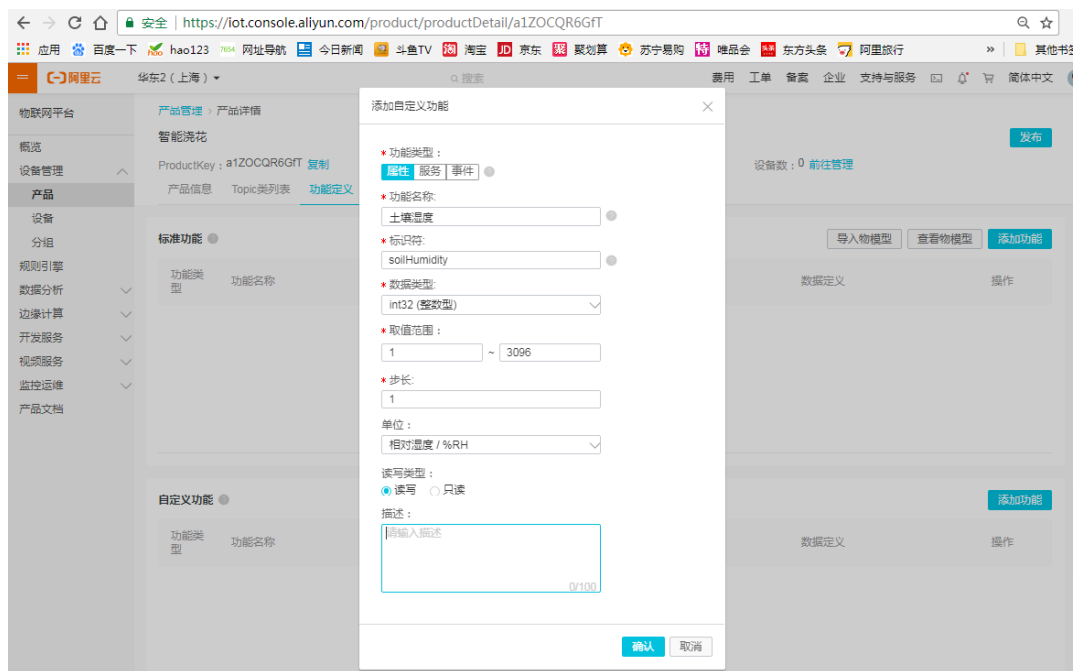
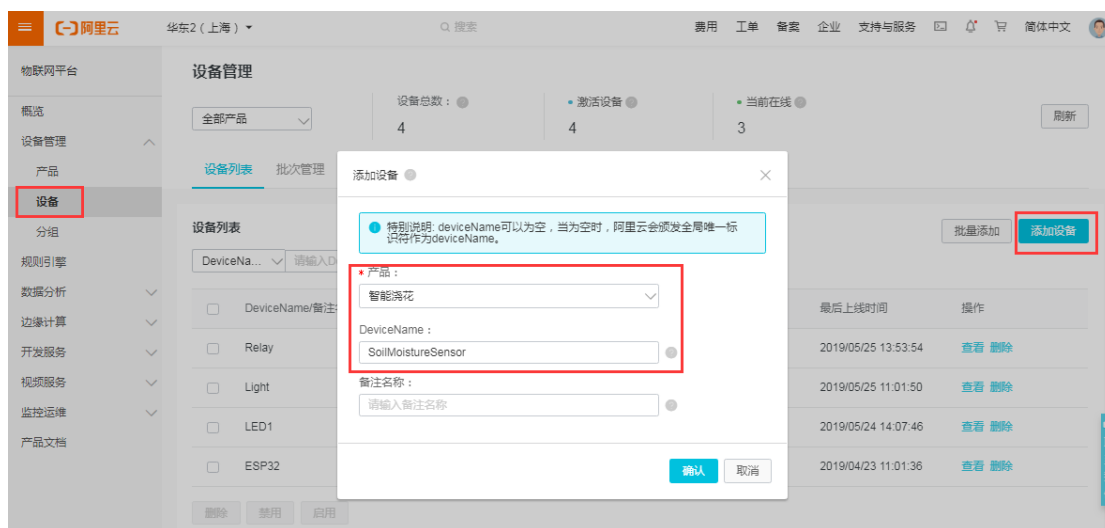




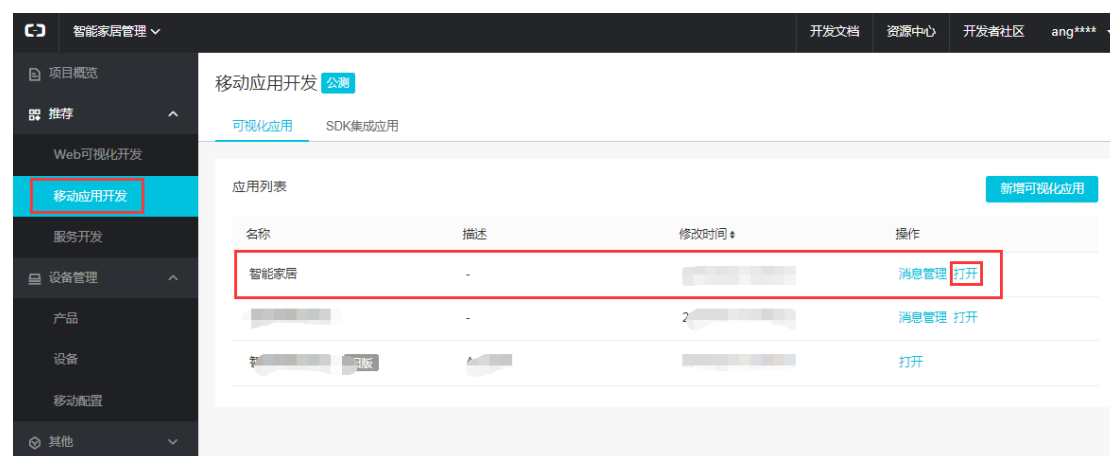
4.为**土壤湿度**添加一个名为 **SoilMoistureSensor** 的设备，如下图所示：



5.将**智能浇花**产品导入**智能家居管理**项目中，如下图所示：

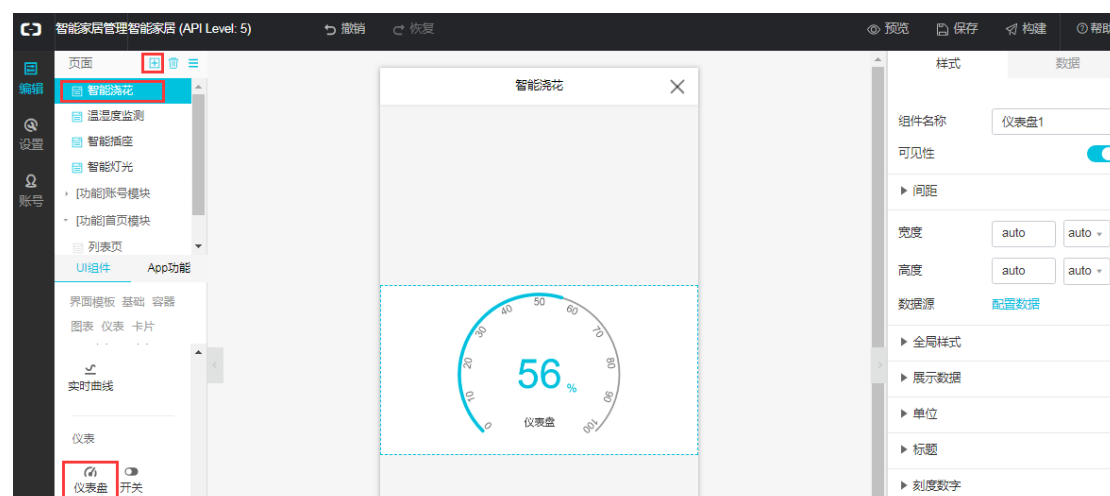


6. 选中左侧导航栏的**移动应用开发**,在应用列表找到**智能家居**,单击**打开**,如下图所示:



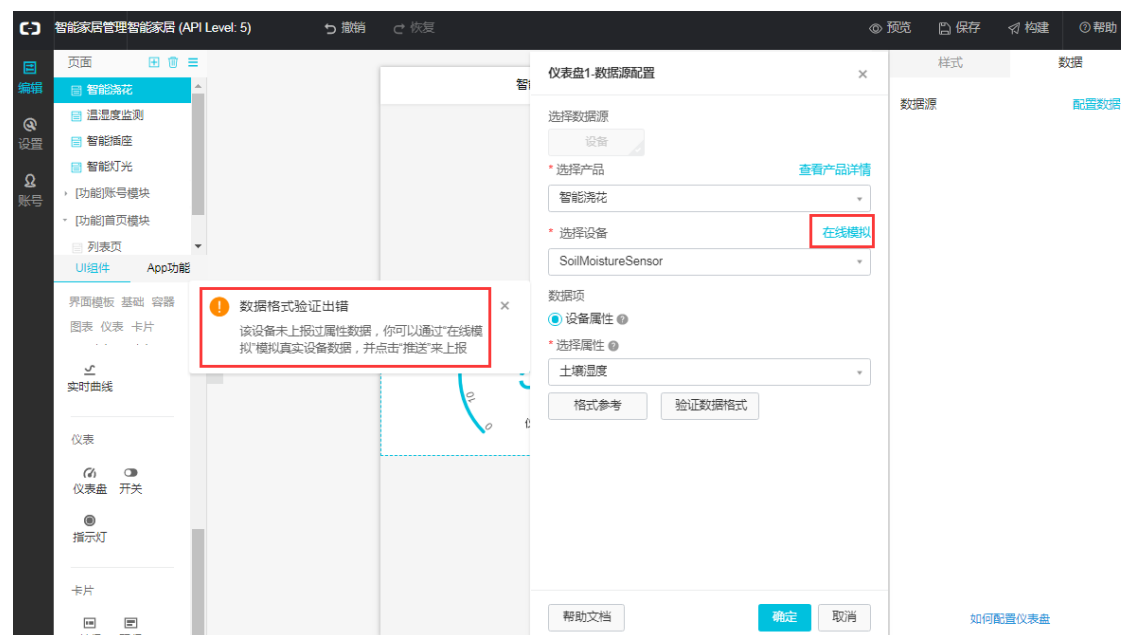
7. 智能插座 App 设计与开发。

① 点击**自定义页面**后的+号, 编辑, 将“自定义新增页 1”改为**智能浇花**, 用鼠标将可视化组建下的**仪表盘**控件拖动到中间的页面上, 如下图所示:



② 选中中间页面的仪表盘, 点击**数据->配置数据**, 在弹出的选项框里选择相应的选项, 并点击确定, 若出现下图所示“数据格式验证出错”, 这是因为该设备被创建以来, 还没有

被使用过，可以通过[在线模拟](#)解决该问题。



③点击“在线模拟”，进入如下所示页面，（若真实设备在线，则可选择“调试真实设备”）这里我们选择启动虚拟设备。



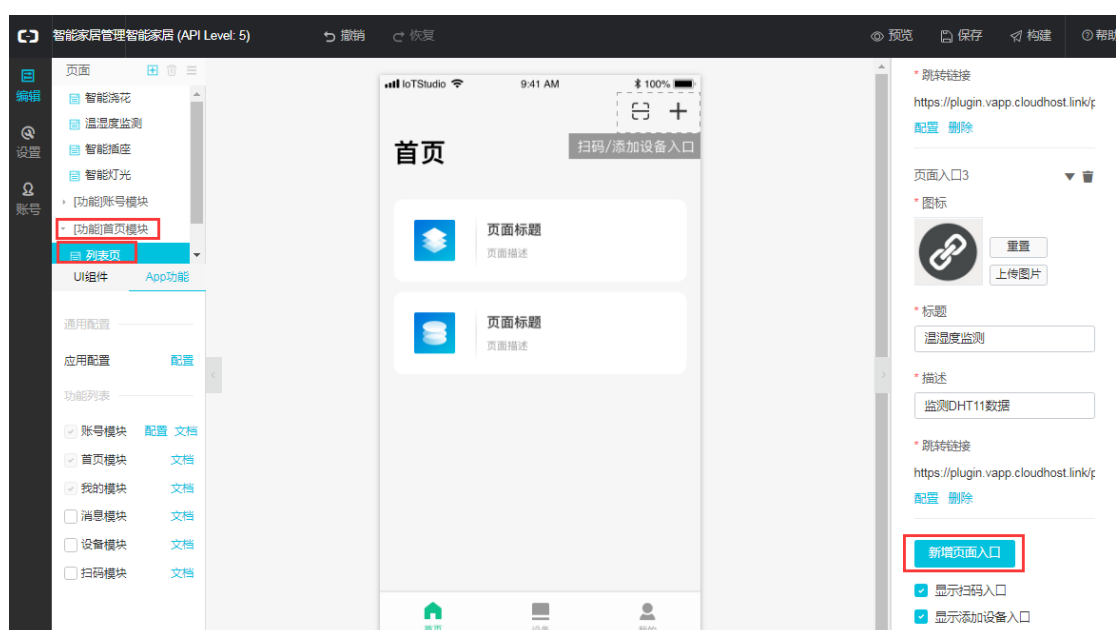
④启动虚拟设备后，会进入如下所示页面，任意选择一种状态，点击[推送](#)即可，稍后会收到设备上报信息。



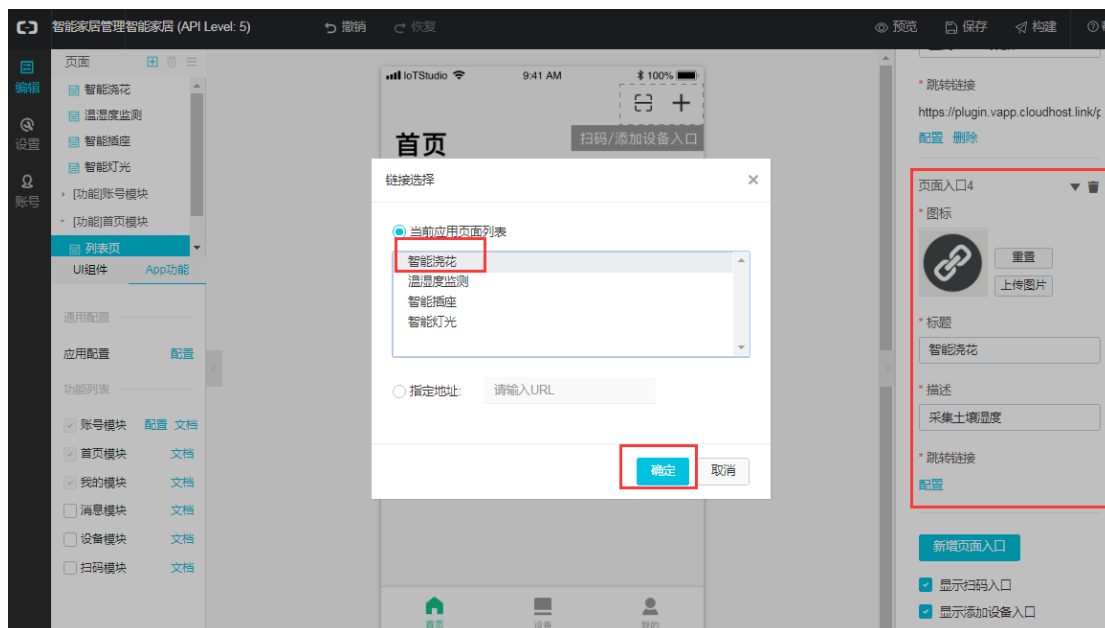
⑤收到设备上报数据后，在回到移动应用开发页面，点击**确定**，即可完成“仪表盘”的配置了，如果你不知如何配置指示灯可以点击右下角的“如何配置仪表盘”，这样就将仪表盘链接到“DHT11”设备上了，当“DHT11”设备采集到温湿度信息，“仪表盘”控件的状态也会发生改变，如下图所示：



⑥配置**首页模块**。这一步是为了设置一个导航栏，链接到上面设计的“智能灯光”页面，。选中**[功能]首页模块->列表**，选中**新增页面入口**，如下图所示：

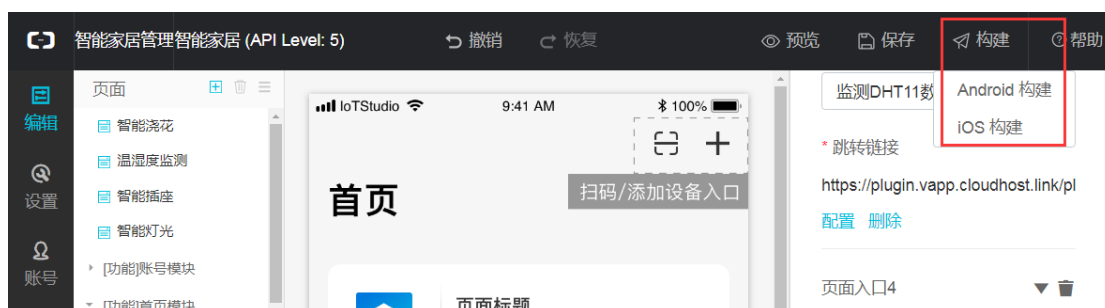


在弹出的“**页面入口 4**”下，完成相应信息的选择，将其链接到刚才设计的“智能灯光”页面，如下图所示，点击**确定**完成配置后，别忘了点击**保存**。

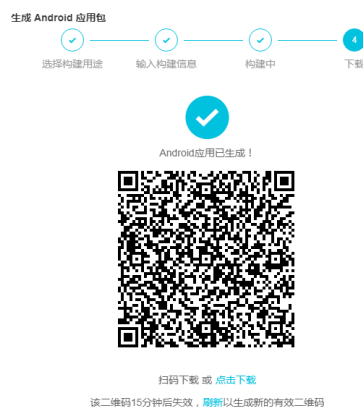


⑦账号配置。由于在应用一中已经添加了账号，这里就无需再添加了，直接构架安装包即可。

⑧安装包生成。在**编辑**页面，点击**构建**，选择 **Android 构建**或 **iOS 发构建**，如下图所示：



点击**我只想自己使用->随机生成->下一步**，等待打包完成，最终终会在页面上显示一个二维码。

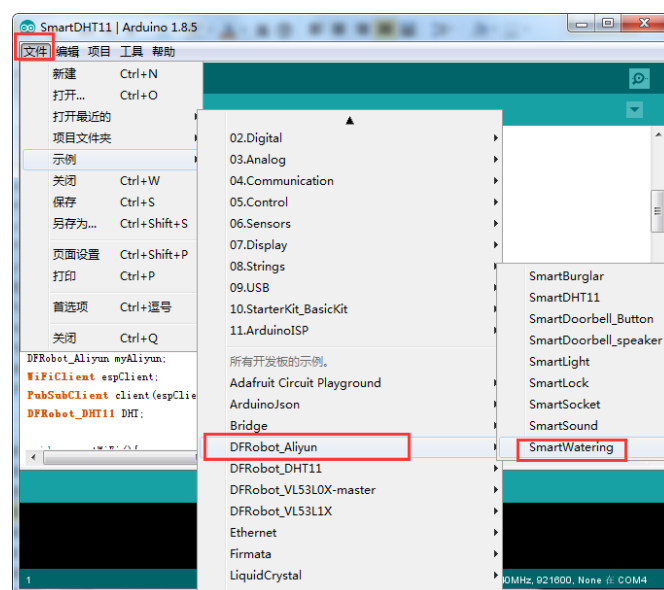


⑩手机 App 下载。手机微信或其他二维码扫描器扫描下载并安装该 App，用刚才创建的账号登录，即可打开 App 即可控制 LED 设备的亮灭了，如下图所示：

打开名为智能家居控制台的 App，如果已经登录，则会进入 App 首页，如下图所示：



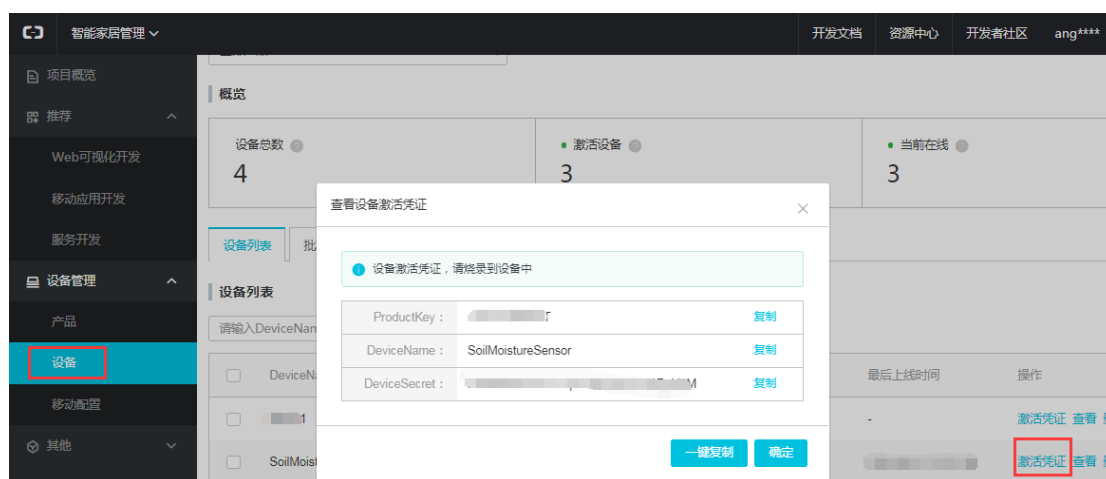
8.Arduino 程序修改。点击文件->示例->DFRobot_Aliyun->SmartWatering 项目，如下图所示：



① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示) 更换为可用的 WiFi。

```
/*配置WIFI名和密码*/  
const char * WIFI_SSID      = "WIFI_SSID";  
const char * WIFI_PASSWORD = "WIFI_PASSWORD";
```

② 填写设备证书信息。获取设备 **SoilMoistureSensor** 的设备信息，如下图所示：



将获取到的 **SoilMoistureSensor** 的设备证书信息，复制粘贴到如下图所示的程序模块中。

```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

③ 替换订阅、上报 Topic。在阿里云 IoT 上找到此应用创建的设备 **SoilMoistureSensor** 的 **Topic** 列表，如下图所示，并将它替换到程序的 **TOPIC** 设置中,由于只需要向上面上传信息，而无需下发，所以只需配置一个上报变量即可：



```
/*需要上报的IOTIC*/
const char * pubTopic = "you pub Topic";//*****post
```

④ 替换产品标识符。在设备管理的产品项找到名为**智能浇花**的产品，点击**查看**，点击**功能定义**，即可在**自定义功能栏**看到**标识符**，如下图所示，将其替换到程序中的产品标识符即可。

自定义功能 ●

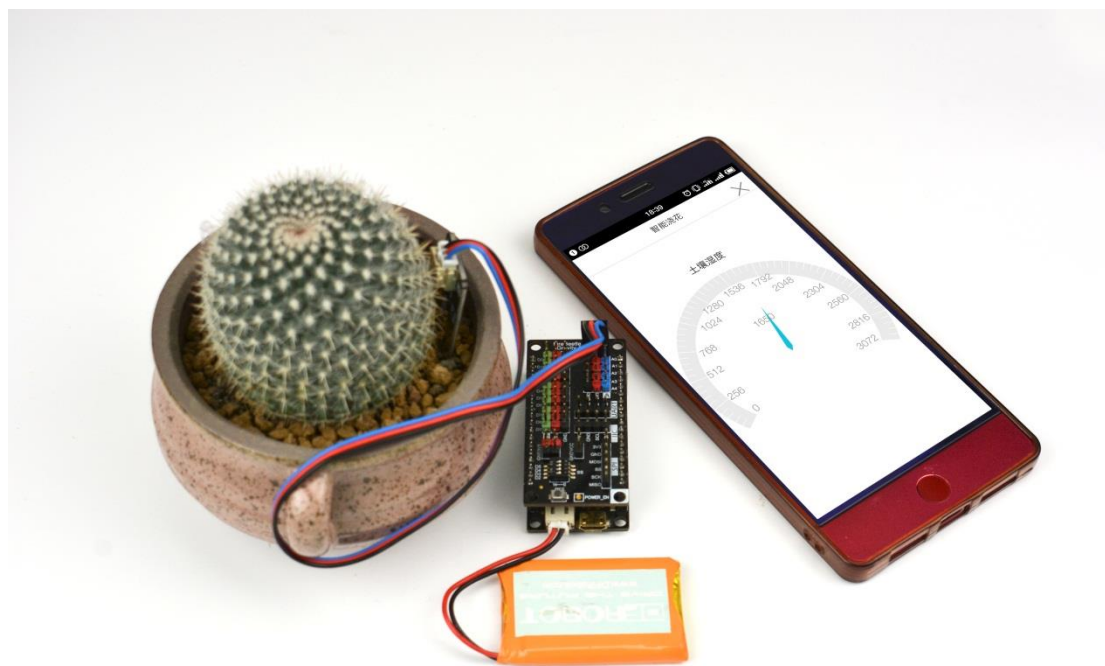
添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	土壤湿度	soilHumidity	int32 (整数型)	取值范围：1 ~ 3096	编辑 删除

```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

⑤编译上传至 FireBettle Board-ESP32 主板。选好开发板类型和端口号后，点击编译上传即可将程序上传至 ESP32 主板。

此时设备即成功与 Aliyun 平台上的 SoilMoistureSensor 设备关联，打开下载的智能家居控制台，选择智能浇花，即可进入控制页面，一段时间后，即可测得植物的土壤湿度，如下图所示：



3.5 应用五：智能防盗

前几个应用，都是通过在阿里云 IoT 平台上开发一个移动应用服务，来控制物理设备，这一节，将介绍如何通过阿里云的服务开发来关联控制设备。

应用目标：红外热释电将检测状态（有人：1，无人：0）上传至阿里云 IoT 上的某个服务，若服务的条件节点判断传过来的数据为有人（1），则进入钉钉机器人节点，将消息“家里有人闯入”推送至钉钉，同时功放喇叭会唱歌提示。

所需元件

PIR 红外热释电传感器 x1

功放喇叭 x1

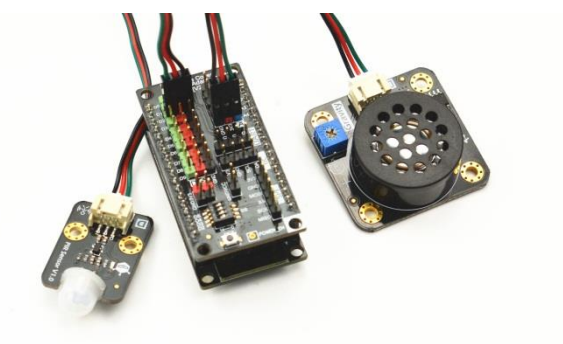
FireBettle Board-ESP32 x1

FireBettle Gravity 扩展板 x1

智能手机 x1

硬件连接

把 FireBettle Gravity 扩展板直插在 FireBettle Board-ESP32 主板上，再把 PIR 红外热释电传感器接在扩展板的 D2 数字引脚上，功放喇叭接在扩展板的 A4 引脚，确保电源引脚、接地引脚和信号引脚都连接正确。



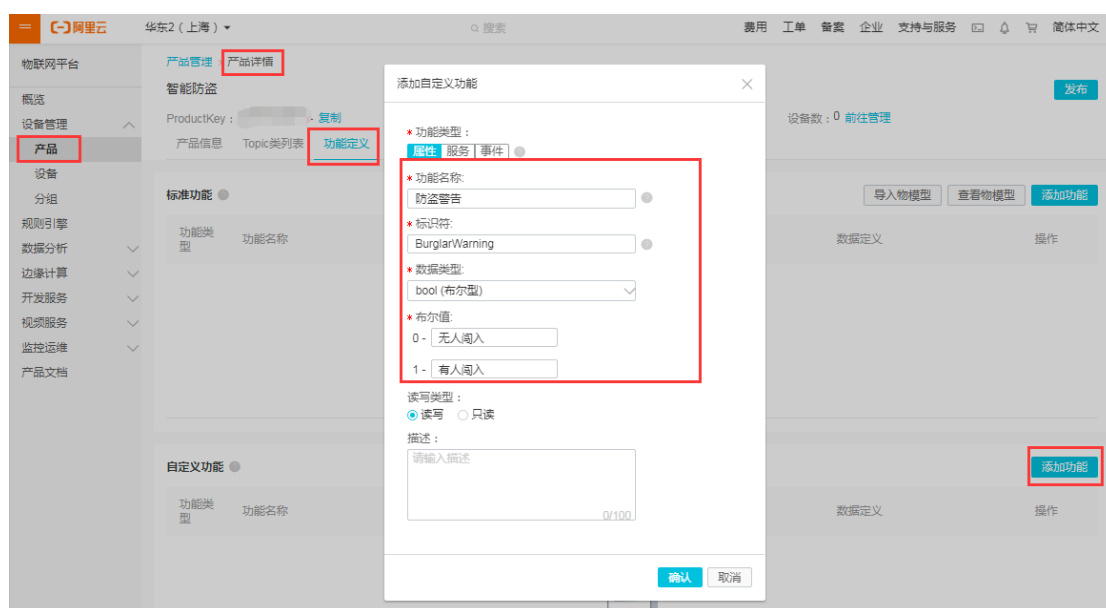
应用分析：智能防盗应用中，PIR 检测到有人，则启动喇叭报警提示，同时将“家里有人闯入”的消息推送至钉钉，提醒用户。这个项目需要我们做以下事情：1.在阿里云 IoT 上创建一个名为“**智能防盗**”的产品，并为其定义一个功能为“**防盗警告**”的属性；2.为“**智能防盗**”产品添加一个名为“**PIRMotion**”的设备；3、创建并开发一个名为“**智能防盗**”的服务开发；4 获取设备“**PIRMotion**”的设备证书，发布订阅 Topic 列表，以及获取产品的标识符，并在 Arduino 程序中填写修改。

制作步骤

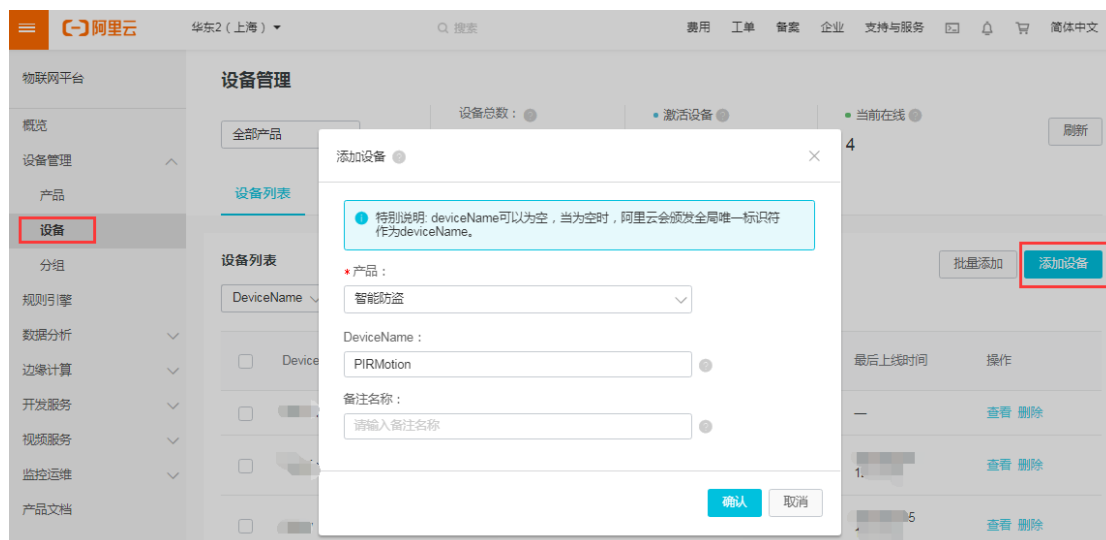
- 1.打开浏览器，输入网址 <https://iot.aliyun.com>,登录并进入**物联网平台**；
- 2.创建一个名为**智能防盗**的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



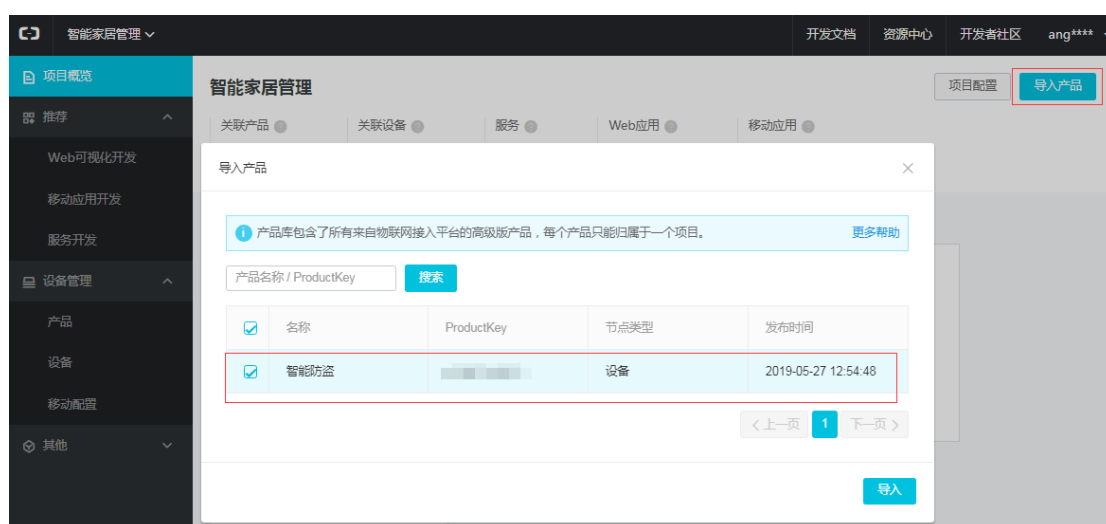
3. 为智能防盗产品定义一个功能名为“**防盗警告**”的属性，自定义一个产品标识符，防盗警告有 2 种状态：有人闯入（1），无人闯入（0）。如下图所示：



4. 为**智能防盗**添加一个名为 **PIRMotion** 的设备，如下图所示：



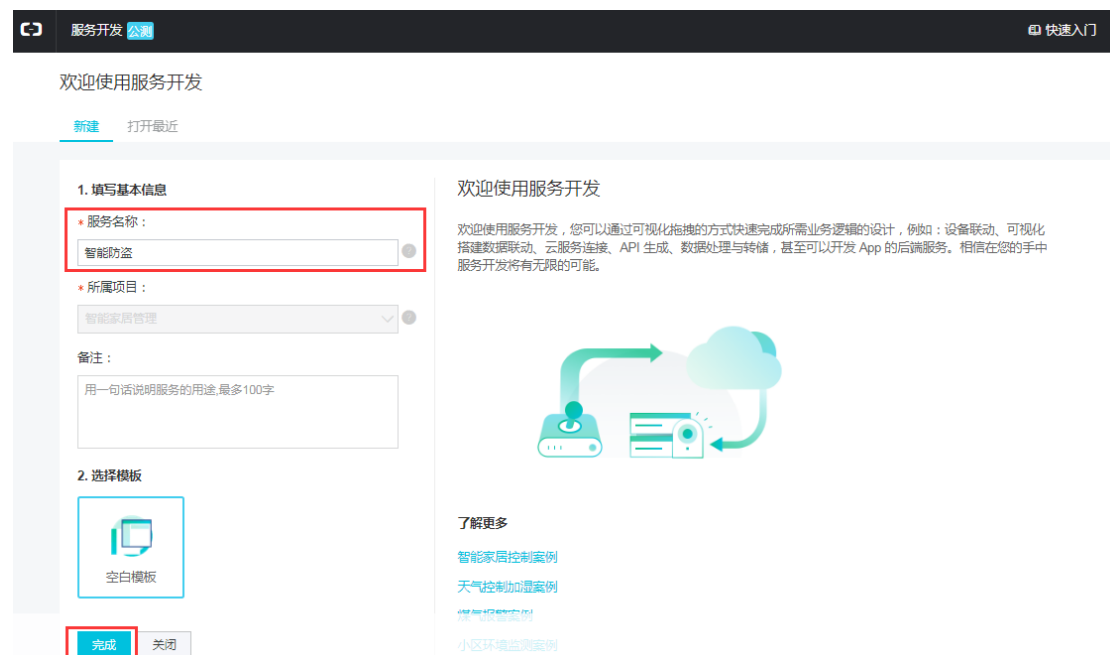
5.将**智能防盗**产品导入**智能家居管理**项目中，如下图所示：



6. 选中左侧导航栏的**推荐->服务开发**,点击**新建服务**，如下图所示：

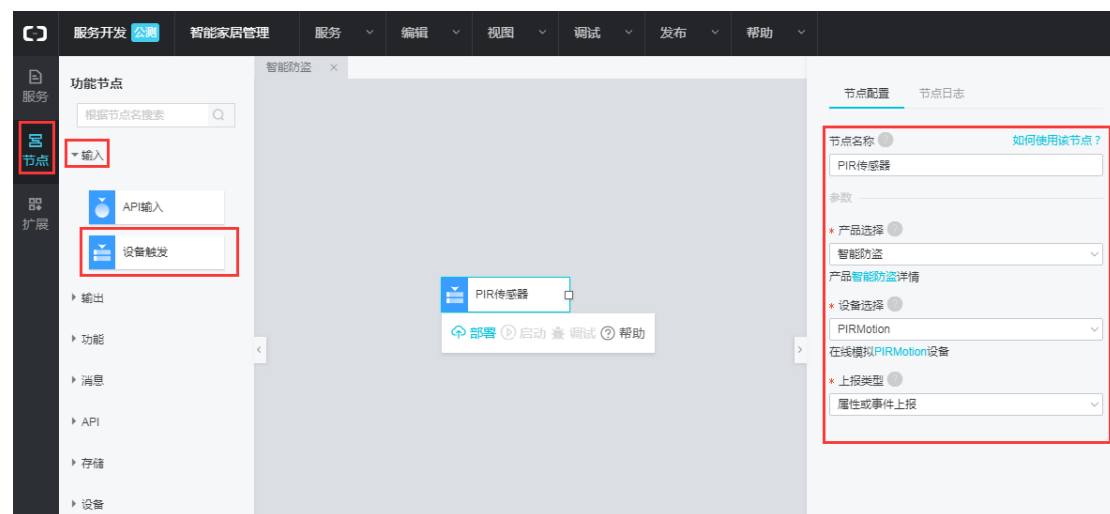


根据页面提示，创建一个名为“**智能防盗**”的服务，该服务内容大致为：当 PIR 传感器检测到有人闯入则启动喇叭报警，并将信息推送至钉钉，如下图所示：

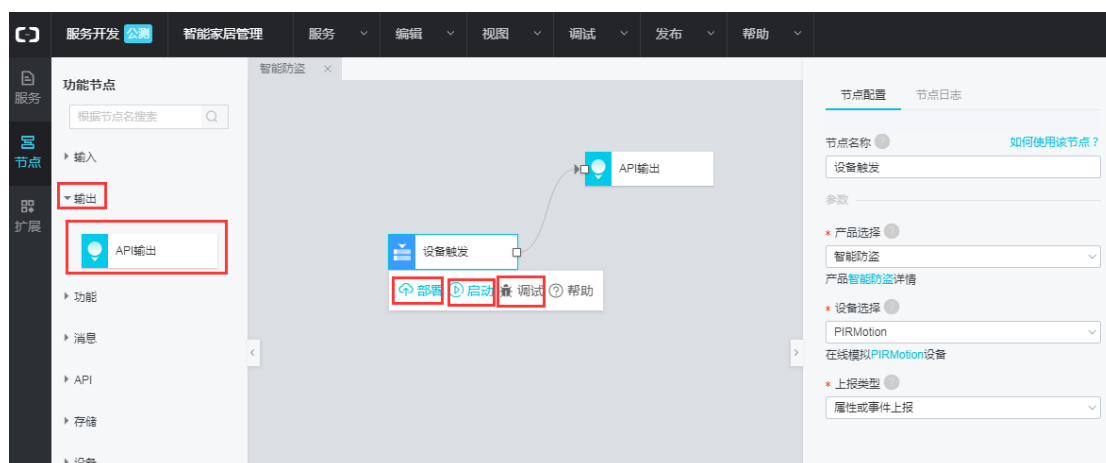


7. 点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址：
<https://linkdevelop.aliyun.com/studioservice-doc#index.html>

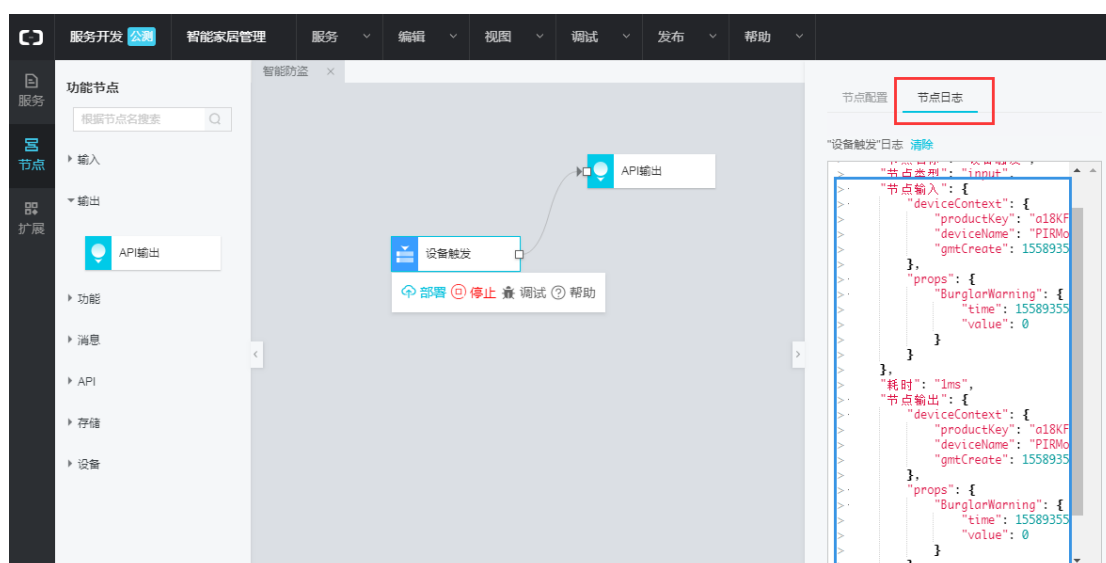
❶ 点击左侧导航栏的**节点**，在节点列表中选择**输入->设备触发**，并将其拖动到服务画布内，把这个设备触发节点改为**PIR 传感器**；右侧栏配置项分别选择“**智能家居安防**”产品，创建的设备名以及选择监听“**属性或事件上报**”，如下图所示：



