜色、图标，文本，根据path路径下数据的范围动态匹配对应的值。同时写icon和text时，形成icon上，文字下的上下结构 | "icon": {  "defaultIcon": "",  "path": "battery/useDay",  "disableIcon": "",  "value": [  {  "scope": "-50",  "target": ""  },  {  "scope": "50-100",  "target": ""  },  {  "scope": "100-",  "target": ""  }  ]  } |
| iconText | 可选 | NormalText/DynamicText | 图标描述文本 | "iconText":{  "color": "",  "type": "normalText",  "value": "电量"  } |
| textDisplay | 可选 | DiaplayContent | 文字模板，分为上左，上右和下。3处文字根据实际需求填写，分为普通文本类型NormalText和动态文本类型DynamicText | "textDisplay": {  "down": {  "color": "11FCFC",  "type": "normalText",  "value": "刷头可用>"  },  "upLeft": {  "color": {  "path":"battery/useDay",  "value": [  {  "scope": "-50",  "target": "11FCFC"  },  {  "scope": "50-100",  "target": "22FCFC"  },  { "  scope": "100-",  "target": "33FCFC"  } ]  },  "ratio": 1,  "type": "dynamicText",  "value":"${battery/useDay}"  },  "upRight": {  "color": "",  "type": "normalText",  "value": "天" }  } |

子属性DynamicContent

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| path | 必选 | String (1-128) | 动态内容条件值，可能需要动态获取设备数据填充 | 例如：“pm2p5/pm2p5Value” ，表示当前设备上报PM2.5的值 |
| value | 必选 | Variation | 动态范围值 | { "scope": "-50", "target": "11FCFC" } |

子属性Variation

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| scope | 必选 | String (1-128) | 条件范围 | 用“-”区分范围，有三种类型： 1，-50：表示小于等于50的值 2，50-100；表示大于50小于等于100； 3，100-：表示大于100的值 4，"scope": "-"时，表示可以匹配任意场景。 5，50：表示等于50 |
| target | 必选 | String (1-128) | 条件对应的值 | 可能是颜色值，资源值等，例如："11FCFC"，图标url |
| defaultTarget | 可选 | String (1-128) | 默认图标url | 默认图标url |
| disableTarget | 可选 | String (1-128) | 不可用图标url | 不可用图标url |

子属性NormalText

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| type | 必选 | String (1-128) | 类型字段 | 固定为NormalText |
| value | 必选 | String (1-128) | 当前控件显示名称 | 固定写死的值，例如："value": "天" |
| color | 可选 | String (1-128) | 当前控件显示颜色，固定值 | 固定写死的值，例如："11FCFC" |

子属性DynamicText

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| type | 必选 | String (1-128) | 类型字段 | 固定为DynamicText |
| value | 必选 | String (1-128) | 当前控件显示名称 | 例如：“${pm2p5/pm2p5Value}”注意：此处**${}**表示占位符，内部数据需要根据设备上报结果进行动态匹配 |
| ratio | 可选 | float | 补偿因子 | 例如：空气净化器检测PM2.5为500微克，实际页面上显示单位为毫克，显示内容为0.5，那么补偿因子为0.001，再选中页面数值后，控制的过程中，要再次除以补偿因子，进行命令下发。 |
| color | 可选 | DynamicContent | 当前控件显示颜色，动态值 | "color": {  "path": "battery/useDay",  "value": [  {  "scope": "-50",  "target": "11FCFC"  },  {  "scope": "50-100",  "target": "22FCFC"  },  {  "scope": "100-",  "target": "33FCFC"  }  ]  } |
| description | 可选 | DynamicContent | 当前控件描述转换，动态值 | "description": {  "path": "pm2p5/pm2p5Value",  "value": [  {  "scope": "-50",  "target": "优 "  },  {  "scope": "60-90",  "target": "良 "  },  {  "scope": "100-",  "target": "差 "  }  ]  } |

#### Enum控件

Enum控件用于类型选择，支持显示一行或两行区域内容，每行里面有一到四个区域内容，内容之间带分割线。每个区域内容由图片和文字构成，上面是图片，下面是文字。每个区域内容可以点击，点击后图片会发生改变，并只有一个区域内容处于被点击状态。