❷ 获取“PIR 传感器”节点的节点输出信息。“PIR 传感器”节点配置完成后，选中左侧节点列表的**输出->API 输出->部署->启动->调试**，如下图所示：



在弹出的“在线调试”页面中选择某个参数，点击推送，回到服务开发页面，点击节点日志，即可看到“PIR 传感器”节点的输入输出节点信息，如图：



③添加一个“条件”节点，条件节点相当于一个 if-else 判断。选中左侧节点列表的功能->条件节点，将其改为“有人闯入”，并根据上面得到的“PIR 传感器”输出节点信息配置“有

人闯入”节点的输入设置，将其配置为“上一个节点”+“props.BurglarWarning.value”；将路径 1 的第二个选择框设置为“数字”，第三个框输入 1（表示有人闯入），可以参考[如何使用该节点？->如何配置条件节点？](#)



④删除“API 输出”节点，选中消息->钉钉机器人，添加钉钉机器人节点。并将 PIR 传感器、有人闯入、钉钉机器人节点连接起来。



⑤配置钉钉机器人节点。若不知道如何配置钉钉机器人，可参考[如何使用该节点？->如何配置钉钉机器人节点？](#)如下图所示：



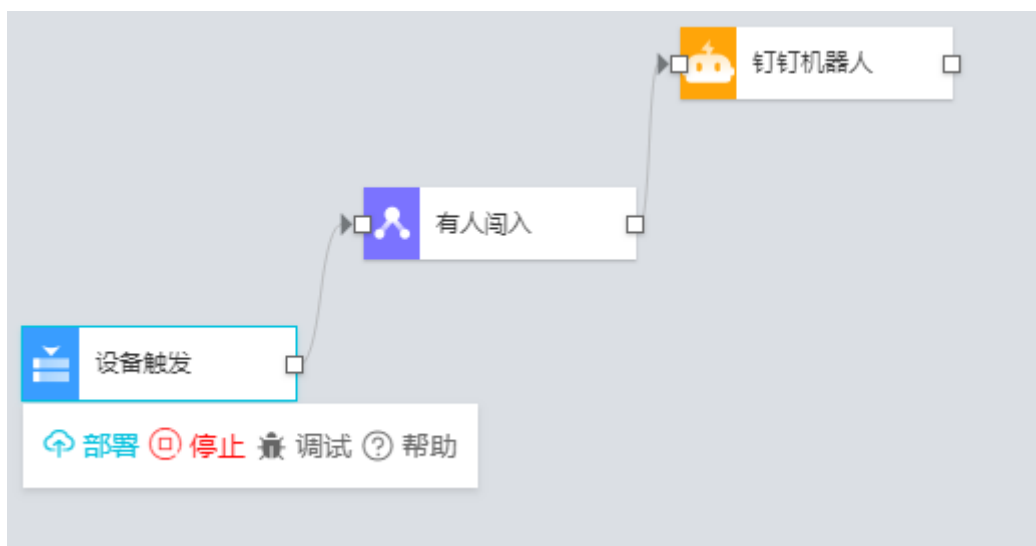
a. 获取 Webhook。在获取之前，你需要在钉钉群中，添加一个钉钉机器人(注意查看钉钉机器人的 Webhook 时只能用电脑端的钉钉)，具体添加方法请参考[如何使用该节点？->如何配置钉钉机器人节点？](#)最终获取的 Webhook，复制粘贴至右侧栏的 Webhook 处，如下图所示：



b. 选择“text”作为消息类型，并在 content 的双引号内填入信息“家里有人闯入”。

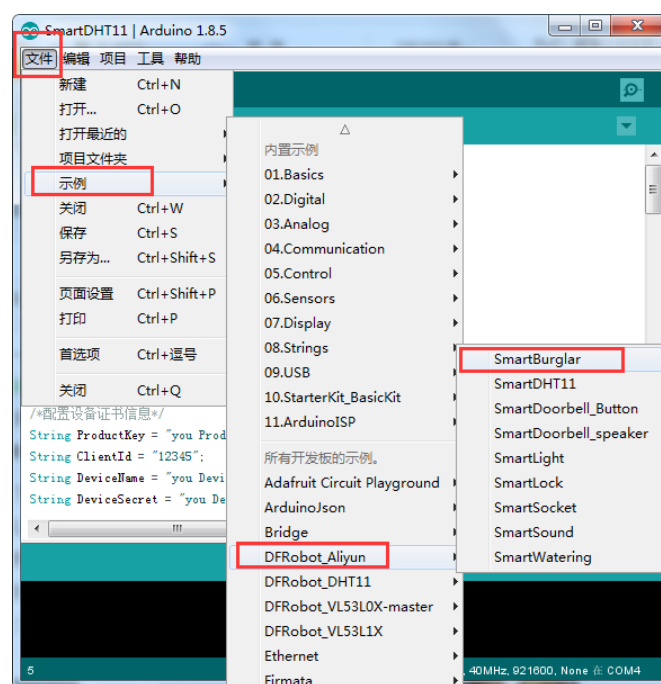


(4) 点击设备触发节点的**部署**按钮，点击**启动**，如下图所示：



如此，一个名为“智能防盗”的服务就完成了。

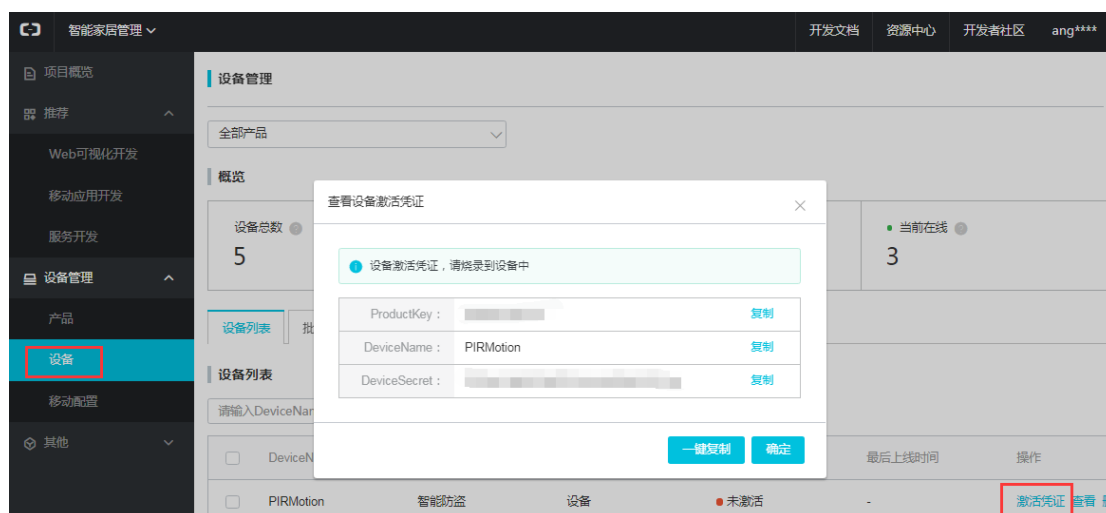
8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartBurglar** 项目，如下图所示：



① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示) 更换为可用的 WiFi。

```
/*配置WIFI名和密码*/  
const char * WIFI_SSID      = "WIFI_SSID";  
const char * WIFI_PASSWORD = "WIFI_PASSWORD";
```

② 填写设备证书信息。获取设备 **PIRMotion** 的设备信息，如下图所示：



将获取到的 **PIRMotion** 的设备证书信息，复制粘贴到如下图所示的程序模块中。

```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

③ 替换订阅、上报 Topic。在阿里云 IoT 上找到此应用创建的设备 PIRMotion 的 Topic 列表，如下图所示，并将它替换到程序的 TOPIC 设置中：



```
/*需要上报的TOPIC*/
const char * pubTopic = "you pub Topic";//*****post
```

④ 替换产品标识符。在设备管理的产品项，找到名为智能防盗的产品，点击查看，点击功能定义，即可在自定义功能栏看到标识符，如下图所示，将其替换到程序中的产品标识符即可。

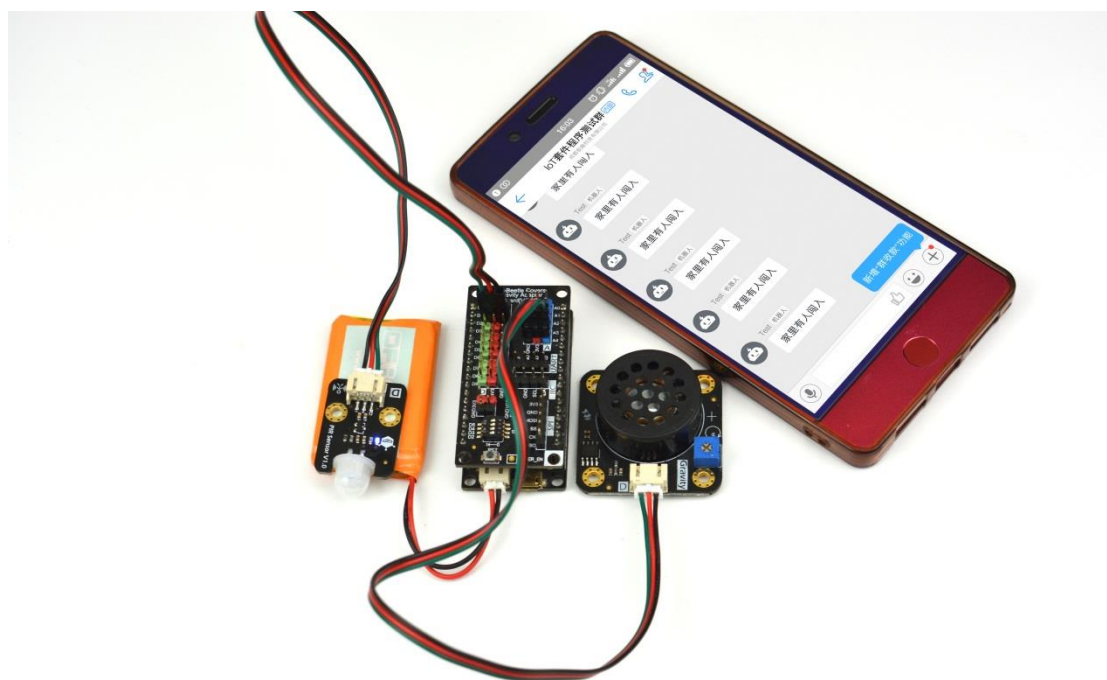
```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

自定义功能 ● 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	防盗警告	BurglarWarning	bool (布尔型)	布尔值：无人闯入 - 0；有人闯入 - 1；	编辑 删除

⑤ 编译上传至 FireBettle Board-ESP32 主板。选好开发板类型和端口号后，点击编译上传即可将程序上传至 ESP32 主板。

此时设备即成功与 Aliyun 平台上的 PIRMotion 设备关联，此时当 PIR 传感器检测到有人，即调动喇叭报警，同时钉钉机器人会把“家里有人闯入”消息推送至手机钉钉，如下图所示：



3.6 应用六：智能门锁

通过前几个应用的学习，了解了如何在阿里云 IoT 上创建移动应用开发和服务开发。智接下来，将结合这两个服务，组成一个新应用——智能门锁。

应用目标：通过手机 App 控制继电器的开关从而打开锁，若监测到到锁是开着的，则将消息“门没关好!!”推送至手机钉钉。

所需元件

继电器模块 x1

电池锁 x1

FireBettle Board-ESP32 x1

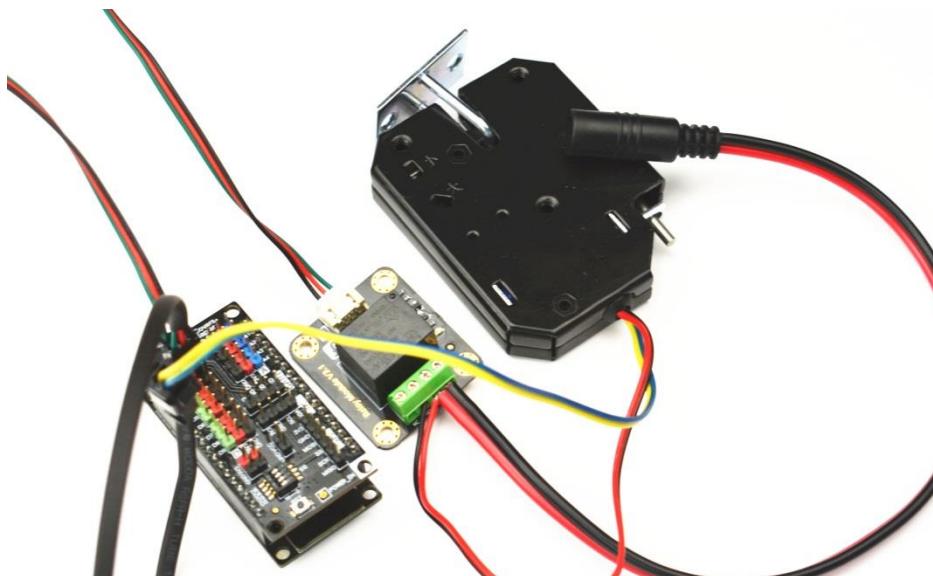
FireBettle Gravity 扩展板 x1

智能手机 x1

硬件连接

把 FireBettle Gravity 扩展板直插在 FireBettle Board-ESP32 主板上，再把数字继电器模块接在扩展板的 D2 数字引脚上，将电池锁的反馈线（蓝黄线）通过转接线接 D3 数字引脚上，确保电源引脚、接地引脚和信号引脚都连接正确。将电池锁的电源线经电源配线和继电器，接到 12V 电源上。

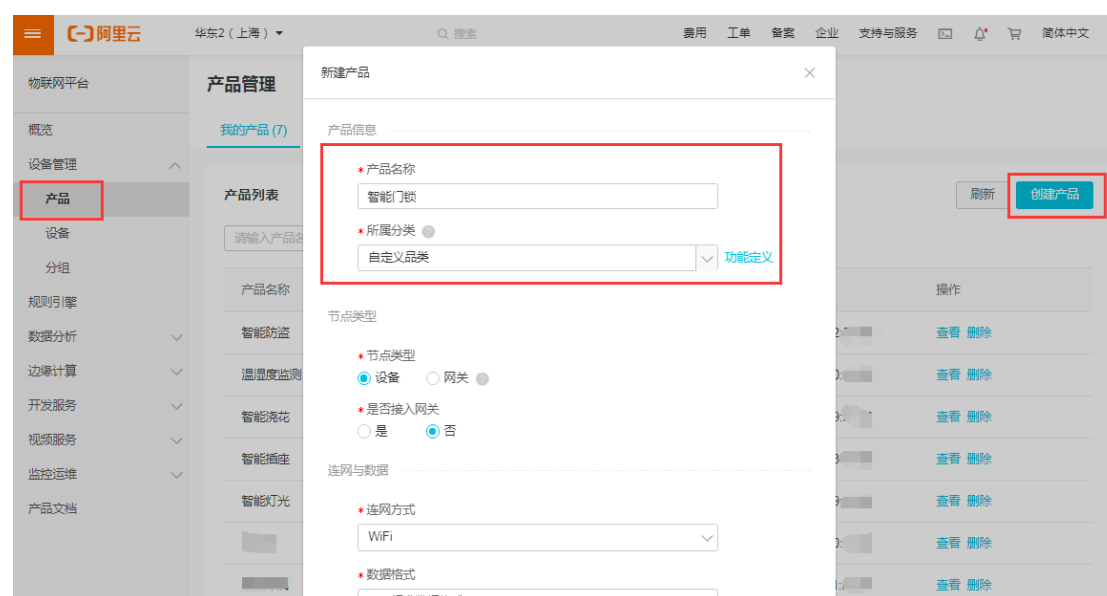
电池锁原理：电池锁有 2 根线，反馈线（黄蓝色）和电源线（焊锡红黑线）。电源线两端工作电压 12V，通电时开锁，电池锁关闭时，反馈线互相导通，打开时，不通。按其反馈线接法，当电池锁关闭时，在 D3 引脚处读到的信号为 0，开启时，信号为 1。



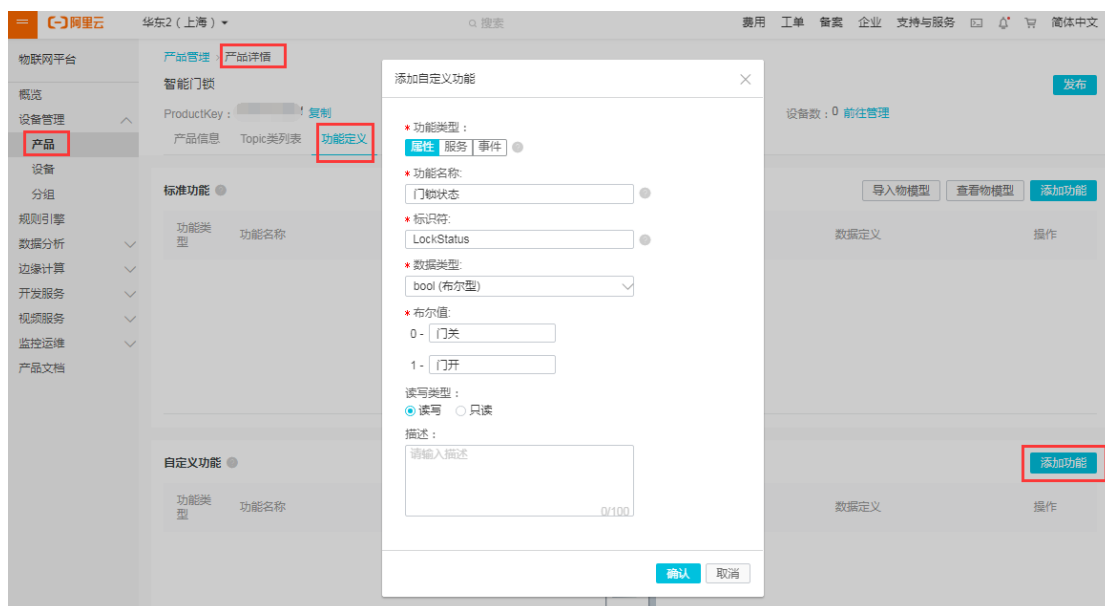
应用分析：智能门锁项目需要创建 2 个不同的开发服务，即移动应用开发来控制继电器通断，钉钉服务开发“门没关好!!”的消息推送至钉钉。其次将智能防盗产品的设备的证书信息、Topic 列表，产品标识符替换到 Arduino 程序的相应位置中。

制作步骤

1. 打开浏览器，输入网址 <https://iot.aliyun.com>, 登录并进入 **物联网平台**;
2. 创建一个名为 **智能门锁** 的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



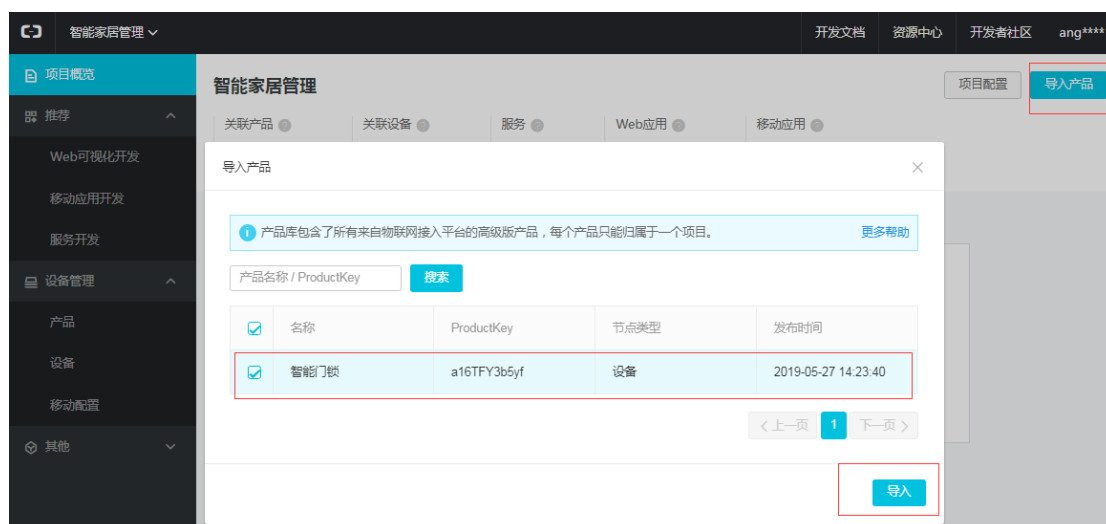
3. 为智能门锁产品定义一个功能名为 **“门锁状态”** 的属性，自定义一个产品标识符，定义门开为 1，门关为 0。如下图所示：