示例图：



对应样例：

{   
 "gravity": "centerDivider",   
 "heightType": "default",   
 "name": "模式",   
 "span": 1,   
 "command": [   
 {   
 "characteristic": "mode",   
 "enumInfo": [   
 {   
 "displayItem":{   
 "icon": {   
 "path": "lightMode/mode",   
 "value": [   
 {   
 "defaultTarget": "/ic\_read\_nor.png",   
 "scope": "-",   
 "target": "/ic\_read\_on.png"   
 }   
 ],   
 "valueNight": [   
 {   
 "defaultTarget": "/ic\_read\_nor\_dark.png",   
 "scope": "-",   
 "target": "/ic\_read\_on\_dark.png"   
 }   
 ]   
 },   
 "iconText": {   
 "type": "normalText",   
 "value": "阅读"   
 },   
 "subType": "iconTextVertical"   
 },   
 "value": 2   
 },   
 {   
 "displayItem": {   
 "icon": {   
 "path": "lightMode/mode",   
 "value": [   
 {   
 "defaultTarget": "/ic\_write\_nor.png",   
 "scope": "-",   
 "target": "/ic\_write\_on.png"   
 }   
 ],   
 "valueNight": [   
 {   
 "defaultTarget": "/ic\_write\_nor\_dark.png",   
 "scope": "-",   
 "target": "/ic\_write\_on\_dark.png"   
 }   
 ]   
 },   
 "iconText": {   
 "type": "normalText",   
 "value": "书写"   
 },   
 "subType": "iconTextVertical"   
 },   
 "value": 3   
 },   
 {   
 "displayItem": {   
 "icon": {   
 "path": "lightMode/mode",   
 "value": [   
 {   
 "defaultTarget": "/ic\_computer\_nor.png",   
 "scope": "-",   
 "target": "/ic\_computer\_on.png"   
 }   
 ],   
 "valueNight": [   
 {   
 "defaultTarget": "/ic\_computer\_nor\_dark.png",   
 "scope": "-",   
 "target": "/ic\_computer\_on\_dark.png"   
 }   
 ]   
 },   
 "iconText": {   
 "type": "normalText",   
 "value": "电脑"   
 },   
 "subType": "iconTextVertical"   
 },   
 "value": 4   
 },   
 {   
 "displayItem": {   
 "icon": {   
 "path": "lightMode/mode",   
 "value": [   
 {   
 "defaultTarget": "/ic\_favorite\_nor.png",   
 "scope": "-",   
 "target": "/ic\_favorite\_on.png"   
 }   
 ],   
 "valueNight": [   
 {   
 "defaultTarget": "/ic\_favorite\_nor\_dark.png",   
 "scope": "-",   
 "target": "/ic\_favorite\_on\_dark.png"   
 }   
 ]   
 },   
 "iconText": {   
 "type": "normalText",   
 "value": "最爱"   
 },   
 "subType": "iconTextVertical"   
 },   
 "value": 5   
 }   
 ],   
 "sid": "lightMode"   
 }   
 ],   
 "uiType": "ENUM"   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String (1-128) | 当前控件显示名称 | 例如：“亮度” |
| span | 必选 | float | 跨度 | 根据普通手机设置控件的宽度占比，为1表示占满普通手机的宽。 |
| uiType | 必选 | String(1-128) | 控件风格 | 控件风格为：“DISPLAY” ，“REVERSAL”,“INCREASE\_DECREASE”，“ENUM” ，“MULTIFUNCTION” |
| tag | 可选 | String(1-128) | 标记 | 扩展字段，用作标记特殊控件处理 |
| heightType | 可选 | String(1-128) | 高度类型 | “default”，“low”；默认为“default” |
| command | 必选 | Command[] | 控件对应的动作数组 | ACTION\_NORMAL在前面，ACTION\_ENUM在后面，下发命令为串行下发，先发完所有的ACTION\_NORMAL再发ACTION\_ENUM |
| gravity | 可选 | String(1-128) | 子视图位置样式 | 样式分为：”default”,” center”，“centerDivider”。默认为“default”，位置样式为俩边靠，无分隔线。” center”为等分居中，” centerDivider”为等分居中且带分隔线。 |

NormalCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| sid | 必选 | String (1-32) | 命令对应的服务 | 例如：“lightMode” |
| characteristic | 必选 | String (1-32) | 命令对应的属性 | 例如：“mode” |
| value | 必选 | int | 下发命令值 | 例如："value": 1 |

EnumCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| sid | 必选 | String (1-32) | 命令对应的服务 | 例如：“lightMode” |
| characteristic | 必选 | String (1-32) | 命令对应的属性 | 例如：“mode” |
| value | 必选 | int | 下发命令值 | 例如："value": 1 |

EnumInfo

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| displayItem | 可选 | DisplayItem | 通用展示子视图 | 通用展示子视图 |
| value | 必选 | int | 命令对应的服务属性值 | 例如：4 |
| disable | 可选 | String[] | 命令触发后禁用的标签数组 | 例如：["模式", "亮度"]，例如：空气净化器关闭的时候，禁用睡眠模式。 |

#### Reversal控件

Reversal控件用于开关选择，支持显示两个或三个区域，中间的区域可以不显示，左边和中间的区域内容支持文字， 或者文字加图片（图片显示大小为 24vp \*24vp），文字颜色支持自定义，颜色和图片支持根据数据范围来选择。右边的区域内容用于显示开关图片。

示例图：



对应样例：

{   
 "gravity": "default",   
 "heightType": "default",   
 "name": "电源",   
 "span": 1,   
 "command": [   
 {   
 "characteristic": "on",   
 "reversal": [   
 {   
 "disableIcon": "/ic\_power\_nor.png",   
 "icon": "/ic\_on.png",   
 "disableIconNight": "/ic\_power\_nor\_dark.png",   
 "iconNight": "/ic\_on\_dark.png",   
 "value": 1   
 },   
 {   
 "disable": {   
 "name": [   
 "模式",   
 "亮度"   
 ]   
 },   
 "disableIcon": "/ic\_power\_nor.png",   
 "icon": "/ic\_off.png",   
 "disableIconNight": "/ic\_power\_nor\_dark.png",   
 "iconNight": "/ic\_off\_dark.png",   
 "value": 0   
 }   
 ],   
 "sid": "switch"   
 }   
 ],   
 "displayItemOne": {   
 "subType": "text",   
 "textDisplay": {   
 "down": {},   
 "upLeft": {   
 "description": {   
 "path": "switch/on",   
 "value": [   
 {   
 "scope": "0",   
 "target": "已关闭"   
 },   
 {   
 "scope": "1",   
 "target": "已开启"   
 }   
 ]   
 },   
 "type": "dynamicText",   
 "value": "${switch/on}"   
 },   
 "upRight": {}   
 }   
 },   
 "uiType": "REVERSAL"   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String (1-128) | 当前控件显示名称 | 例如：“Battery” |
| span | 必选 | float | 跨度 | 根据普通手机设置控件的宽度占比，为1表示占满普通手机的宽。0.5表示占普通手机一半的宽。 |
| uiType | 必选 | String(1-128) | 控件风格 | 控件风格为：“DISPLAY” ，“REVERSAL”,“INCREASE\_DECREASE”，“ENUM” ，“MULTIFUNCTION” |
| tag | 可选 | String(1-128) | 标记 | 扩展字段，用作标记特殊控件处理 |
| heightType | 可选 | String(1-128) | 高度类型 | “default”，“low”；默认为“default” |
| displayItemOne | 必选 | DisplayItem | 通用展示子视图 | 排第一个的展示视图 |
| displayItemTwo | 可选 | DisplayItem | 通用展示子视图 | 可选，用于一排显示3个的情况，前俩个显示通用视图，第3个为反转开关 |
| command | 必选 | ReversalCommand[] | 控件对应的动作数组 |  |
| gravity | 可选 | String(1-128) | 子视图位置样式 | 样式分为：”default”,” center”，“centerDivider”。默认为“default”，位置样式为俩边靠，无分隔线。” center”为等分居中，” centerDivider”为等分居中且带分隔线。 |

ReversalCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| sid | 必选 | String (1-32) | 命令对应的服务 | 例如：“switch” |
| characteristic | 必选 | String (1-32) | 命令对应的属性 | 例如：“on” |
| reversal | 必选 | Reversal[] | 反转命令属性 | 数组，对应switch/on的不同属性值，控制的时候，可以相互切换 |

Reversal

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| value | 可选 | int | 命令下发对应的属性值 | 表示开关值 |
| icon | 必选 | String (1-128) | 命令对应的图像标签 | 命令对应的图标url |
| disableIcon | 可选 | String (1-128) | 不可用图标url | 不可用图标url |
| disable | 可选 | String[] | 命令触发后禁用的标签数组 | 例如：["模式", "亮度"]，例如：空气净化器关闭的时候，禁用睡眠模式。 |
| dialogList | 可选 | String[] | 命令触发后会弹出的对话框列表 | dialogList里面的值对应dialog的id值 |

Property

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| path | 必选 | String (1-32) | 属性路径 | 例如："switch/on"，其中switch表示服务，on表示属性 |
| value | 必选 | String (1-32) | 属性值 | 例如：1 |

#### Increase\_Decrease控件

Increase\_Decrease控件用于增加或减少数据，支持显示三个区域内容，左边区域内容是减少按钮，右边区域内容是增加按钮，中间区域内容用于显示操作的数据，点击增加减少按钮可以改变中间的数据。