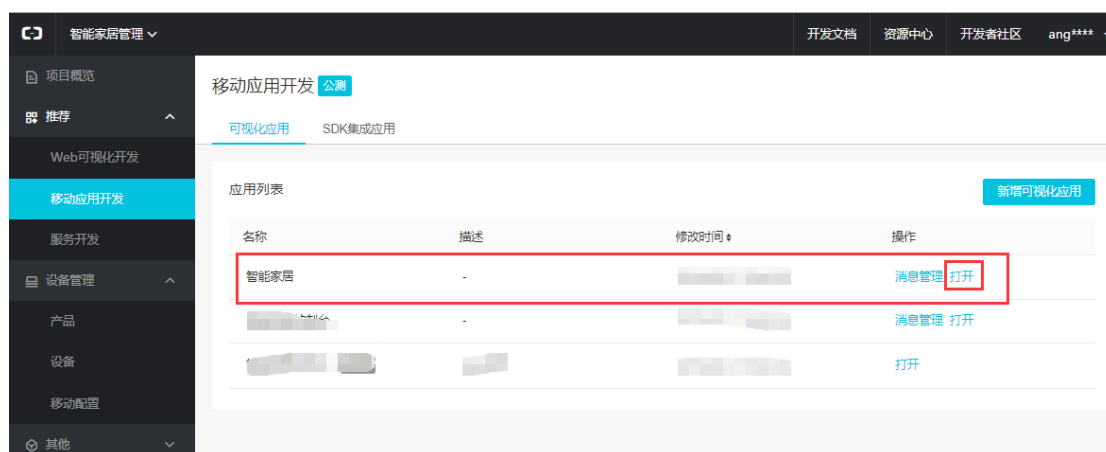
4.为**智能门锁**添加一个名为 **Lock** 的设备，如下图所示：



5.将**智能门锁**产品导入**智能家居管理**项目中，如下图所示：

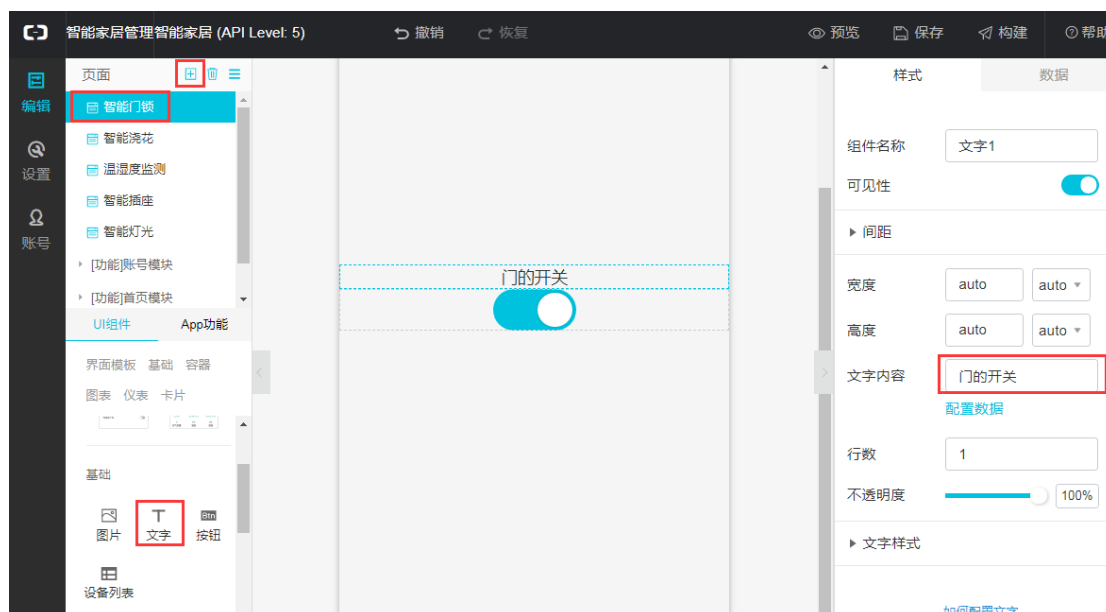


6. 选中左侧导航栏的**移动应用开发**,在应用列表找到**智能家居**,单击**打开**,如下图所示:

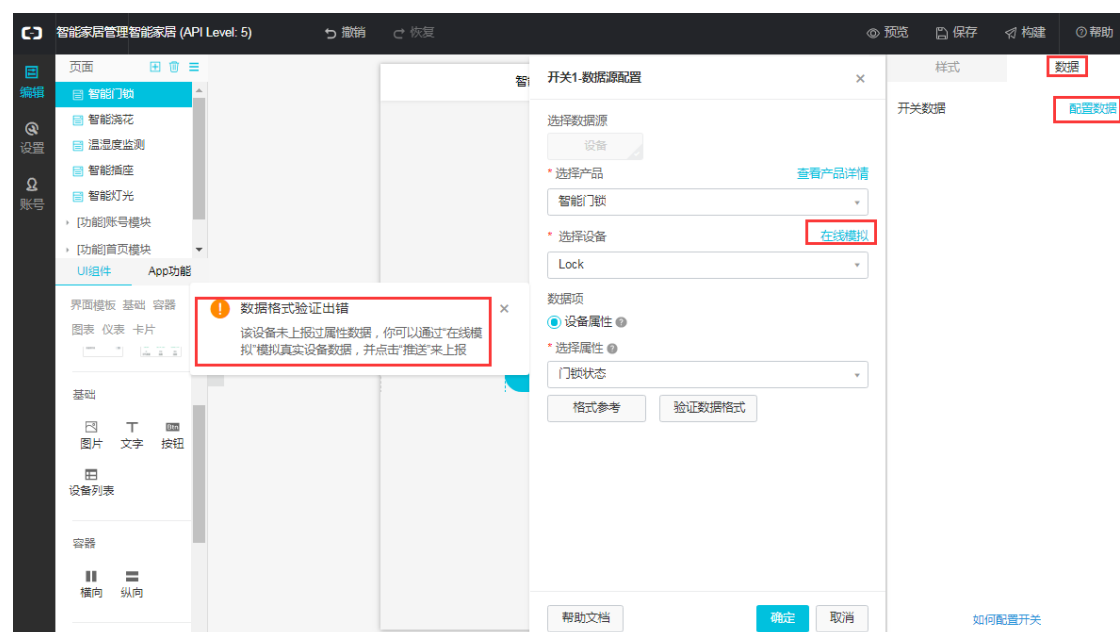


7. 智能门锁 App 设计与开发。

① 点击**自定义页面**后的+号, 编辑, 将“自定义新增页 1”改为**智能门锁**, 用鼠标将可视化组建下的**文字**、**开关**控件拖动到中间的页面上, 如下图所示:



②选中中间页面的开关，点击**数据->配置数据**，在弹出的选项框里选择相应的选项，并点击确定，若出现下图所示“数据格式验证出错”，这是因为该设备被创建以来，还没有被使用过，可以通过**在线模拟**解决该问题。



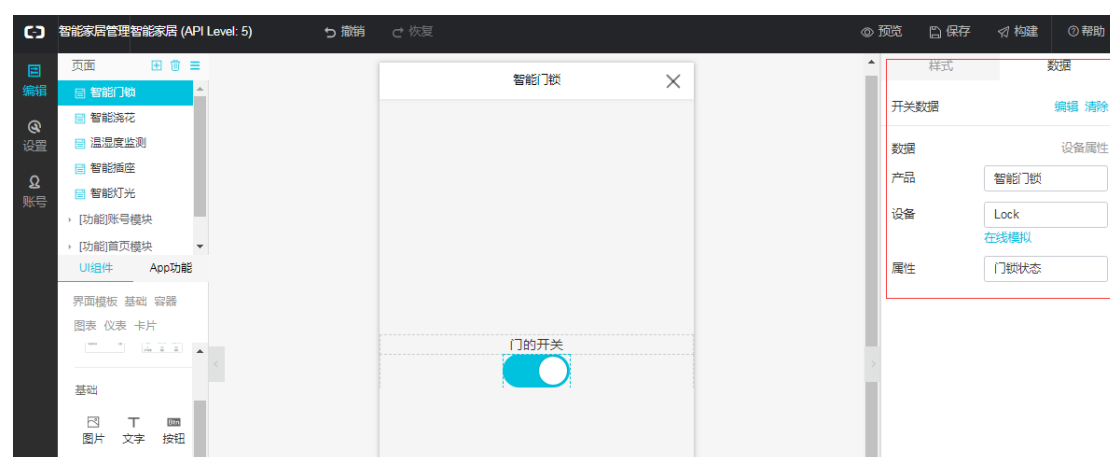
③点击“在线模拟”，进入如下所示页面，（若真实设备在线，则可选择“调试真实设备”）这里我们选择**启动虚拟设备**。



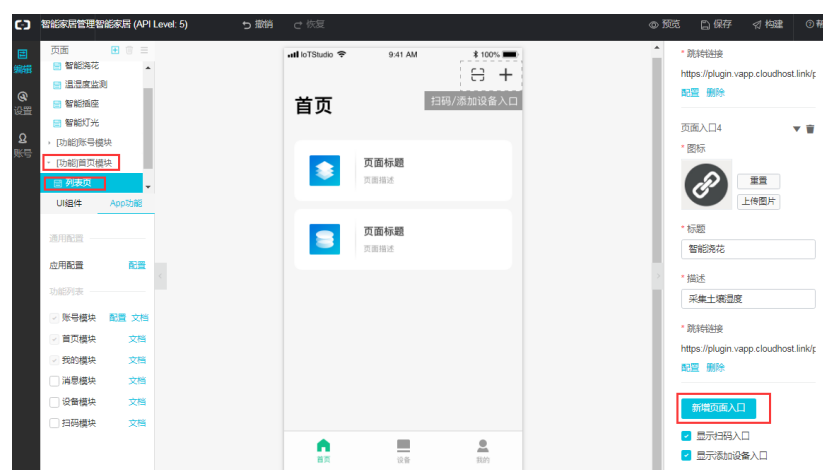
④启动虚拟设备后，会进入如下所示页面，任意选择一种状态，点击**推送**即可，稍后会收到设备上报信息。



⑤收到设备上报数据后，在回到移动应用开发页面，点击**确定**，即可完成“开关”的配置了，如果你不知如何配置开关可以点击右下角的“如何配置开关”，这样就将指示灯链接到“Lock”设备上了。



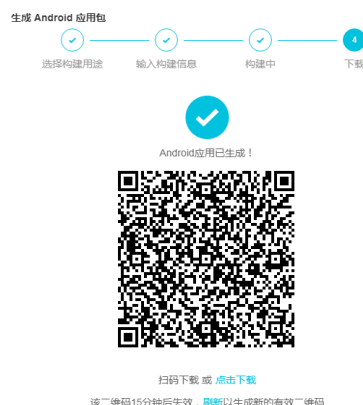
⑥配置**首页模块**。这一步是为了设置一个导航栏，链接到上面设计的“智能门锁”页面，。选中**[功能]首页模块->列表**，选中**新增页面入口**，如下图所示：



在弹出的“**页面入口 1**”下，完成相应信息的选择，将其链接到刚才设计的“智能灯光”页面，如下图所示，点击**确定**完成配置后，别忘了点击**保存**。

⑦账号配置。由于在应用一中已经添加了账号，这里就无需再添加了，直接构架安装包即可。

⑧安装包生成。在**编辑**页面，点击**构建**，选择 **Andriod 构建**或 **iOS 发构建**，如下图所示：

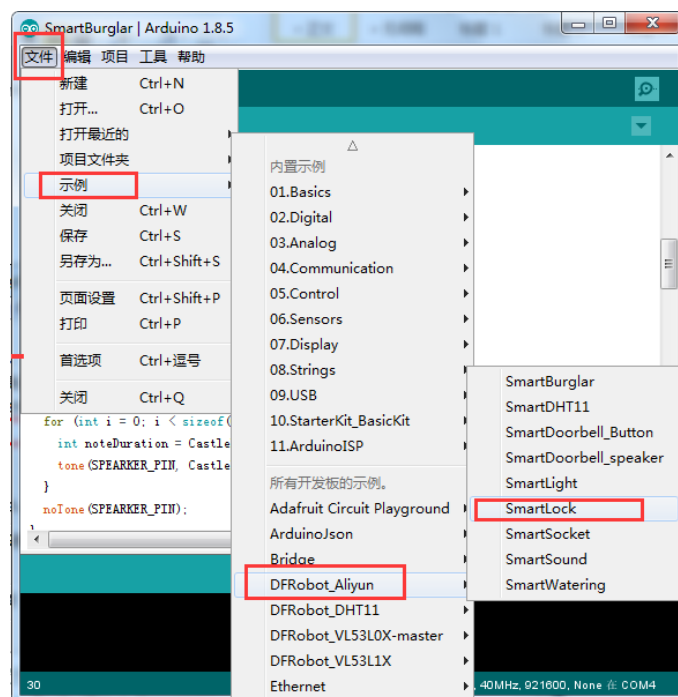


⑨手机 App 下载。手机微信或其他二维码扫描器扫描下载并安装该 App，用刚才创建的账号登录，即可打开 App 控制门锁设备的开关。

打开名为智能家居控制台的 App，如果已经登录，则会进入 App 首页，如下图所示：



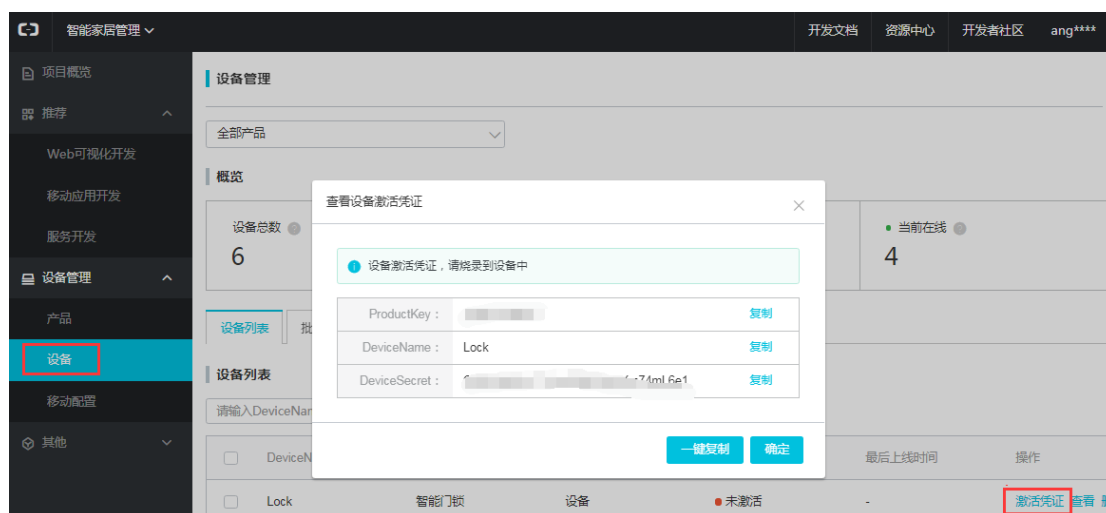
8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartLock** 项目，如下图所示：



① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示) 更换为可用的 WiFi。

```
/*配置WIFI名和密码*/  
const char * WIFI_SSID      = "WIFI_SSID";  
const char * WIFI_PASSWORD = "WIFI_PASSWORD";
```

② 填写设备证书信息。获取设备 **Lock** 的设备信息，如下图所示：



将获取到的 **Lock** 的设备证书信息，复制粘贴到如下图所示的程序模块中。

```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

③ 替换订阅、上报 Topic。在阿里云 IoT 上找到此应用创建的设备 Lock 的 Topic 列表，如下图所示，并将它替换到程序的 TOPIC 设置中：



```
/*需要上报和订阅的两个TOPIC*/
const char * subTopic = "you sub Topic"; /****set
const char * pubTopic = "you pub Topic"; /****post
```

④ 替换产品标识符。在设备管理的产品项找到名为智能门锁的产品，点击查看，点击功能定义，即可在自定义功能栏看到标识符，如下图所示，将其替换到程序中的产品标识符即可。

自定义功能 ● 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	门锁状态	LockStatus	bool (布尔型)	布尔值：门关 - 0；门开 - 1；	编辑 删除

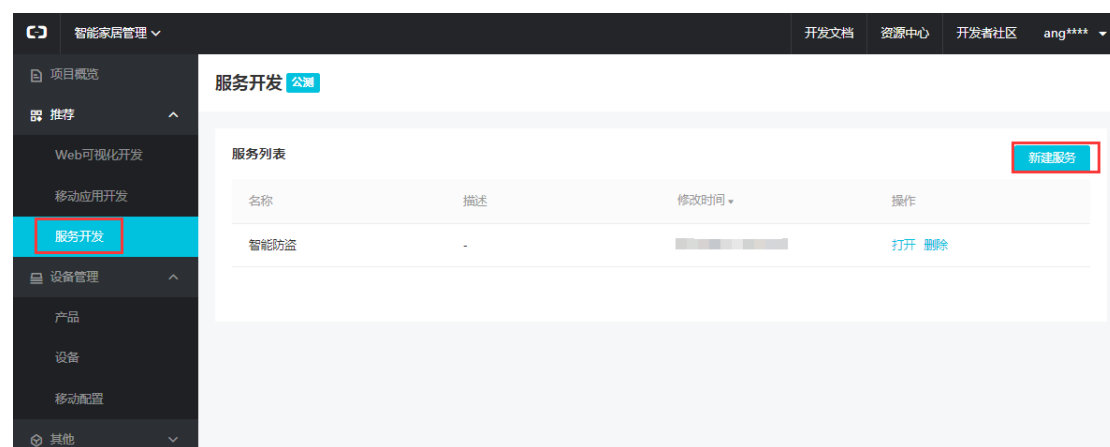
```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

⑤ 编译上传至 FireBettle Board-ESP32 主板。选好开发板类型和端口号后，点击编译上传即可将程序上传至 ESP32 主板。

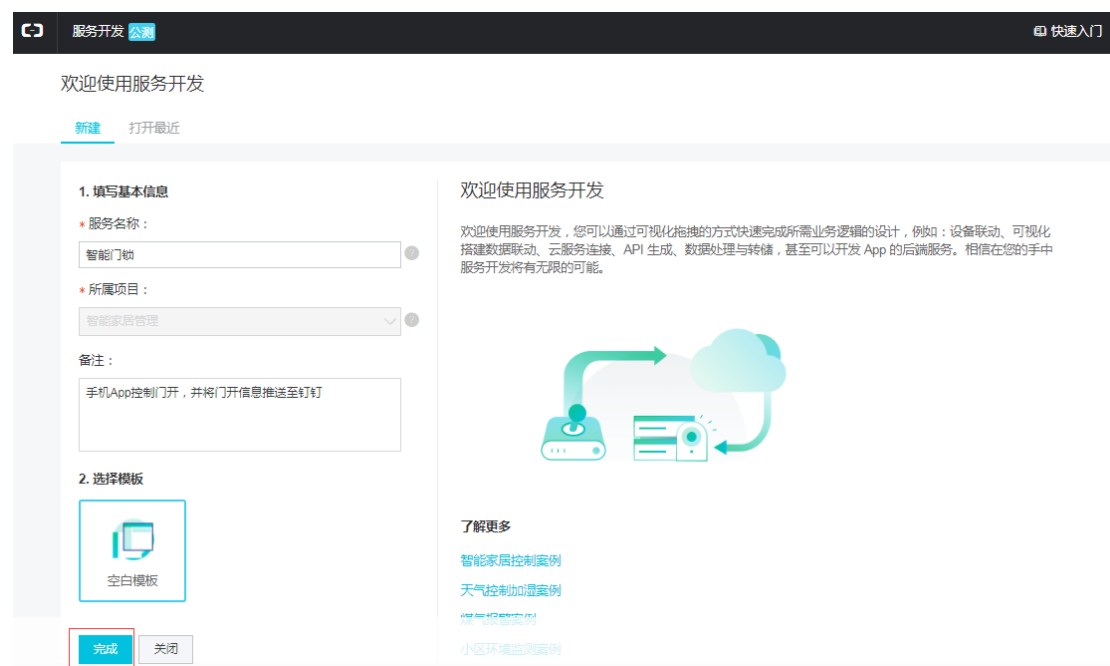
此时设备即成功与 Aliyun 平台上的 Lock 设备关联，打开下载的智能家居控制台，选择智能门锁，即可进入控制页面，点击按钮控件，即可控制继电器开关（继电器的红色灯亮，

表示开), 从而开锁。接下来, 就开始服务开发, 将消息推送至钉钉了。

9. 创建一个名为**智能门锁**的服务开发。选中左侧导航栏的**推荐->服务开发**, 点击**新建服务**, 如下图所示:

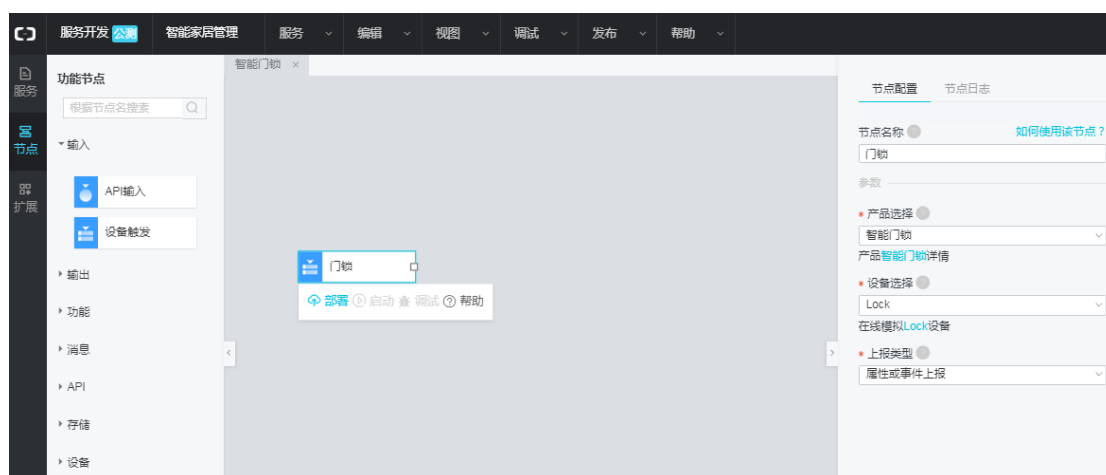


根据页面提示, 创建一个名为“**智能门锁**”的服务, 该服务内容当检测到门锁开了后, 将信息推送至钉钉, 如下图所示:

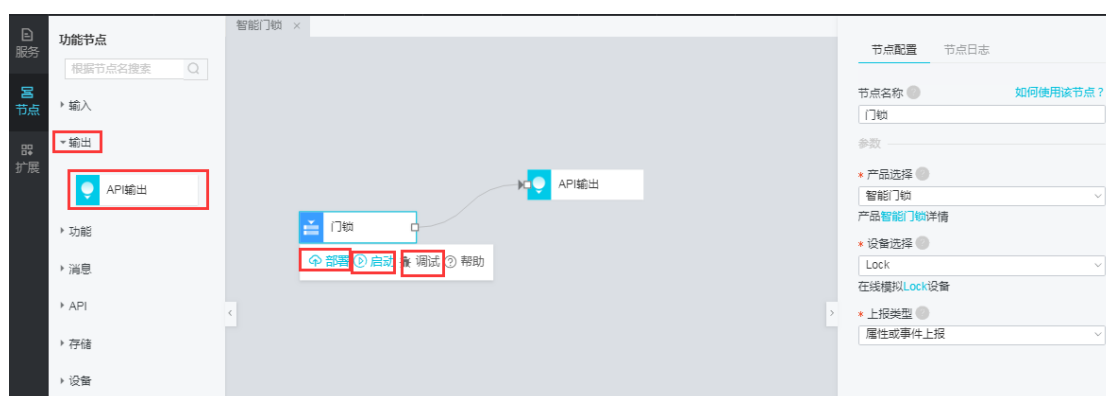


10. 点击**完成**, 进入**服务编辑器**。具体开发技巧请参考网址:
<https://linkdevelop.aliyun.com/studioservice-doc#index.html>

① 点击左侧导航栏的**节点**, 在节点列表中选择**输入->设备触发**, 并将其拖动到服务画布内, 把这个设备触发节点改为**门锁**; 右侧栏配置项分别选择“**智能门锁**”产品, 创建的设备名以及选择监听“**属性或事件上报**”, 如下图所示:

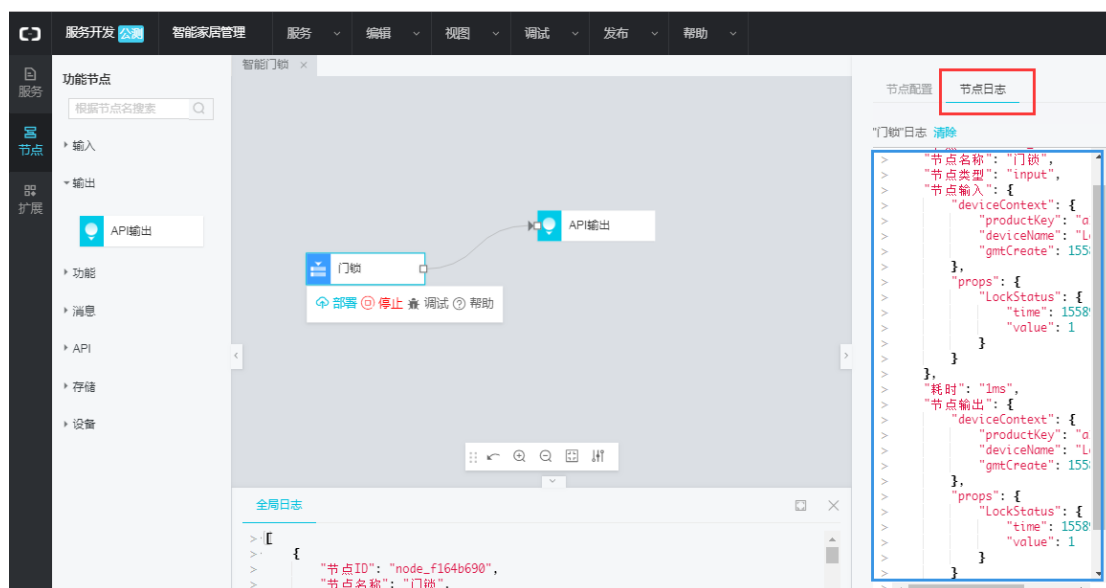


②获取“门锁”节点的节点输出信息。“门锁”节点配置完成后，选中左侧节点列表的**输出->API 输出**，点击**部署->启动->调试**，如下图所示：

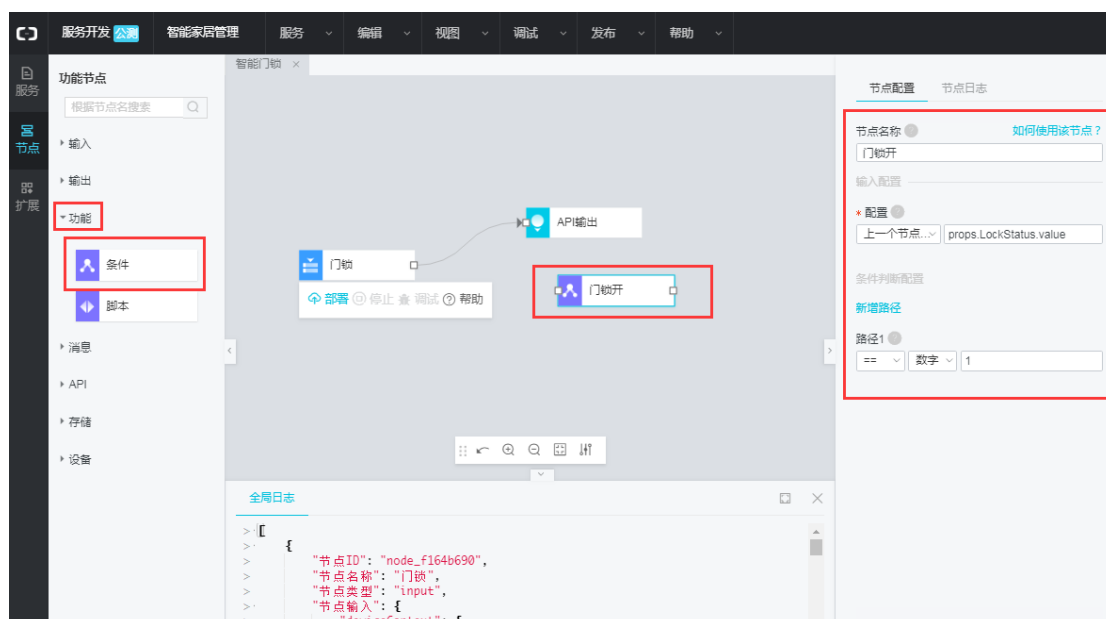


在弹出的“在线调试”页面中选择某个参数，点击推送，回到服务开发页面，点击节点日志，即可看到“门锁”节点的输入输出节点信息，如图：

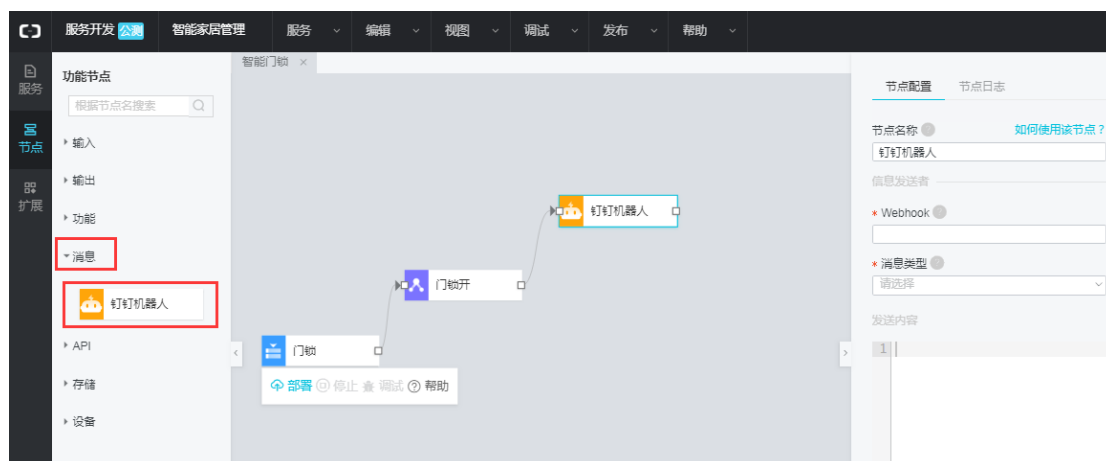




③ 添加一个“**条件**”节点，条件节点相当于一个 if-else 判断。选中左侧节点列表的功能->条件节点，将其改为“门锁开”，并根据上面得到的“门锁”输出节点信息配置“门锁开”节点的输入设置，将其配置为“上一个节点”+“props.LockStatus.value”；将路径 1 的第二个选择框设置为“数字”，第三个框输入 1（表示门开了），可以参考[如何使用该节点？->如何配置条件节点？](#)



④ 删除“API 输出”节点，选中消息->钉钉机器人，添加钉钉机器人节点。并将门锁、门锁开、钉钉机器人节点连接起来。

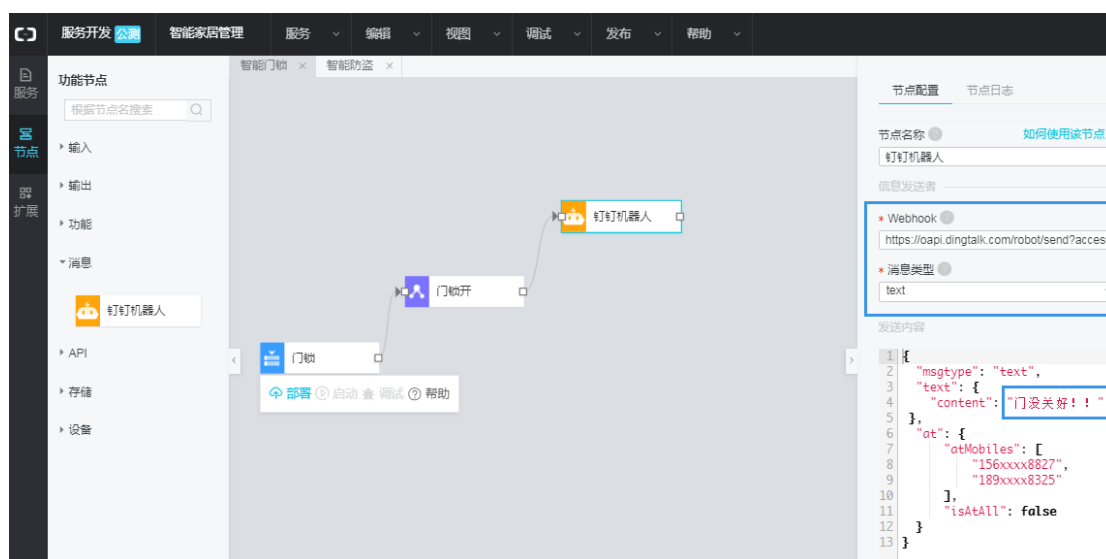


⑤配置钉钉机器人节点。若不知道如何配置钉钉机器人，可参考[如何使用该节点？->如何配置钉钉机器人节点？](#)

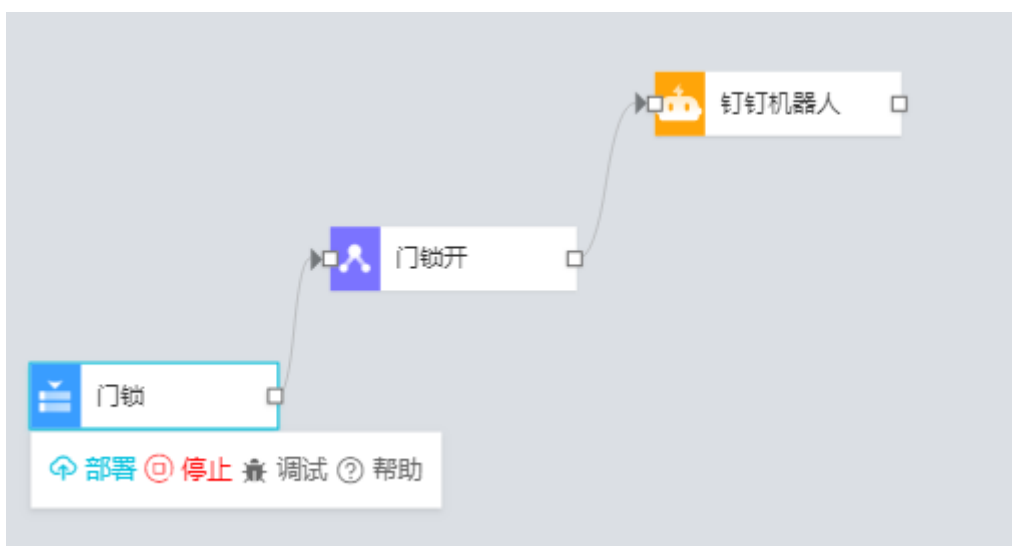
a.获取 Webhook。在获取之前，你需要在钉钉群中，添加一个钉钉机器人(注意查看钉钉机器人的 Webhook 时只能用电脑端的钉钉)，具体添加方法请参考[如何使用该节点？->如何配置钉钉机器人节点？](#)最终获取的 Webhook，复制粘贴至右侧栏的 Webhook 处，如下图所示：



b.选择“text”作为消息类型，并在 content 的双引号内填入信息“门没关好!!”。

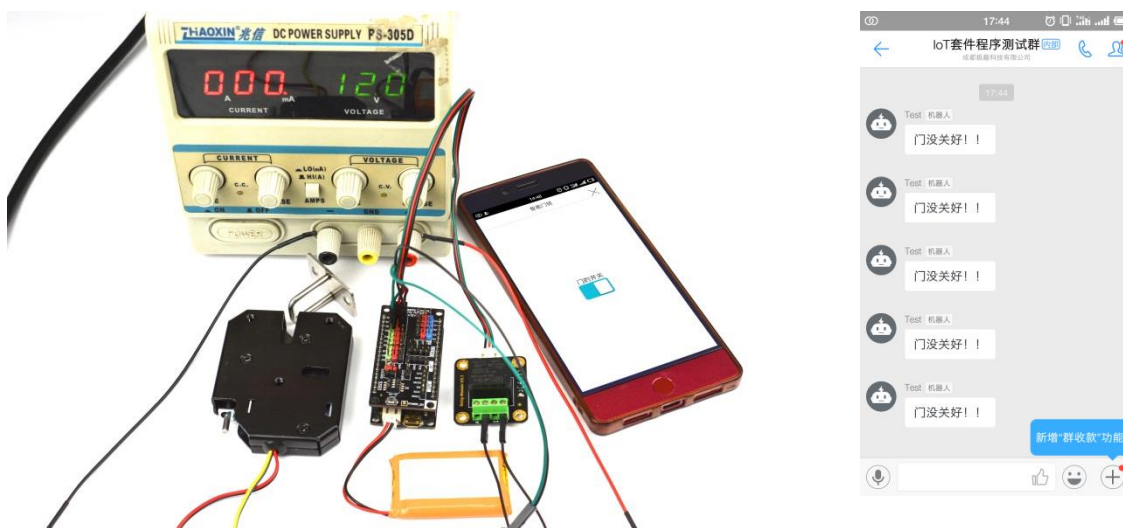


(4) 点击设备触发节点的**部署**按钮，点击**启动**，如下图所示：



如此，一个名为“智能门锁”的服务就完成了。

此时，打开智能家居控制台 App，点击“智能门锁”，进入控制页面，点击按钮，即可实现开锁，同时将消息“门没关好!!”推送至手机钉钉，如下图所示：



3.7 应用七：智能提示音

前面应用五、六小节，介绍了如何创建并开发输入节点为设备触发类型的服务。这节将介绍服务开发的另一种输入节点：**API** 输入。

应用目标：要求创建一个服务，有一个功放喇叭，根据天气的温度结果，决定播放那一首歌。

所需元件

功放喇叭 x1

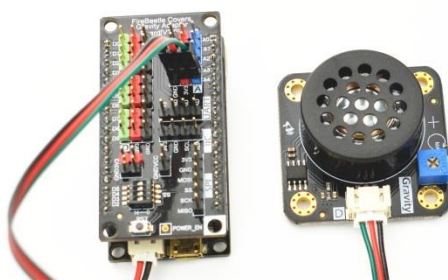
FireBettle Board-ESP32 x1

FireBettle Gravity 扩展板 x1

智能手机 x1

硬件连接

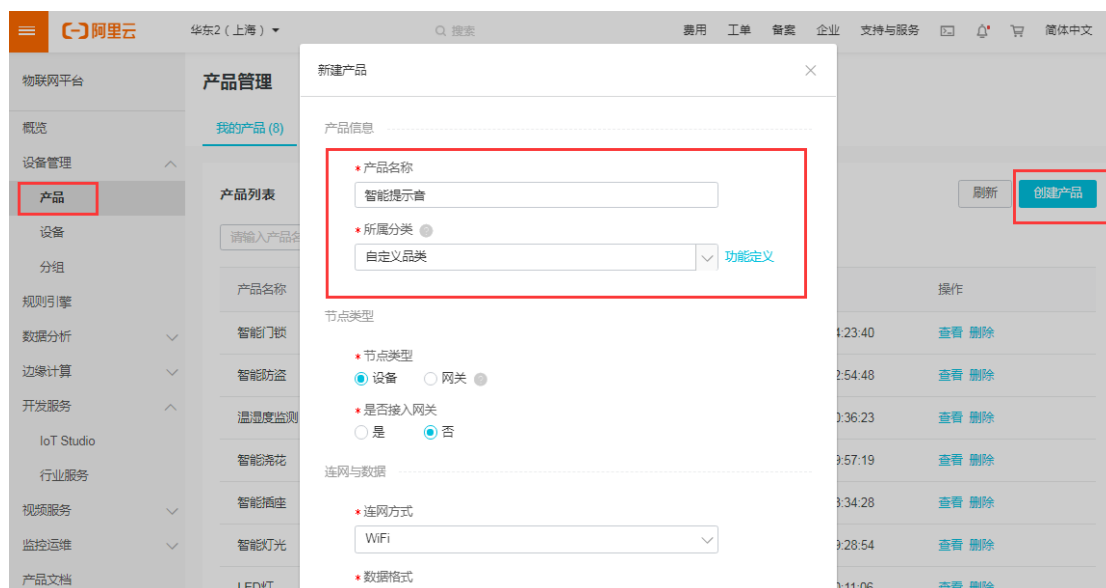
把 FireBettle Gravity 扩展板直插在 FireBettle Board-ESP32 主板上，再把功放喇叭接在扩展板的 A4 引脚，确保电源引脚、接地引脚和信号引脚都连接正确。



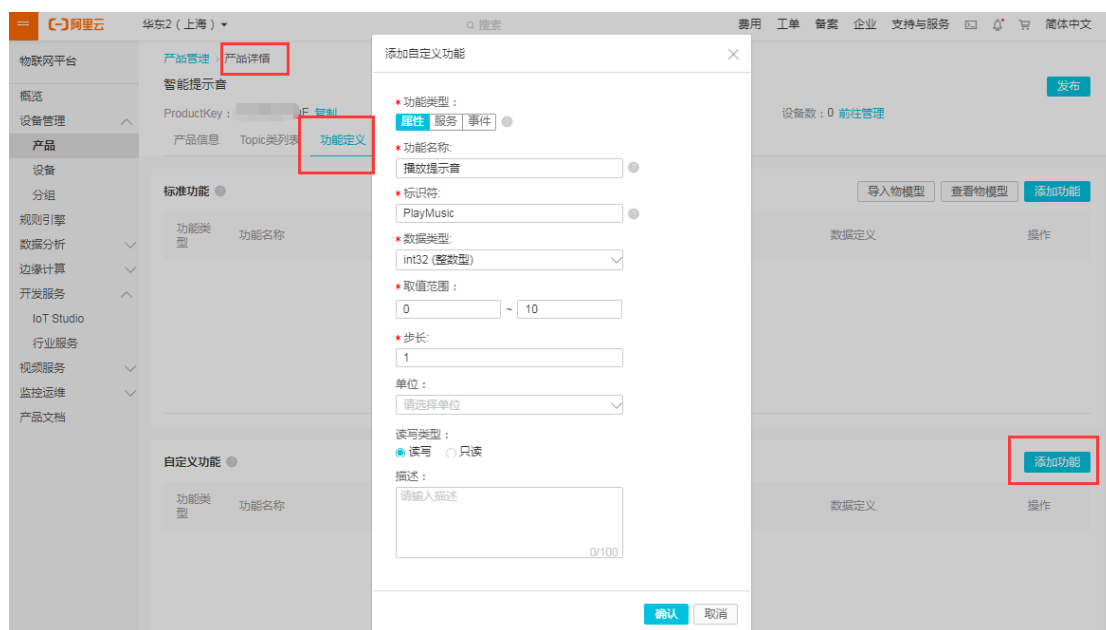
应用分析：此应用在创建产品、设备和服务的同时，还需要购买墨迹天气服务（免费购买）。