示例图：



对应样例：

{   
 "name": "温度",   
 "span": 1,   
 "gravity": "default",   
 "uiType": "INCREASE\_DECREASE",   
 "heightType": "low",   
 "displayItem": {   
 "subType": "text",   
 "textDisplay": {   
 "down": {},   
 "upLeft": {   
 "type": "normalText",   
 "value": "温度"   
 },   
 "upRight": {}   
 }   
 },   
 "command": [   
 {   
 "sid": "temperature",   
 "characteristic": "temperature",   
 "max": 36,   
 "min": 17,   
 "ratio": 1,   
 "step": 1,   
 "defaultValue": 26,   
 "disableIconDecrease": "",   
 "iconDecrease": "",   
 "disableIconIncrease": "",   
 "iconIncrease": ""   
 }   
 ]   
}

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String (1-128) | 当前控件显示名称 | 例如：“亮度” |
| span | 必选 | float | 跨度 | 根据普通手机设置控件的宽度占比，为1表示占满普通手机的宽。 |
| uiType | 必选 | String(1-128) | 控件风格 | 控件风格为：“DISPLAY” ，“REVERSAL”,“INCREASE\_DECREASE”，“ENUM” ，“MULTIFUNCTION” |
| tag | 可选 | String(1-128) | 标记 | 扩展字段，用作标记特殊控件处理 |
| heightType | 可选 | String(1-128) | 高度类型 | “default”，“low”；默认为“default” |
| displayItem | 可选 | DisplayItem | 通用展示子视图 | 中间的展示视图 |
| command | 必选 | IncreaseDecreaseCommand[] | 控件对应的动作数组 |  |
| gravity | 可选 | String(1-128) | 子视图位置样式 | 样式分为：”default”,” center”，“centerDivider”。默认为“default”，位置样式为俩边靠，无分隔线。” center”为等分居中，” centerDivider”为等分居中且带分隔线。 |

IncreaseDecreaseCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| sid | 必选 | String (1-32) | 命令对应的服务 | 例如：“brightness” |
| characteristic | 必选 | String (1-32) | 命令对应的属性 | 例如：“brightness” |
| max | 必选 | int | 范围最大值 | 例如：亮度最大值为100 |
| min | 必选 | int | 范围最小值 | 例如：亮度最大值为0 |
| ratio | 可选 | float | 补偿因子 | 例如：空气净化器检测PM2.5为500微克，实际页面上显示单位为毫克，显示内容为0.5，那么补偿因子为0.001，再选中页面数值后，控制的过程中，要再次除以补偿因子，进行命令下发。 |
| step | 必选 | int | 步进值 | 每次变化的值 |
| iconDecrease | 必选 | String | 减少的图标 | 减少的图标 |
| disableIconDecrease | 必选 | String | 减少的不可用图标 | 减少的不可用图标 |
| iconIncrease | 必选 | String | 增加的图标 | 增加的图标 |
| disableIconIncrease | 必选 | String | 增加的不可用图标 | 增加的不可用图标 |
| defaultValue | 可选 | int | 默认值 |  |
| disable | 可选 | String[] | 命令触发后禁用的标签数组 | 例如：["模式", "亮度"]，例如：空气净化器关闭的时候，禁用睡眠模式。 |

#### Progress控件

Progress控件用于控制进度，支持显示两个区域内容，左边区域内容是对当前进度的描述，支持文字，或者文字加图片（图片显示大小为 24vp \*24vp），文字颜色支持自定义，颜色和图片支持根据数据范围来选择。右边的区域用于显示进度条。

示例图：



对应样例：

{   
 "gravity": "default",   
 "heightType": "low",   
 "name": "亮度",   
 "span": 1,   
 "command": [   
 {   
 "characteristic": "brightness",   
 "defaultValue": 100,   
 "max" : 255,   
 "min": 0,   
 "radio" : 1,   
 "sid": "brightness"   
 }   
 ],   
 "displayItemLeft": {   
 "subType": "text",   
 "textDisplay": {   
 "down": {   
 "color": {   
 "path" : "brightness/brightness",   
 "value" : [{   
 "scope": "-",   
 "target": "007DFF"   
 }]   
 },   
 "ratio": "0.392156",   
 "type": "dynamicText",   
 "value": "${brightness/brightness}%"   
 },   
 "upLeft": {   
 "type": "normalText",   
 "value": "亮度"   
 },   
 "upRight": {}   
 }   
 },   
 "uiType": "PROGRESS"   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String (1-128) | 当前控件显示名称 | 例如：“亮度” |
| span | 必选 | float | 跨度 | 根据普通手机设置控件的宽度占比，为1表示占满普通手机的宽。 |
| uiType | 必选 | String(1-128) | 控件风格 | 控件风格为：“DISPLAY” ，“REVERSAL”,“INCREASE\_DECREASE”，“ENUM” ，“MULTIFUNCTION” |
| heightType | 可选 | String(1-128) | 高度类型 | “default”，“low”；默认为“default” |
| displayItem | 可选 | DisplayItem | 通用展示子视图 | 中间的展示视图 |
| command | 必选 | ProgressCommand[] | 控件对应的动作数组 |  |

ProgressCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| sid | 必选 | String (1-32) | 命令对应的服务 | 例如：“brightness” |
| characteristic | 必选 | String (1-32) | 命令对应的属性 | 例如：“brightness” |
| max | 必选 | int | 范围最大值 | 例如：亮度最大值为100 |
| min | 必选 | int | 范围最小值 | 例如：亮度最大值为0 |
| ratio | 可选 | float | 补偿因子 | 例如：空气净化器检测PM2.5为500微克，实际页面上显示单位为毫克，显示内容为0.5，那么补偿因子为0.001，再选中页面数值后，控制的过程中，要再次除以补偿因子，进行命令下发。 |
| step | 必选 | int | 步进值 | 每次变化的值 |
| iconDecrease | 必选 | String | 减少的图标 | 减少的图标 |
| disableIconDecrease | 必选 | String | 减少的不可用图标 | 减少的不可用图标 |
| iconIncrease | 必选 | String | 增加的图标 | 增加的图标 |
| disableIconIncrease | 必选 | String | 增加的不可用图标 | 增加的不可用图标 |
| defaultValue | 可选 | int | 默认值 |  |
| disable | 可选 | String[] | 命令触发后禁用的标签数组 | 例如：["模式", "亮度"]，例如：空气净化器关闭的时候，禁用睡眠模式。 |

#### Multifunction控件

Multifunction控件用于媒体控制，支持显示两个或三个区域内容，中间的区域内容可不显示，左边的区域内容用于描述当前状态，中间的区域内容仅用于计时，右面的区域内容用于显示操作的按钮。示例图：



对应样例：

{   
 "name": "制作",   
 "span": 1,   
 "uiType": "MULTIFUNCTION",   
 "displayItemOne": {   
 "subType":"text",   
 "textDisplay": {   
 "down": {},   
 "upLeft": {   
 "description": {   
 "path": "workingStatus/status",   
 "value": [   
 {   
 "scope": "1",   
 "target": "待机中"   
 },   
 {   
 "scope": "2",   
 "target": "准备中"   
 },   
 {   
 "scope": "3",   
 "target": "制作中"   
 },   
 {   
 "scope": "4",   
 "target": "制作完成"   
 },   
 {   
 "scope": "5",   
 "target": "预约中"   
 }   
 ]   
 },   
 "type": "dynamicText",   
 "value": "${workingStatus/status}"   
 },   
 "upRight": {}   
 }   
 },   
 "displayItemTwo": {   
 "subType":"text",   
 "textDisplay":{   
 "down":{   
 "type":"normaltext",   
 "value":"剩余时间"   
 },   
 "upLeft":{   
 "type":"dynamicText",   
 "value":"${remainingTime/time}"   
 },   
 "upRight":{   
   
   
 }   
 }   
 },   
 "multifunctionInfo": {   
 "path": "workingStatus/action",   
 "value": [   
 {   
 "scope": "1",   
 "commandOne": [   
 {   
 "characteristic": "action",   
 "disableIcon": "/ic\_prepare.png",   
 "icon": "/ic\_prepare.png",   
 "disableIconNight": "/ic\_prepare.png",   
 "iconNight": "/ic\_prepare.png",   
 "value": 3,   
 "sid": "workingStatus"   
 }   
 ]   
 },   
 {   
 "scope": "2",   
 "commandOne": [   
 {   
 "characteristic": "action",   
 "disableIcon": "/ic\_prepare.png",   
 "icon": "/ic\_prepare.png",   
 "disableIconNight": "/ic\_prepare.png",   
 "iconNight": "/ic\_prepare.png",   
 "value": 3,   
 "sid": "workingStatus"   
 }   
 ],   
 "commandTwo": [   
 {   
 "characteristic": "action",   
 "disableIcon": "/ic\_cancel.png",   
 "icon": "/ic\_cancel.png",   
 "disableIconNight": "/ic\_cancel.png",   
 "iconNight": "/ic\_cancel.png",   
 "value": 0,   
 "sid": "workingStatus"   
 }   
 ]   
 },   
 {   
 "scope": "3",   
 "commandOne": [   
 {   
 "characteristic": "action",   
 "disableIcon": "/ic\_cancel.png",   
 "icon": "/ic\_cancel.png",   
 "disableIconNight": "/ic\_cancel.png",   
 "iconNight": "/ic\_cancel.png",   
 "value": 0,   
 "sid": "workingStatus"   
 }   
 ]   
 },   
 {   
 "scope": "4",   
 "commandOne": [   
 {   
 "characteristic": "action",   
 "disableIcon": "/ic\_complete.png",   
 "icon": "/ic\_complete.png",   
 "disableIconNight": "/ic\_complete.png",   
 "iconNight": "/ic\_complete.png",   
 "value": 0,   
 "sid": "workingStatus"   
 }   
 ]   
 },   
 {   
 "scope": "5",   
 "commandOne": [   
 {   
 "characteristic": "action",   
 "disableIcon": "/ic\_cancel.png",   
 "icon": "/ic\_cancel.png",   
 "disableIconNight": "/ic\_cancel.png",   
 "iconNight": "/ic\_cancel.png",   
 "value": 0,   
 "sid": "workingStatus"   
 }   
 ]   
 }   
 ]   
 }   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String (1-128) | 当前控件显示名称 | 例如：“Battery” |
| span | 必选 | float | 跨度 | 根据普通手机设置控件的宽度占比，为1表示占满普通手机的宽。 |
| uiType | 必选 | String(1-128) | 控件风格 | 控件风格为：“DISPLAY” ，“REVERSAL”,“INCREASE\_DECREASE”，“ENUM” ，“MULTIFUNCTION” |
| heightType | 可选 | String(1-128) | 高度类型 | “default”，“low”；默认为“default” |
| displayItemOne | 必选 | DisplayItem | 通用展示子视图 | 排第一个的展示视图 |
| displayItemTwo | 可选 | DisplayItem | 通用展示子视图 | 可选，用于一排显示3个的情况，前俩个显示通用视图，第3个为多功能按钮 |
| multifunctionInfo | 必选 | DynamicCommandContent | 多功能按钮 |  |

DynamicCommandContent

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| path | 必选 | String (1-128) | 动态内容条件值，可能需要动态获取设备数据填充 | 例如：“pm2p5/pm2p5Value” ，表示当前设备上报PM2.5的值 |
| value | 必选 | VariationCommand | 动态范围值 | { "scope": "-50", "target": "11FCFC" } |

VariationCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| scope | 必选 | String (1-128) | 条件范围 | 用“-”区分范围，有三种类型： 1，-50：表示小于等于50的值 2，50-100；表示大于50小于等于100； 3，100-：表示大于100的值 4，"scope": "-"时，表示可以匹配任意场景。 5，50：表示等于50 |
| commandOne | 必选 | MultifunctionCommand | 执行动作1 | { "characteristic": "action", "disableIcon": "icon\_start\_dis.png", "icon": "icon\_start.png", "value": 1, "sid": "action" } |
| commandTwo | 可选 | MultifunctionCommand | 执行动作2 | { "characteristic": "action", "disableIcon": "icon\_start\_dis.png", "icon": "icon\_start.png", "value": 1, "sid": "action" } |

MultifunctionCommand

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| sid | 必选 | String (1-32) | 命令对应的服务 | 例如：“switch” |
| characteristic | 必选 | String (1-32) | 命令对应的属性 | 例如：“on” |
| value | 可选 | int | 命令下发对应的属性值 | value和valueString二者存在其一。其中wifi设备使用value，蓝牙设备使用value或者valueString。 |
| valueString | 可选 | String (1-128) | 命令下发对应的属性值 | value和valueString二者存在其一。其中wifi设备使用value，蓝牙设备使用value或者valueString。 |
| icon | 必选 | String (1-128) | 命令对应的图像标签 | 命令对应的图标url |
| disableIcon | 可选 | String (1-128) | 不可用图标url | 不可用图标url |
| disable | 可选 | String[] | 命令触发后禁用的标签数组 | 例如：["模式", "亮度"]，例如：空气净化器关闭的时候，禁用睡眠模式。 |

### Dialog类型控件配置说明

Dialog类型控件一共有七个，分贝是枚举类型Dialog、单选类型Dialog、展示类型Dialog、时间选择器类型Dialog、日期选择器类型Dialog、日期时间选择器类型Dialog、多列文本选择器类型Dialog。