制作步骤

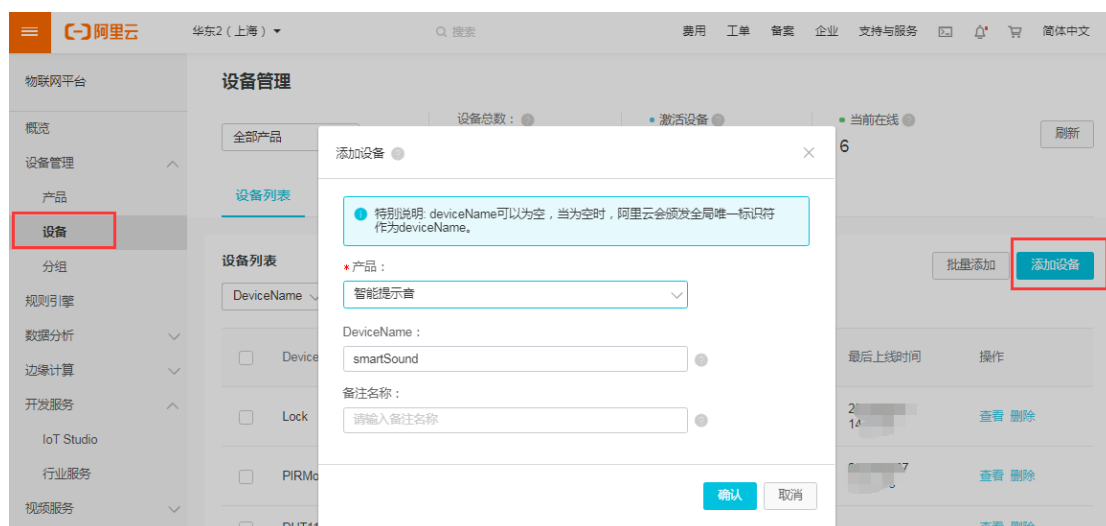
1. 打开浏览器，输入网址 <https://iot.aliyun.com>, 登录并进入**物联网平台**；
2. 创建一个名为**智能提示音**的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



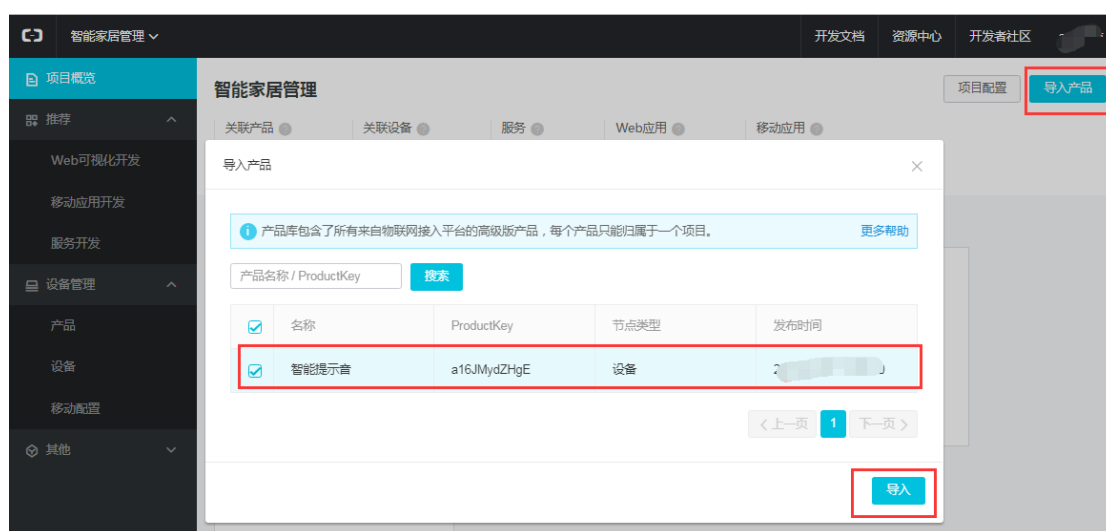
3. 为智能防盗产品定义一个功能名为“播放提示音”的属性，自定义产品标识符，定义其数据为整型，取值 0~10，步长为 1，如下图所示：



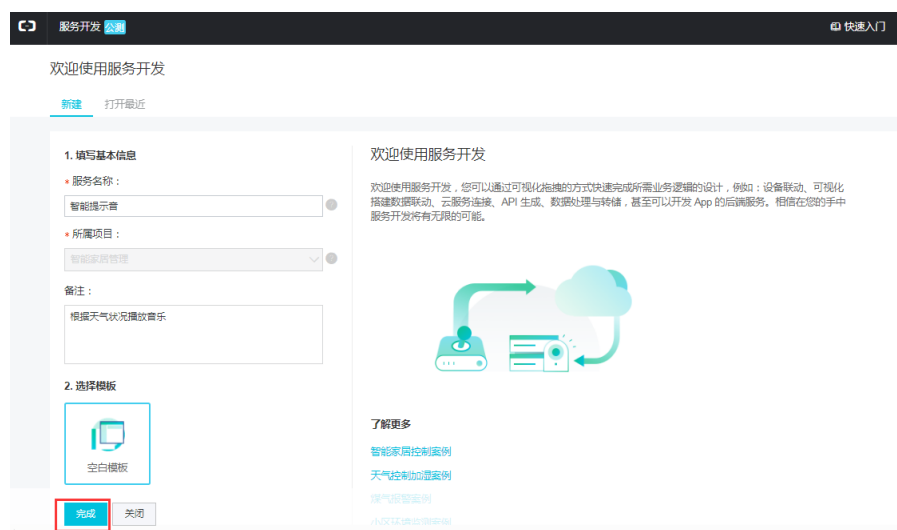
4. 为智能提示音添加一个名为 SmartSound 的设备，如下图所示：



5. 将**智能提示音**产品导入**智能家居管理**项目中, 如下图所示:

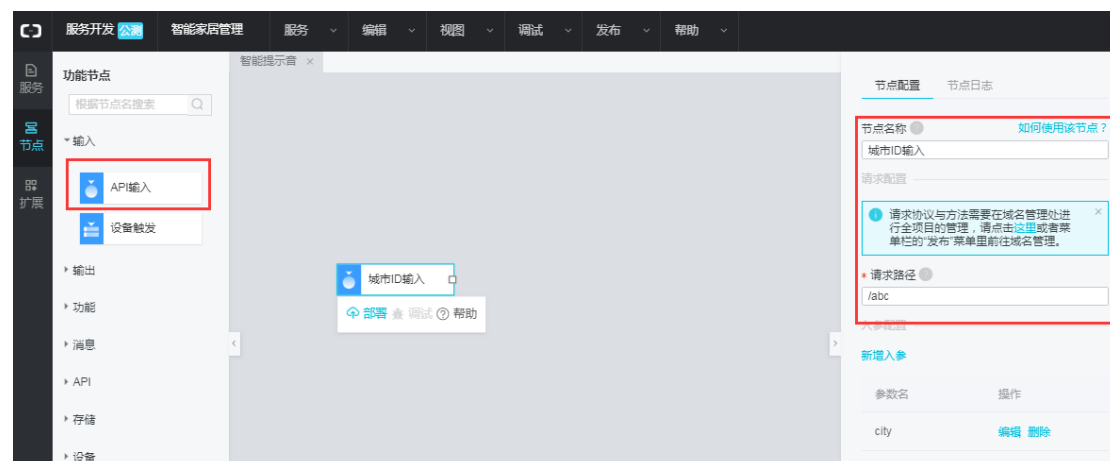


6. 选中左侧导航栏的**推荐->服务开发**, 点击**新建服务**, 根据页面提示, 创建一个名为“**智能提示音**”的服务, 该服务主要内容为根据某城市天气的信息智能播放不同的音乐, 如下图所示:

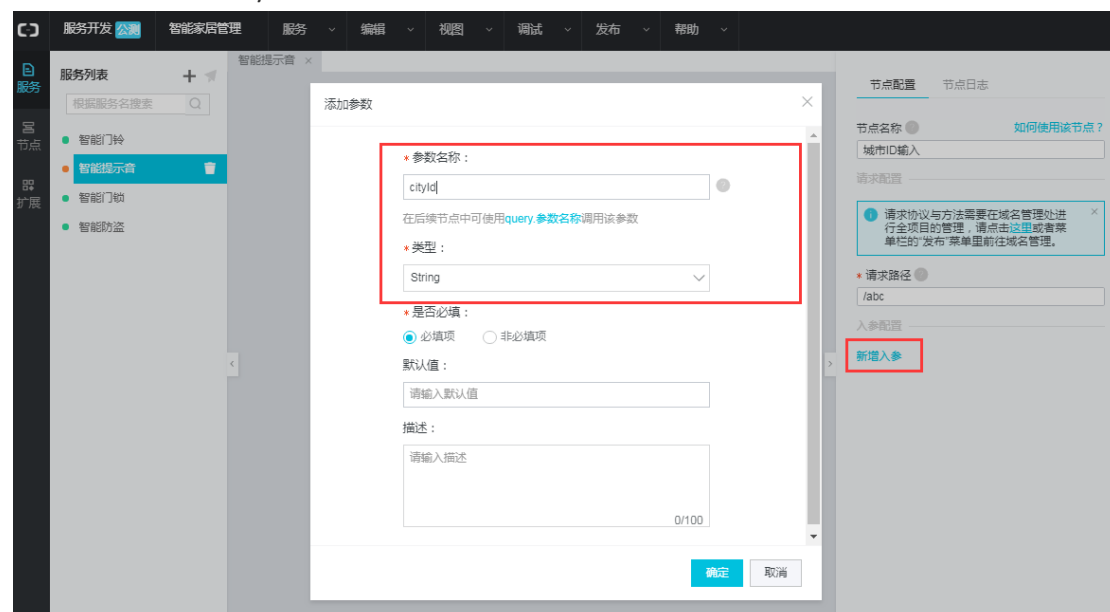


7. 点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址：
<https://linkdevelop.aliyun.com/studioservice-doc#index.html>

❶ 点击左侧导航栏的**节点**，在节点列表中选择**输入->API 输入**，并将其拖动到服务画布内，把这个设备触发节点改为**城市 ID 输入**；如下图所示：



❷ 添加一个 API 输入参数。由于后面查询天气的时候需要用到所在城市，因此 API 参数设定为字符“cityId”，配置如图所示。

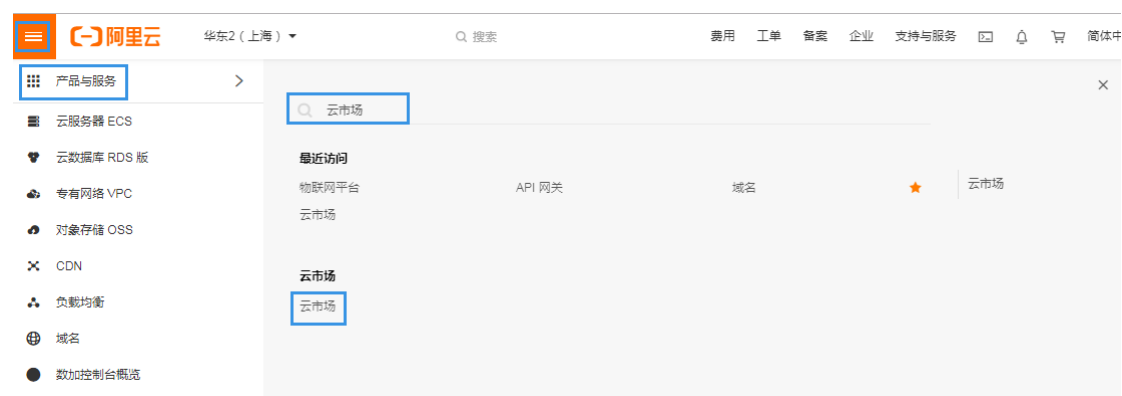


❸ 添加一个**云市场 API** 节点。选中 **API->云市场 API**。并将其改为**墨迹天气**，如下图所示：



a. 购买墨迹天气服务。打开新页面前往阿里云市场选择 API，购买墨迹天气服务，该服务是免费的，具体购买方式请参考阿里云 IoT 案例“[天气加湿器](#)”

b. 获取 AppCode。墨迹天气购买成功后，回到物联网平台，选中左侧导航栏的“产品与服务”，搜索“云市场”即可找到 AppCode。如下图所示：



复制该 AppCode 值到墨迹天气的配置栏“AppCode”即完成了 AppCode 的配置，如下图所示：



c. 获取精简实况的调用地址和请求方式。回到刚才的**云市场**页面，点击“**免费版气象服务 (cityid) - 墨迹天气**”，进入购买页面，下拉点击**精简实况**，即可获得**精简实况的调用地址和请求方式**，如下图所示：

The screenshot shows the product page for '免费版气象服务 (cityid) - 墨迹天气'. The left sidebar contains navigation links like '订单列表', '需求管理', etc. The main content area shows product details, including '商品名称', '商品类型', '状态', and '到期时间'. Below this, there's a section for '精简实况' (Simplified Real-time) with a red box highlighting the '调用地址' (Call Address) and '请求方式' (Request Method). The '调用地址' is 'https://api.openweathermap.org/data/2.5/weather?appid=...' and the '请求方式' is 'POST'. There are also links for '立即购买' and '加入购物车'.

复制该调用地址即可配置“墨迹天气”的调用地址，并将该请求方式改为“POST”，如下图所示：

The screenshot shows a configuration form with two fields: '* 请求方式' (Request Method) and '* 调用地址' (Call Address). The '请求方式' is set to 'POST' in a dropdown menu. The '调用地址' field is empty, with a red box highlighting it.

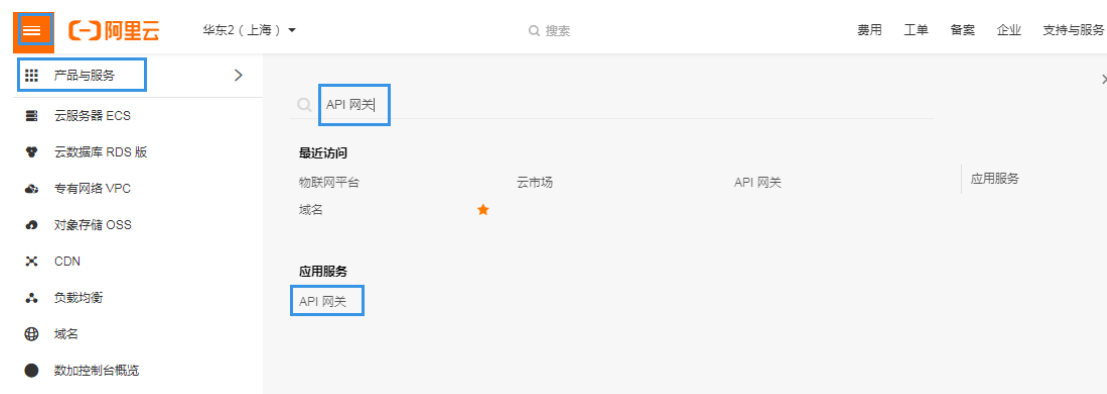
d. 获取成都温江的 cityid 参数。在刚才的购买页面上点击**产品详情->cityid 对照表**，即可获得各个城市的城市 ID，通过它我们可以指定获取某个城市的天气信息，如下图所示：

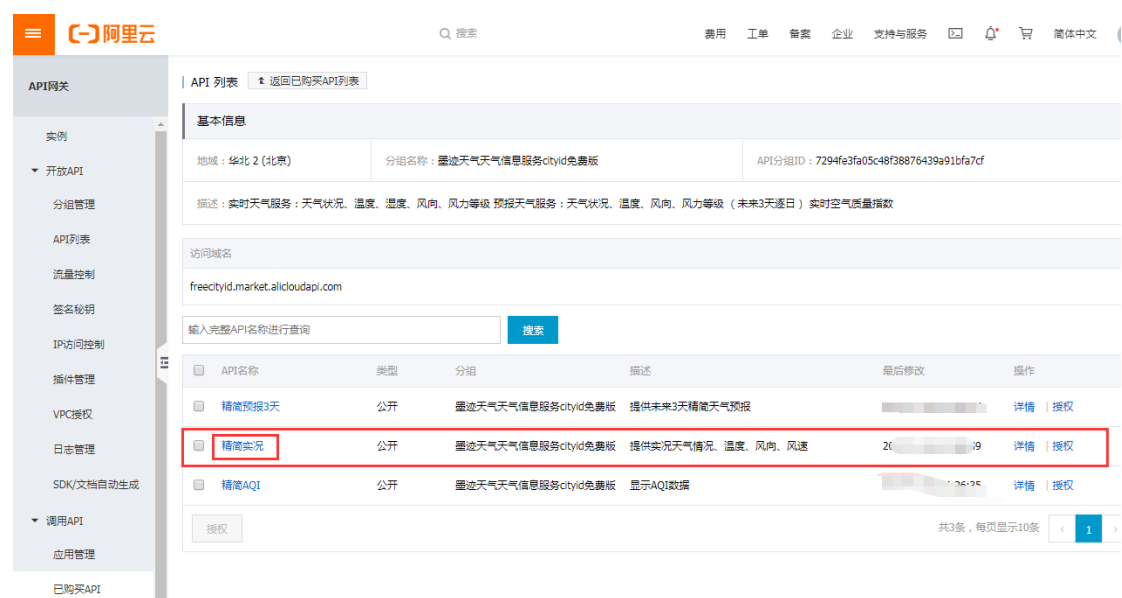


如成都温江的 ID 为 2637。

2151	2632	新疆维吾尔自治区	阿勒泰地区	阿勒泰地区	Altay Prefec	Aletaidiqu
2152	2635	四川省	成都市	成都市	Chengdu	chengdushi
2153	2636	四川省	成都市	新都区	Xindu Distric	xinduqu
2154	2637	四川省	成都市	温江区	Wenjiang D	wenjiangqu
2155	2638	四川省	成都市	都江堰市	Duijiangyan	dujiangyanchi
2156	2639	四川省	成都市	彭州市	Pengzhou	pengzhoushi
2157	2640	四川省	成都市	邛崃市	Qionglai	qionglaiishi

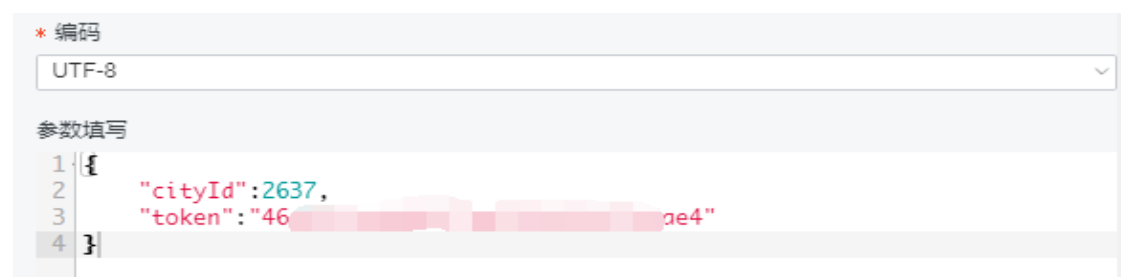
e. 获取 token 参数。回到物联网平台，选择左侧的产品与服务，搜索 API 网关，点击 API 网关，选择“华北 2（北京）”，选择左侧导航栏的调用 API->已购买 API。点击查看->精简实况。下拉网页即可找到 token 信息，如下图所示：





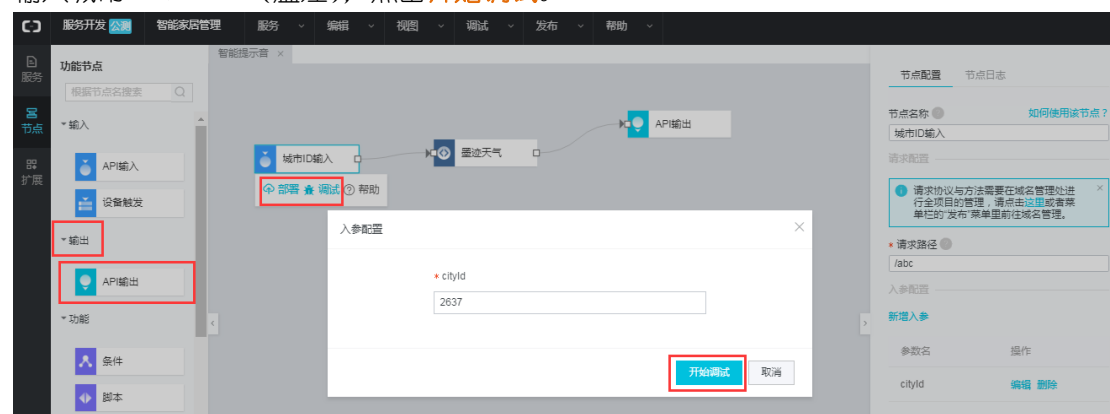
请求Body ☐ 使用二进制数据					
参数名称	类型	是否必填	示例值	描述	操作
cityId	STRING	必填	2	城市id	更多
token	STRING	可选	46[REDACTED]ae4	请求token (默认参数, 必填)	更多

f. 配置“墨迹天气”的编码方式和参数。如下图所示，

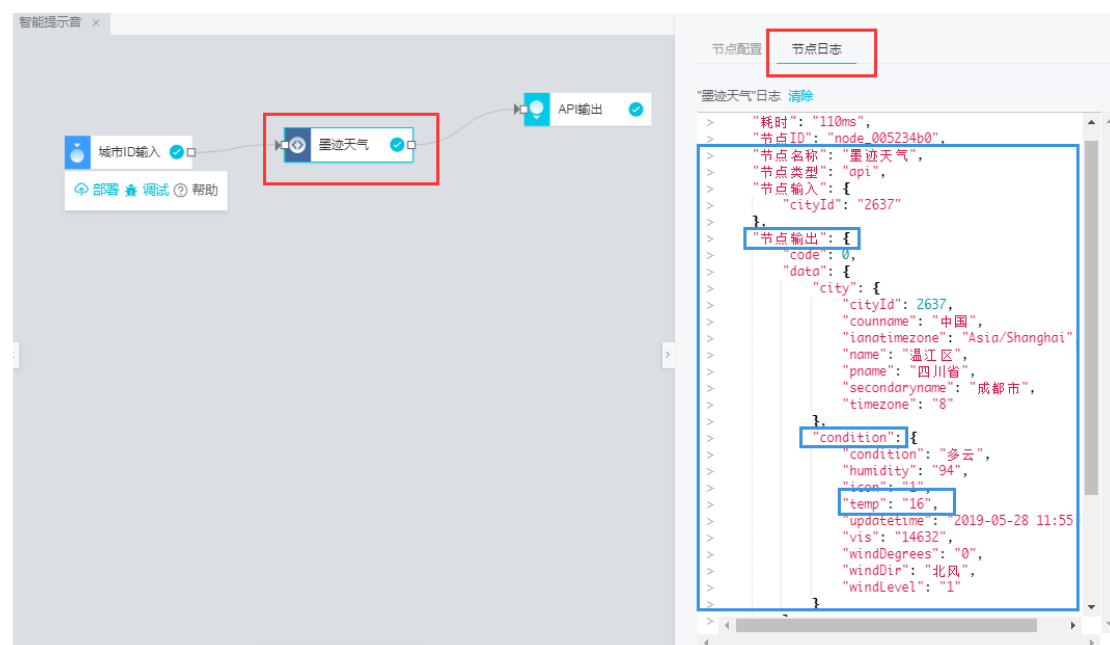


如此即可获取成都温江的天气信息了。由此“墨迹天气”节点则配置完成。

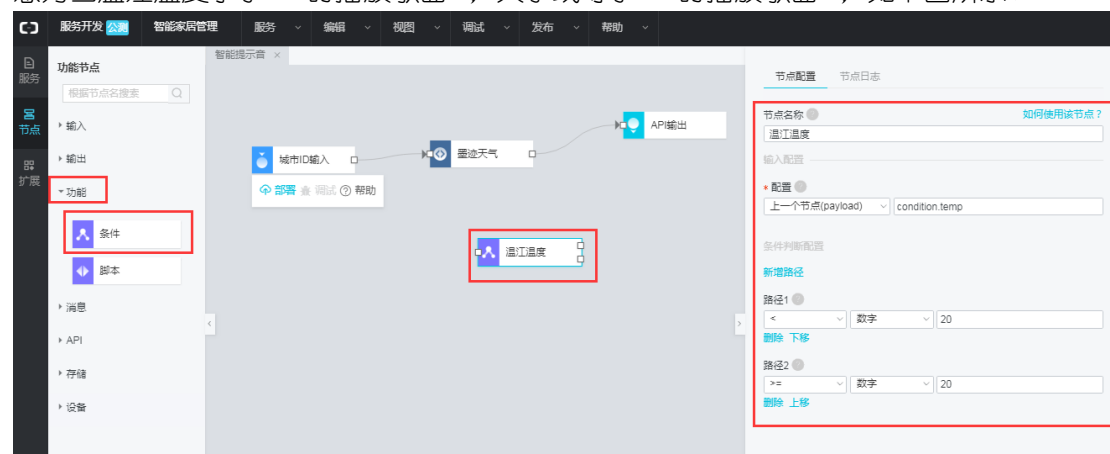
④ 获取**墨迹天气**节点的输出信息。选择左侧导航栏的**输出**→**API 输出**，点击**部署**→**调试**，输入城市 ID：2637（温江），点击**开始调试**。



鼠标选中**墨迹天气**节点，点击右侧栏节点日志，即可获取**墨迹天气**节点的输出信息，如下图所示：



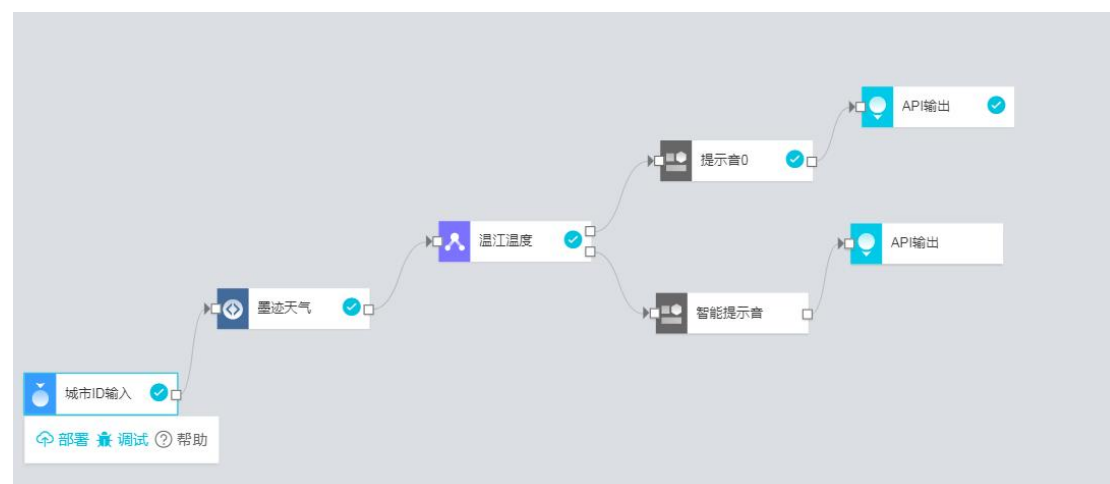
⑤ 添加**条件**节点信息。选择功能→条件节点，并将其节点名称改为“**温江温度**”配置为：
上一个节点+ `data.condition.temp`，新增一条路径，一条为**<数字 20**；一条为**>=数字 20**，
意为当温江温度小于 20 时播放歌曲 0，大于或等于 20 时播放歌曲 1，如下图所示：



⑥添加 2 个**智能提示音**设备。选中设备->智能提示音，一个连接到“**温江温度**”的条件节点路径 1 下，并将设备节点名改为“**提示音 0**”；另一个连接到“**温江温度**”的条件节点路径 2 下，并将歌曲改为“**提示音 1**”配置如下所示：



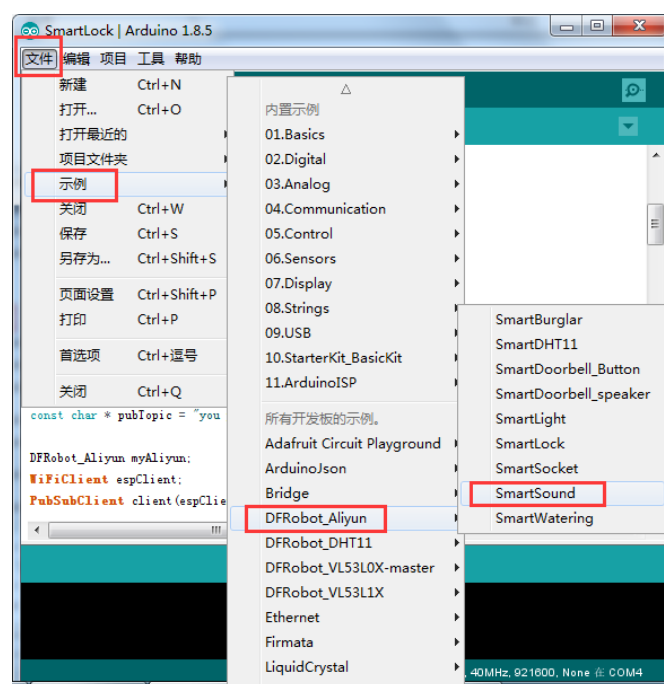
⑦添加 **API 输出** 节点。将城市 ID 输入、墨迹天气、温江温度、提示音 0/1、API 输出连接起来，再点击**部署->调试**，即可播放相应的提示音，如下图所示：



如此一个“智能提示音”的服务就搭建好了，当输入某个城市的 Id，检测到温度小于 20 摄

度时播放提示音 0，当检测到大于或等于 20 摄氏度时，播放提示音 0。

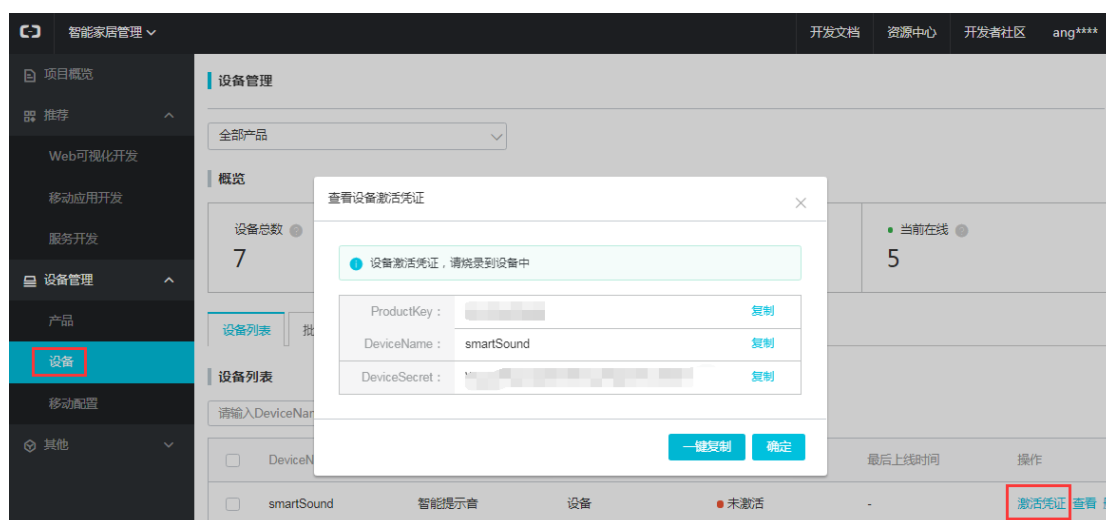
8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartSound** 项目，如下图所示：



① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示) 更换为可用的 WiFi。

```
/*配置WIFI名和密码*/  
const char * WIFI_SSID    = "WIFI_SSID";  
const char * WIFI_PASSWORD = "WIFI_PASSWORD";
```

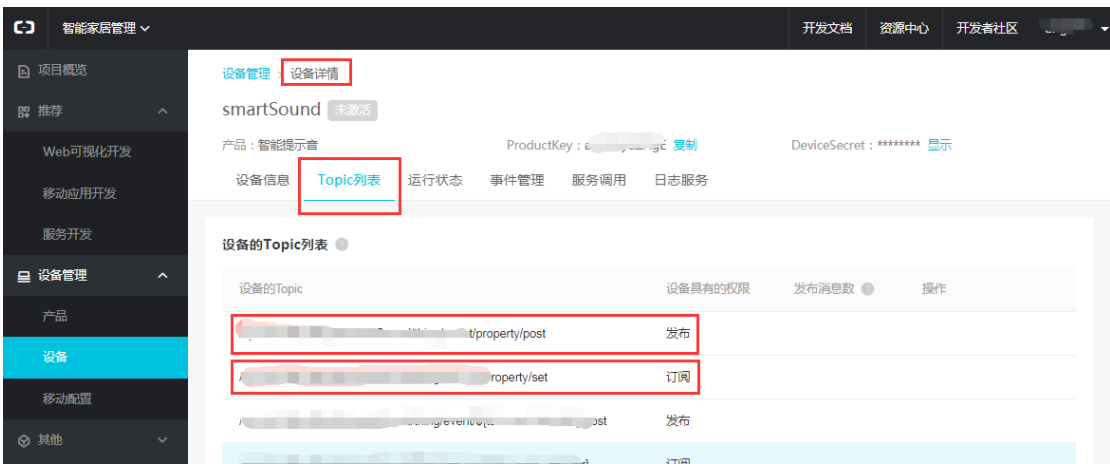
② 填写设备证书信息。获取设备 **SmartSound** 的设备信息，如下图所示：



将获取到的 **SmartSound** 的设备证书信息，复制粘贴到如下图所示的程序模块中。

```
/*配置设备证书信息*/  
String ProductKey = "you Product Key";  
String ClientId = "12345";  
String DeviceName = "you Device Name";  
String DeviceSecret = "you Device Secret";
```

③ 替换订阅、上报 Topic。在阿里云 IoT 上找到此应用创建的设备 SmartSound 的 Topic 列表，如下图所示，并将它替换到程序的 TOPIC 设置中：



```
/*需要订阅的TOPIC*/  
const char * subTopic = "you sub Topic";//****set
```

④ 替换产品标识符。在设备管理的产品项，找到名为智能提示音的产品，点击查看，点击功能定义，即可在自定义功能栏看到标识符，如下图所示，将其替换到程序中的产品标识符即可。

自定义功能 ● 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	播放提示音	PlayMusic	int32 (整数型)	取值范围：0 ~ 10	编辑 删除

```
/*需要操作的产品标识符*/  
String Identifier = "you Identifier";
```

⑤ 编译上传至 FireBettle Board-ESP32 主板。选好开发板类型和端口号后，点击编译上传即可将程序上传至 ESP32 主板。

此时设备即成功与 Aliyun 平台上的 SmartSound 设备关联，此时开始 API 调试，会下发唱那首歌的指令，如此喇叭就能根据天气唱不同的歌。

3.8 应用八：智能门铃(升级篇)

前几个应用，都是通过网页调试，或手机控制一个物理设备。那么两个物理设备，在不同的主控上，如何通过阿里云实现控制？为此，我们特别推出了智能门铃这款应用，这款应用需要你额外有块 FireBeetle Board-ESP32 主板。

应用目标：一个按钮，一个喇叭，分别接在不同的 FireBeetle Board-ESP32 主控上，要求按按钮，喇叭响，同时将信息“hello，门铃响了！”推送至钉钉。

所需元件

按钮 x1

功放喇叭 x1

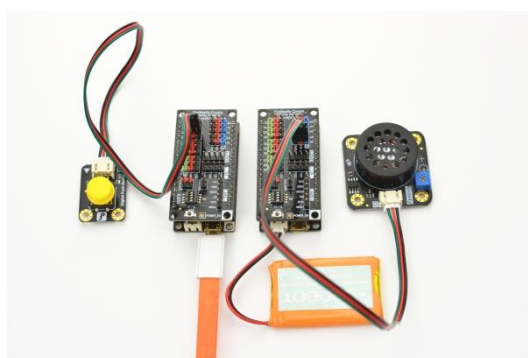
FireBeetle Board-ESP32 x2

FireBeetle Gravity 扩展板 x2

智能手机 x1

硬件连接

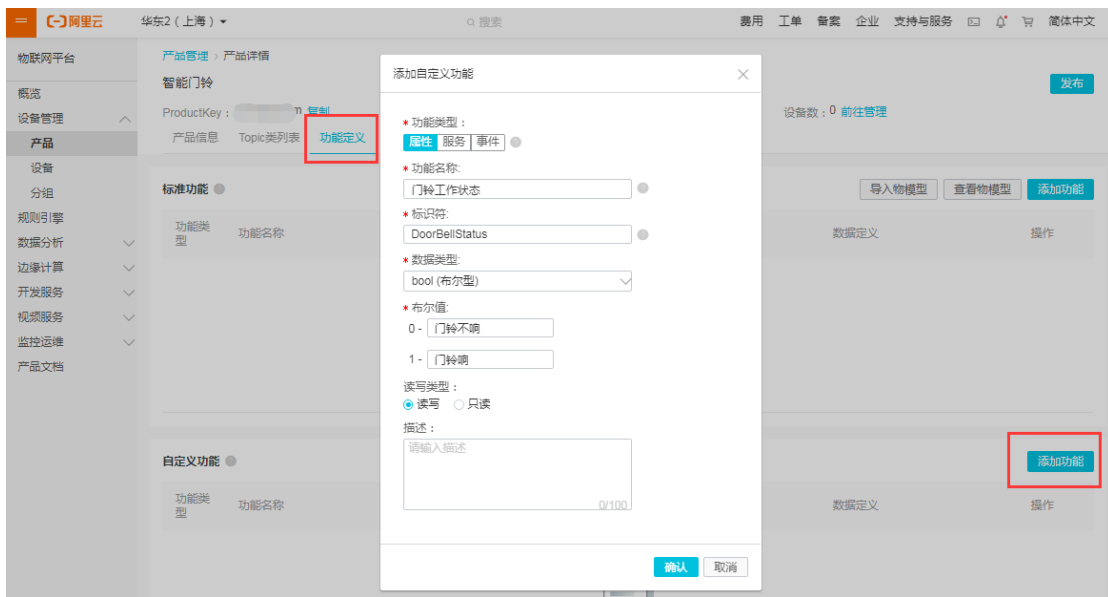
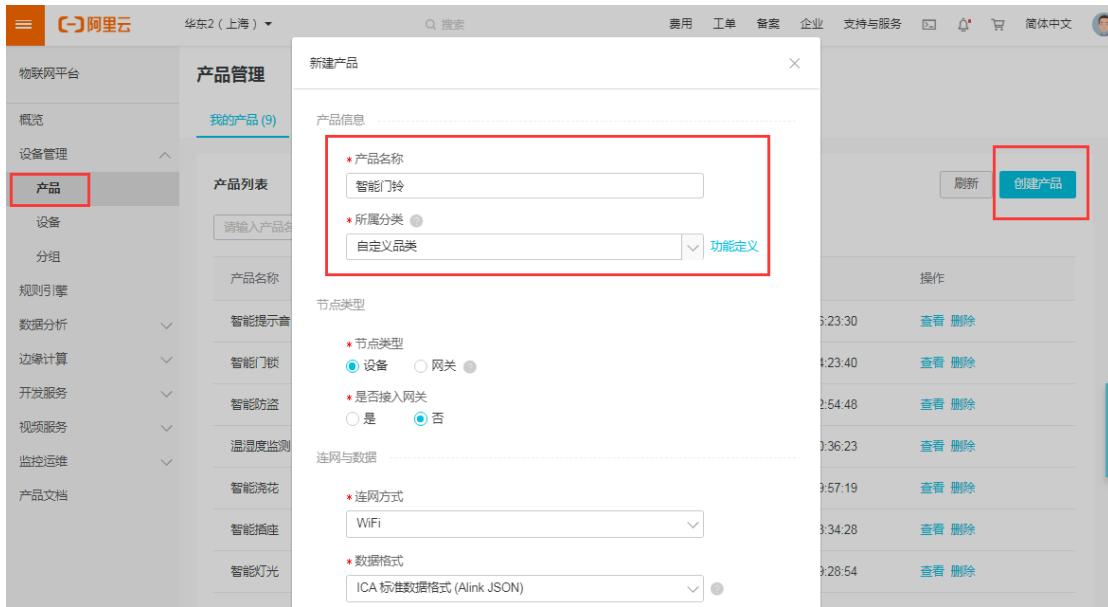
把 FireBeetle Gravity 扩展板直插在 FireBeetle Board-ESP32 主板上，再把按钮接在扩展板的 D2 数字引脚上，把功放喇叭接在另一扩展板的 A4 引脚，确保电源引脚、接地引脚和信号引脚都连接正确。