#### 枚举类型Dialog

枚举类型Dialog用于在Dialog中类型的选择，可以显示一到八个内容区域，每个区域可以点击，点击后图片发生改变，并只有一个区域内容处于被点击的状态。

示例图：



对应样例：

{   
 "name": "饮功能选择",   
 "id": "drinkKind",   
 "path": "drinkKind",   
 "dialogType": "ENUM",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "下一步",   
 "type": "submit"   
 }   
 ],   
 "command": [   
 {   
 "name": "五谷",   
 "icon": {   
 "value": {   
 "defaultTarget": "/ic\_grains\_nor.png",   
 "target": "/ic\_grains\_on.png"   
 },   
 "valueNight": {   
 "defaultTarget": "/ic\_grains\_nor.png",   
 "target": "/ic\_grains\_on.png"   
 }   
 },   
 "dialogList": ["duration", "temperature", "capacity", "speed"],   
 "value": 1   
 },   
 {   
 "name": "豆浆",   
 "icon": {   
 "value": {   
 "defaultTarget": "/ic\_soy\_milk\_nor.png",   
 "target": "/ic\_soy\_milk\_on.png"   
 },   
 "valueNight": {   
 "defaultTarget": "/ic\_soy\_milk\_nor.png",   
 "target": "/ic\_soy\_milk\_on.png"   
 }   
 },   
 "dialogList": ["duration", "temperature", "capacity", "speed"],   
 "value": 2   
 }   
 ]   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“饮功能选择” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“drinkKind” |
| path | 必选 | String（1-128） | 动态内容条件值 | 例如：“drinkKind” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| command | 必选 | Command[] | 弹框对应的动作数组 |  |

ControlItems

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 控制项显示名称 | 例如：“取消” |
| type | 必选 | String（1-128） | 控制项类型 | 控制项类型为：“submit”，“cancel” |

Command

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 显示名称 | 例如：“五谷” |
| icon.value.defaultTarget | 必选 | String（1-128） | 默认图标url | 例如：“/ic\_doujiang\_nor.png” |
| icon.value.target | 必选 | String（1-128） | 点击状态图标url | 例如：“/ic\_doujiang\_on.png” |
| icon.valueNight.defaultTarget | 必选 | String（1-128） | 深色模式默认图标url | 例如：“/ic\_doujiang\_nor.png” |
| icon.valueNight.target | 必选 | String（1-128） | 深色模式点击状态图标url | 例如：“/ic\_doujiang\_on.png” |
| dialogList | 可选 | Array | 继续显示的弹框列表 |  |
| value | 必选 | int | 命令下发对应的属性值 | 例如：1 |

#### 单选类型Dialog

单选类型Dialog用于在Dialog中进行唯一选择。

示例图：



对应样例：

{   
"name": "时间设置",   
 "id": "drinkTime1",   
 "path": "drinkTime1",   
 "dialogType": "RADIO",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "下一步",   
 "type": "submit"   
 }   
 ],   
 "range": ["1分钟", "3分钟", "5分钟", "10分钟"],   
 "value": [1, 3, 5, 10]   
 }

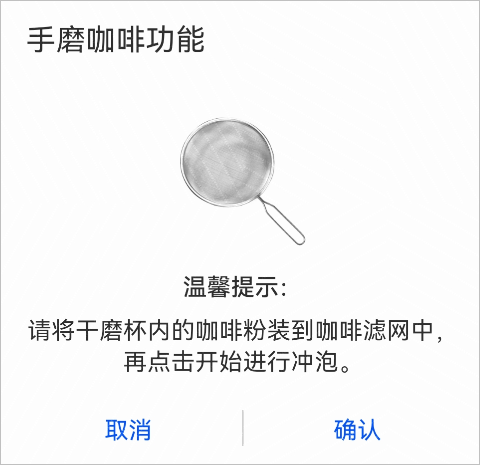
相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“时长设置” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“duration” |
| path | 必选 | String（1-128） | 动态内容条件值 | 例如：“duration” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| range | 必选 | Array | 单选项的数组 | 例如：["1分钟", "3分钟", "5分钟", "10分钟"] |
| value | 必选 | Array | 单选项对应值的数组 | 例如：[1, 3, 5, 10] |

#### 展示类型Dialog

展示类型Dialog用于显示需要展示的图片和描述文字。

示例图：



对应样例：

{   
 "name": "手磨咖啡功能",   
 "id": "handGrindingCoffee",   
 "dialogType": "INFO",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "确认",   
 "type": "submit"   
 }   
 ],   
 "icon": "/ic\_grand\_coffee.png",   
 "headline": "温馨提示",   
 "desc": "请将干磨杯内的咖啡粉装到咖啡滤网中，再点击开始进行冲泡。"   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“手磨咖啡功能” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“handGrindingCoffee” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| icon | 必选 | String（1-128） | 展示图片url | 例如：“/ic\_shoumokafei.png” |
| headline | 必选 | String（1-128） | 标题 | 例如：“温馨提示” |
| desc | 必选 | String（1-128） | 描述 | 例如：“请将干磨杯内的咖啡粉装到咖啡滤网中，再点击开始进行冲泡。” |