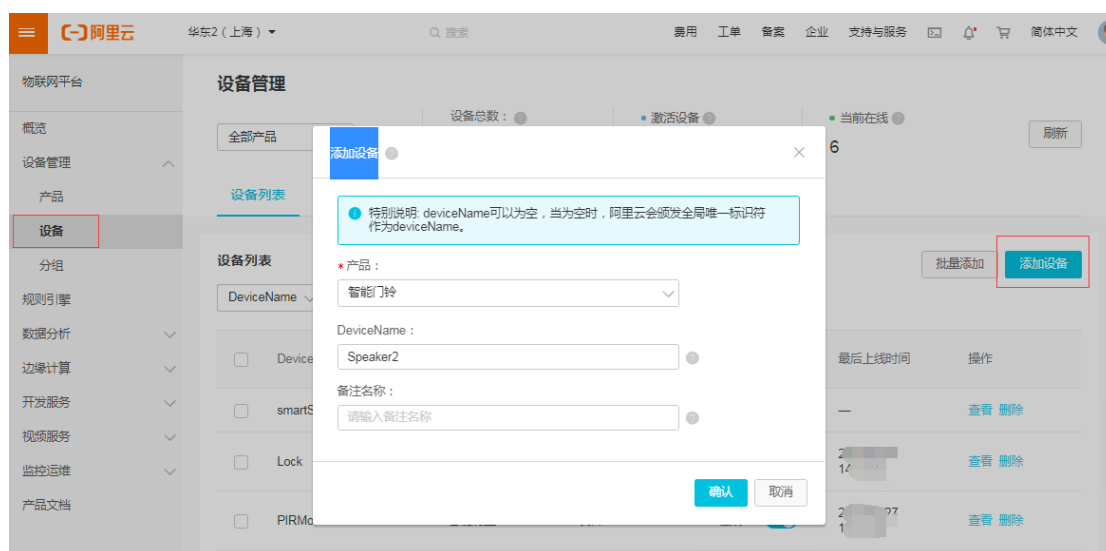
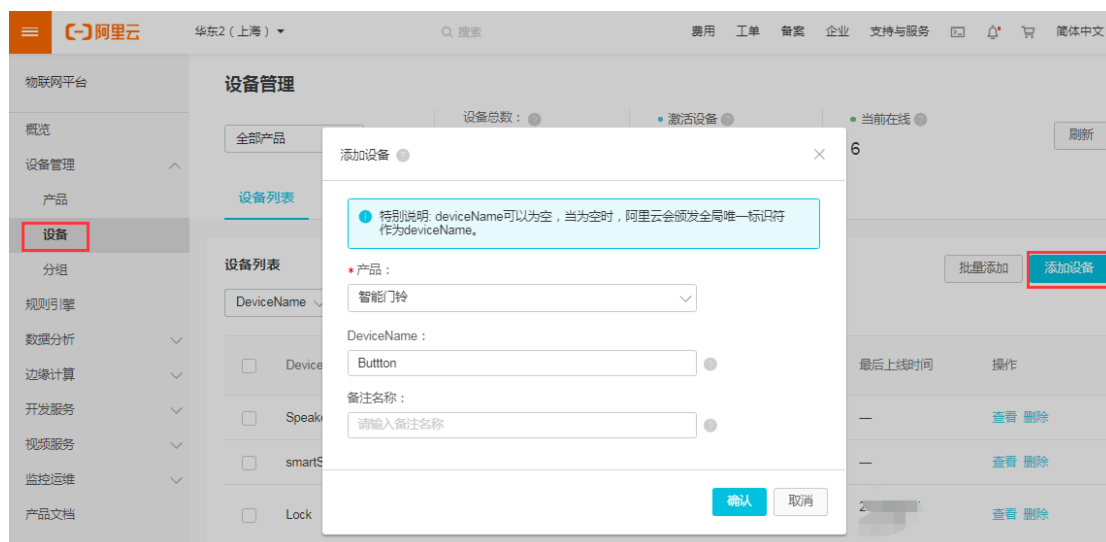
应用分析：通过前面应用的学习，智能门铃项目中有 2 个物理设备，我们需要创建 2 个设备，对于按钮来说，只需要上报属性至阿里云，门铃只需从阿里云下发指令。其次是创建一个钉钉服务来关联按钮和喇叭。

制作步骤

1. 打开浏览器，输入网址 <https://iot.aliyun.com>, 登录并进入 **物联网平台**；
2. 创建一个名为 **智能门铃** 的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



自定义功能						添加功能
功能类型	功能名称	标识符	数据类型	数据定义	操作	
属性	门铃工作状态	DoorBellStatus	bool (布尔型)	布尔值：门铃不响 - 0；门铃响 - 1；	编辑	删除
属性	按钮状态	ButtonStatus	bool (布尔型)	布尔值：松开 - 0；按下 - 1；	编辑	删除

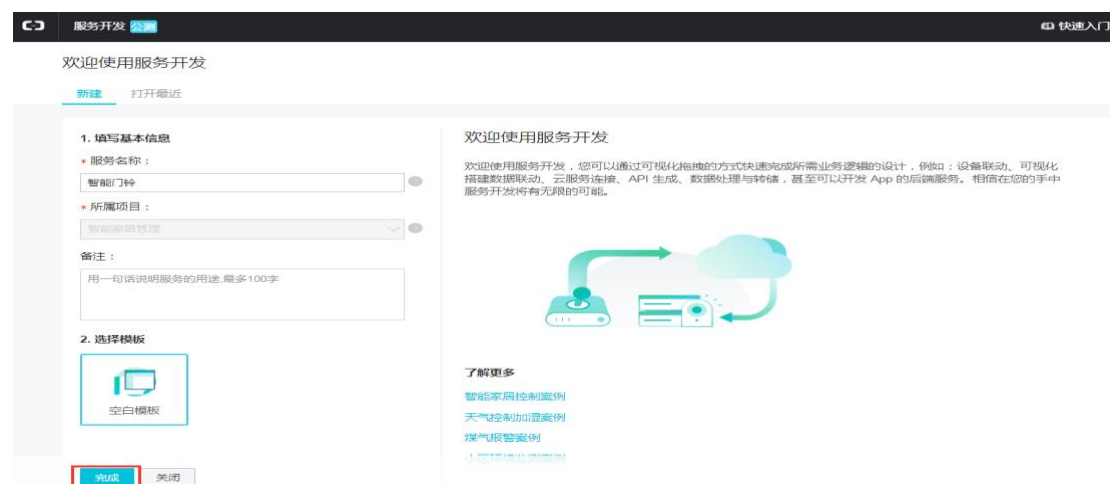


5.将**智能门铃**产品导入**智能家居管理**项目中，如下图所示：



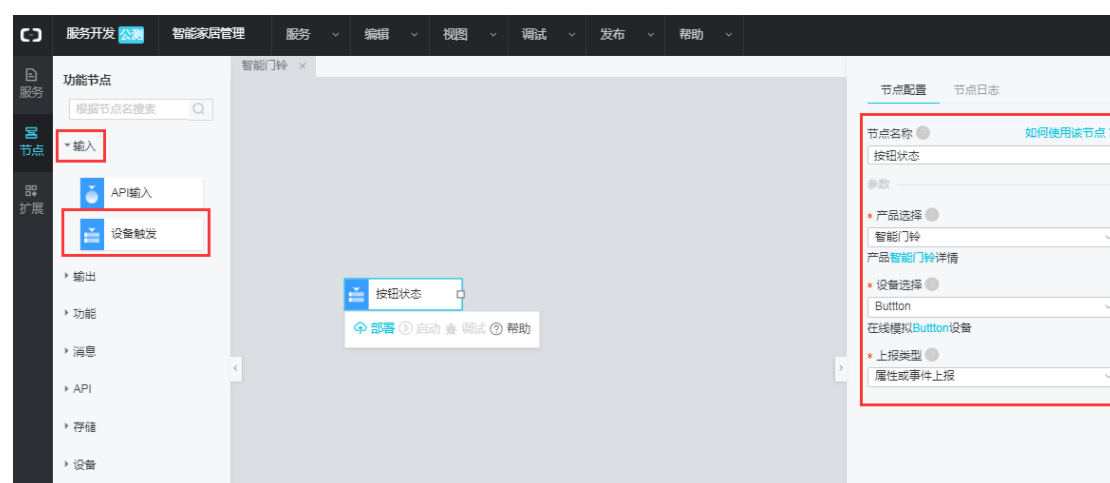
6. 选中左侧导航栏的**推荐->服务开发**,点击**新建服务**,根据页面提示,创建一个名为“**智能门铃**”的服务,该服务的主要功能为:当阿里云平台检测到按钮被按下,则下发播放门铃

指令，让喇叭播放乐曲。如下图所示：

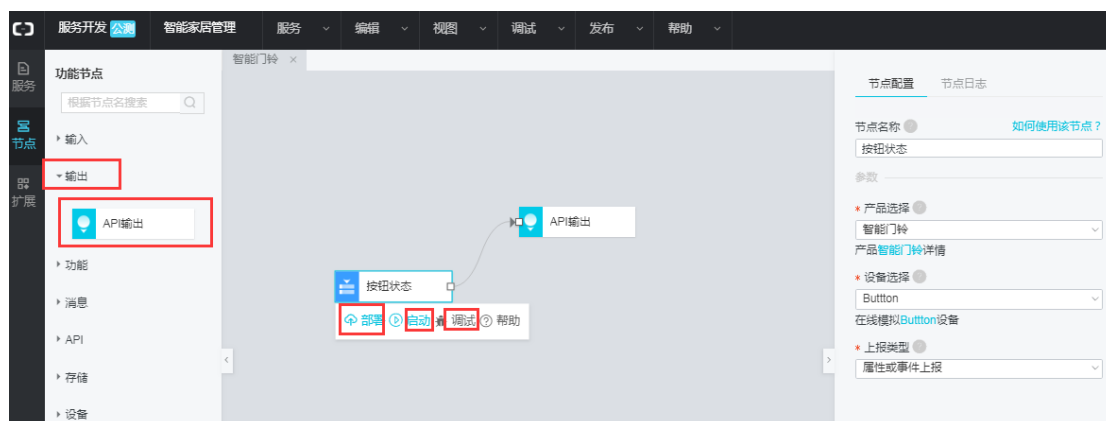


7. 点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址：
<https://linkdevelop.aliyun.com/studioservice-doc#index.html>

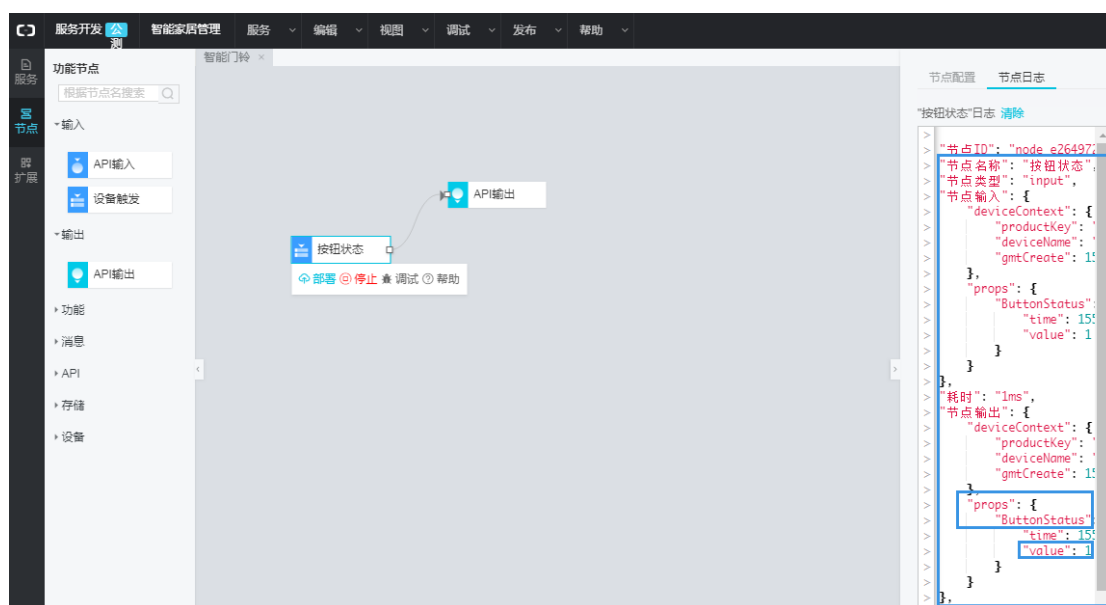
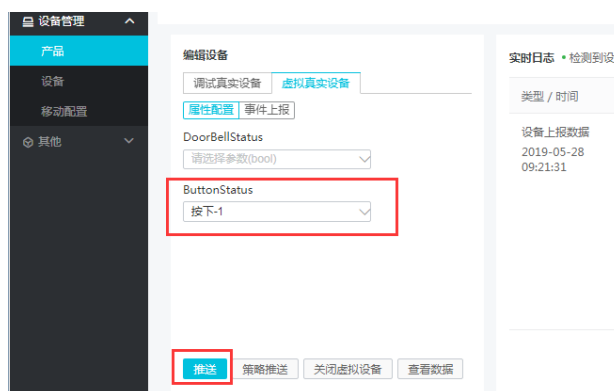
① 点击左侧导航栏的**节点**，在节点列表中选择**输入->设备触发**，并将其拖动到服务画布内，把这个设备触发节点改为**按钮状态**；右侧栏配置项分别选择“**智能门铃**”产品，选择“**Button**”设备，并选择监听“**属性或事件上报**”上报，如下图所示：



② 获取“按钮状态”节点的节点输出信息。“按钮状态”节点配置完成后，选中左侧节点列表的**输出->API 输出**，点击**部署->启动->调试**，如下图所示：



在弹出的“在线调试”页面中选择某个参数，点击**推送**，回到服务开发页面，点击节点日志，即可看到“PIR 传感器”节点的输入输出节点信息，如图：



③添加一个**条件**节点，条件节点相当于一个 if-else 判断。选中左侧节点列表的功能->条件节点，将其改为 **“按下按钮”**，并根据上面得到的 **“按钮状态”** 输出节点信息配置 **“按下按钮”** 条件节点的输入设置，将其配置为 **“上一个节点”** + **“props.ButtonStatus.value”**；将路径 1 的第二个选择框设置为 **“数字”**，第三个框输入 **1**（表示按钮被按下），可以参考**如何使用该节点？->如何配置条件节点？**



④删除“API 输出”节点。选中**设备->智能门铃**，添加**智能门铃**设备节点，配置选择“**Speaker2**”设备和**设备指令下发**，参数填写智能门铃产品的自定义功能：门铃工作状态定义属性“**DoorBellStatus**”: 1。并将按钮状态、按下按钮、智能门铃连接起来



如此，当点击**部署->启动**后，当检测到按钮状态被按下，则会**触发门铃响**动作。最后我们还需要将门铃响的信息推送至钉钉。

⑤选中**消息->钉钉机器人**节点并配置。若不知道如何配置钉钉机器人，可参考[如何使用该节点？->如何配置钉钉机器人节点？](#)

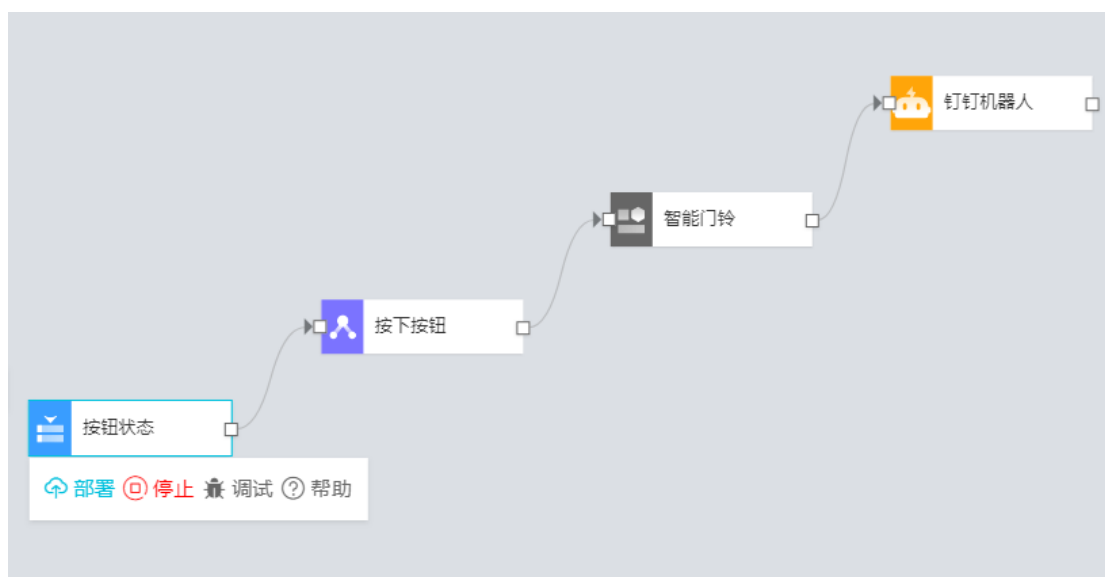
a.获取 **Webhook**。在获取之前，你需要在钉钉群中，添加一个钉钉机器人(注意查看钉钉机器人的 **Webhook** 时只能用电脑端的钉钉)，具体添加方法请参考[如何使用该节点？->如何配置钉钉机器人节点？](#)最终获取的 **Webhook**，复制粘贴至右侧栏的 **Webhook** 处，如下图所示：



b.选择“text”作为消息类型，并在 **content** 的双引号内填入信息“hello,门铃响了!”。

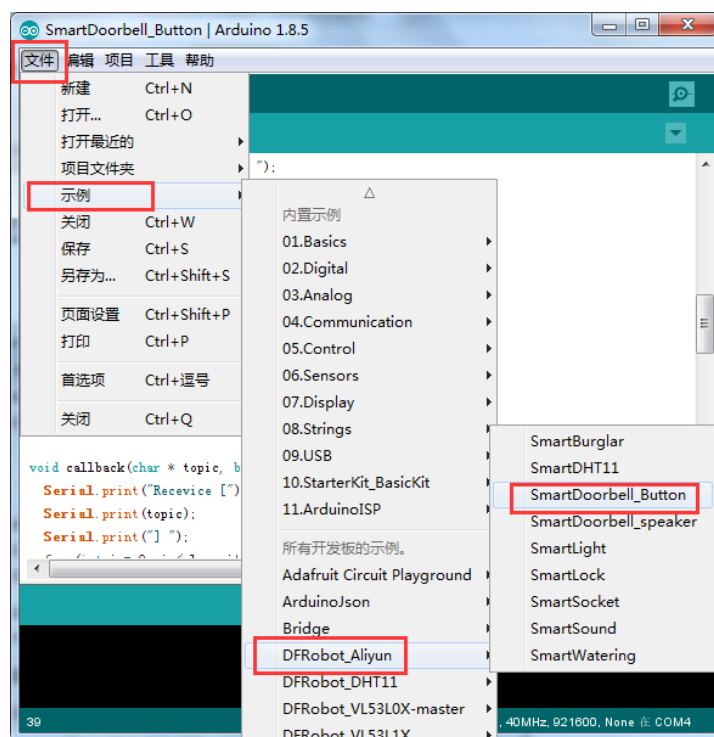


⑥点击设备触发节点的**部署**按钮，点击**启动**，如下图所示：



如此，一个名为“智能门铃”的服务就开发完毕了。

8.Arduino 程序修改。点击文件->示例->DFRobot_Aliyun->SmartDoorBell_Button 项目，如下图所示：



① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示) 更换为可用的 WiFi。

```
/*配置WIFI名和密码*/  
const char * WIFI_SSID      = "WIFI_SSID";  
const char * WIFI_PASSWORD = "WIFI_PASSWORD";
```

② 填写设备证书信息。获取设备 **Button** 的设备信息，如下图所示：



将获取到的 **Button** 的设备证书信息，复制粘贴到如下图所示的程序模块中。

```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

③ 替换上报 Topic。在阿里云 IoT 上找到此应用创建的设备 **Button** 的 **Topic** 列表，如下图所示，并将它替换到程序的 TOPIC 设置中：



```
/*需要上报的TOPIC*/
const char * pubTopic = "you pub Topic";//*****post
```

④ 替换产品标识符。在设备管理的产品项，找到名为**智能门铃**的产品，点击**查看**，点击**功能定义**，即可在**自定义功能栏**看到**标识符**，如下图所示，将其替换到程序中的产品标识符即可。

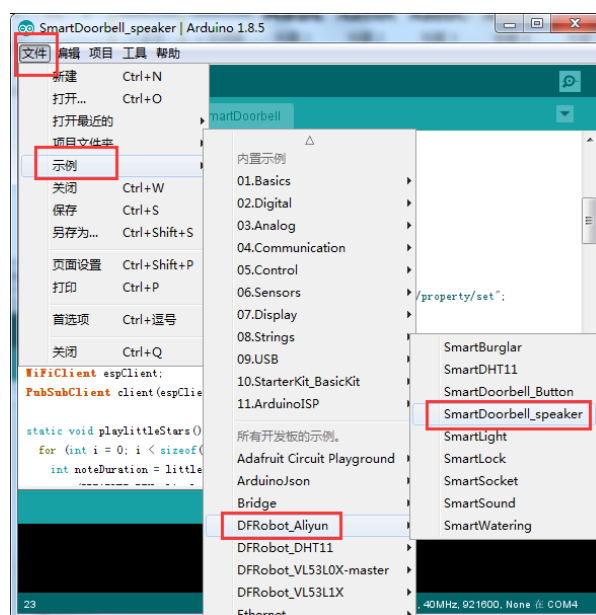


```
/*需要操作的产品标识符*/  
String Identifier = "you Identifier";
```

⑤编译上传至 FireBettle Board-ESP32 主板。选好开发板类型和端口号后，点击编译上传即可将程序上传至 ESP32 主板。

此时按钮设备成功与 Aliyun 平台上的 Button 设备关联。

9 Arduino 程序修改，点击文件->示例->DFRobot_Aliyun->SmartDoorBell_Speaker 目，如下图所示：



① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示) 更换为可用的 WiFi。

```
/*配置WIFI名和密码*/  
const char * WIFI_SSID      = "WIFI_SSID";  
const char * WIFI_PASSWORD = "WIFI_PASSWORD";
```

② 填写设备证书信息。获取设备 **Speaker2** 的设备信息，如下图所示：



将获取到的 **Speaker2** 的设备证书信息，复制粘贴到如下图所示的程序模块中。

```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

③ 替换订阅 Topic。在阿里云 IoT 上找到此应用创建的设备 **Speaker2** 的 **Topic 列表**，如下图所示，并将它替换到程序的 TOPIC 设置中：



```
/*需要订阅的TOPIC*/
const char * subTopic = "you sub Topic";//****set
```

④ 替换产品标识符。在设备管理的产品项，找到名为 **智能门铃** 的产品，点击 **查看**，点击 **功能定义**，即可在 **自定义功能** 栏看到 **门铃工作状态的标识符**，如下图所示，将其替换到程序中的产品标识符即可。



```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

⑤ 编译上传至 FireBettle Board-ESP32 主板。选好开发板类型和端口号后，点击编译上传即可将程序上传至 ESP32 主板。

此时喇叭设备成功与 Aliyun 平台上的 Speaker 设备关联。

按下 2 块 ESP32 的复位键，一段时间后，按按钮，门铃（喇叭）响了，钉钉接收到群机器人发来的消息：“hello，门铃响了”，如下图所示：