#### 时间选择器类型Dialog

时间选择器类型Dialog用于在Dialog中选择时间，可以按12小时制显示时间也可以按24小时显示时间。

示例图：



对应样例：

{   
 "name": "时间设置",   
 "id": "time",   
 "path": "time",   
 "dialogType": "PICKER",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "确认",   
 "type": "submit"   
 }   
 ],   
 "pickerType": "time",   
 "containSecond": true,   
 "hours": 24   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“时间设置” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“time” |
| path | 必选 | String（1-128） | 动态内容条件值 | 例如：“time” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| pickerType | 必选 | String（1-128） | 选择器类型 | 选择器类型为：“text”，“time”，“date”，“datetime”，“multi-text” |
| containSecond | 可选 | boolean | 时间选择器是否包含秒 | 例如：true |
| hours | 可选 | int | 时间选择器采用的时间格式 | 可选值：12：按照12小时制显示，用上午和下午进行区分；24：按照24小时制显示。 |

#### 日期选择器类型Dialog

日期选择器类型Dialog用于在Dialog中选择日期，可以显示农历时间也可以显示公历时间。

示例图：



对应样例：

{   
"name": "设置日期",   
 "id": "date",   
 "path": "date",   
 "dialogType": "PICKER",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "确认",   
 "type": "submit"   
 }   
 ],   
 "pickerType": "date",   
 "start": "1970-1-1",   
 "end": "2100-12-31",   
 "lunar": false   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“设置日期” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“date” |
| path | 必选 | String（1-128） | 动态内容条件值 | 例如：“date” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| pickerType | 必选 | String（1-128） | 选择器类型 | 选择器类型为：“text”，“time”，“date”，“datetime”，“multi-text” |
| start | 可选 | time | 日期选择器的起始时间 | 格式为 YYYY-MM-DD |
| end | 可选 | time | 日期选择器的结束时间 | 格式为 YYYY-MM-DD |
| lunar | 可选 | boolean | 日期选择器是否为农历展示 | 例如：false |

#### 日期时间选择器类型Dialog

日期时间选择器类型Dialog用于在Dialog中选择日期和时间，可以按12小时制显示时间也可以按24小时制显示时间。

示例图：



对应样例：

{   
 "name": "日期时间设置",   
 "id": "datetime",   
 "path": "datetime",   
 "dialogType": "PICKER",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "确认",   
 "type": "submit"   
 }   
 ],   
 "pickerType": "datetime",   
 "lunar": false,   
 "hours": 24   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“日期时间设置” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“datetime” |
| path | 必选 | String（1-128） | 动态内容条件值 | 例如：“datetime” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| pickerType | 必选 | String（1-128） | 选择器类型 | 选择器类型为：“text”，“time”，“date”，“datetime”，“multi-text” |
| lunar | 可选 | boolean | 日期时间选择器是否为农历显示 | 例如：false |
| hours | 可选 | int | 时间选择器采用的时间格式 | 可选值：12：按照12小时制显示，用上午和下午进行区分；24：按照24小时制显示。 |

#### 多列文本选择器类型Dialog

多列文本选择器类型Dialog用于在Dialog中选择多列文本，文本内容通过配置指定。

示例图：



对应样例：

{   
 "name": "时长设置",   
 "id": "duration",   
 "path": "duration",   
 "dialogType": "PICKER",   
 "controlItems": [   
 {   
 "name": "取消",   
 "type": "cancel"   
 },   
 {   
 "name": "下一步",   
 "type": "submit"   
 }   
 ],   
 "pickerType": "multi-text",   
 "range": [   
 ["0分","1分","2分","3分","4分","5分","6分"],   
 ["0秒","1秒","2秒","3秒","4秒","5秒","6秒"]   
 ]   
 }

相关的参数，定义如下：

| 字段名称 | 可选/必选 | 字段类型 | 字段描述 | 备注 |
| --- | --- | --- | --- | --- |
| name | 必选 | String（1-128） | 当前弹框显示名称 | 例如：“时长设置” |
| id | 必选 | String（1-128） | 弹框唯一标识 | 例如：“duration” |
| path | 必选 | String（1-128） | 动态内容条件值 | 例如：“duration” |
| dialogType | 必选 | String（1-128） | 弹框类型 | 弹框类型为：“RADIO”，“ENUM”，“PICKER”，“INFO” |
| controlItems | 必选 | ControlItems[] | 弹框底部控制项 |  |
| pickerType | 必选 | String（1-128） | 选择器类型 | 选择器类型为：“text”，“time”，“date”，“datetime”，“multi-text“ |
| range | 必选 | 二维Array | 多列文本选择器的选择值 | 长度表示多少列，数组的每项表示每列的数据，如 [["a","b"], ["c","d"]] |