Gravity：Starter Kit for Aliyun IoT

阿里云 IoT 入门套件

版本号：V2.0

目录

第一章 Arduino 与 ESP32 3
1.1 Arduino 平台简介 3
1.2 FireBeetle Board-ESP32 简介 4
1.3 Arduino IDE for FireBeetle Board-ESP32 4
STEP1：下载 Arduino IDE 软件 5
STEP2：Arduino IDE 汉化 6
STEP3：安装 FireBeetle Board-ESP32 开发板核心 7
STEP4：连接 FireBeetle Board-ESP32 至电脑 9
STEP5：在 Arduino IDE 中进行编程 12
STEP6：上传代码至 FireBeetle Board-ESP32 主板 13
1.4 认识阿里云 IoT 入门套件 17
第二章 阿里云 IoT 平台 18
2.1 认识阿里云 IoT 18
2.2 注册并登陆阿里云账号 18
2.3 实名认证 19
2.4 开通物联网服务 22
2.5 登陆物联网平台控制台 23
2.6 使用物联网平台 24
STEP1 创建产品 24
STEP2 创建设备 26
STEP3 物联网开发服务 29
第三章 IOT 应用 42
准备工作 42

3.1 应用一：智能灯光 44
所需元件 45
硬件连接 45
制作步骤 45
3.2 应用二：智能插座 58
所需元件 58
硬件连接 58
制作步骤 59
3.3 应用三：温湿度监测 67
所需元件 67
硬件连接 68
制作步骤 68
3.4 应用四：智能浇花 77
所需元件 78
硬件连接 78
制作步骤 78
3.5 应用五：智能防盗 87
所需元件 87
硬件连接 88
制作步骤 88
3.6 应用六：智能门锁 101
所需元件 102
硬件连接 102
制作步骤 103
3.7 应用七：智能提示音 114
所需元件 114
硬件连接 114
制作步骤 115
3.8 应用八：智能门铃(升级篇) 132
所需元件 133
硬件连接 133
制作步骤 133

第一章 Arduino 与 ESP32

1.1 Arduino 平台简介

维基百科描述：“Arduino 是一款单板的微控制器和一整套的开发软件，它的硬件包含一个以 Atmel AVR 单片机为核心的开发板和其他各种 I/O 板。软件包含一个标准编程语言开发环境和在开发板上运行的烧录程序。”

简而言之，Arduino 是一款使用简单，集硬件，软件环境于一身的开发平台，旨在为智能硬件爱好者，交互艺术设计师以及电子软件工程师，提供简单易用的开发体验。它是一个交互式系统，可以通过硬件和软件与它的环境进行互动。例如，一个简单的 Arduino 应用项目是在按下一个按钮时，点亮一盏小灯并保持一段时间，如 30 秒。这个例子中必须有一盏小灯和一个按钮与 Arduino 相连。Arduino 可以一直处于等待按钮被按下的状态，一旦按钮被按下，Arduino 就点亮那盏小灯并开始计时，当它计时到 30 秒时，Arduino 熄灭小灯并且等待下一次按钮被按下。你可以用这个小项目做一个台灯。

当然，你也可以把上面的项目进行扩展，比如，将按钮换成红外探头，用手在探头上挥动一下，探头检测到手势动作并给出一个信号到 Arduino，以此来代替按钮功能；再比如，用微波传感器代替按钮，当周围有物体运动时 (如人经过) 传感器给出一个信号到 Arduino，Arduino 控制小灯亮 30 秒。这样，你可以将这个小项目升级成智能过道开关，代替传统的声控开关，是不是很有意思?

Arduino 可以通过各种各样传感器来检测周围环境，并通过电机，LED 灯以及其他发生器来影响周围环镜。板上微控制器编程使用 Arduino 编程语言 (基于 Wiring) 和 Arduino

开发环境（以 Processing 为基础）Arduino 可以独立运行，也可以与计算机上运行的软件（例如，Flash，Processing，MaxMSP）进行通信 Arduino 开发 IDE 接口基于开放源代码，可以让您免费下载使用，开发出更多令人惊艳的互动作品。

Arduino 是人们连接各种任务的粘合剂。我们可以用它来做许多有趣的项目。

1.2 FireBeetle Board-ESP32 简介

FireBeetle Board-ESP32 是 DFRobot FireBeetle 萤火虫系列主控之一，是一款专为物联网设计的低功耗微控制器。它采用乐鑫的 ESP32 芯片，集 WiFi&蓝牙，MCU 于一体，设计上采用超低功耗外围硬件设计，支持 USB 及锂电池供电，支持板载锂电池充电，编程方式完全兼容 Arduino IDE 编程等功能，帮助你快速搭建物联网平台，省去了外围硬件的搭建。默认配置及对应硬件 IO 如下图 1 所示:

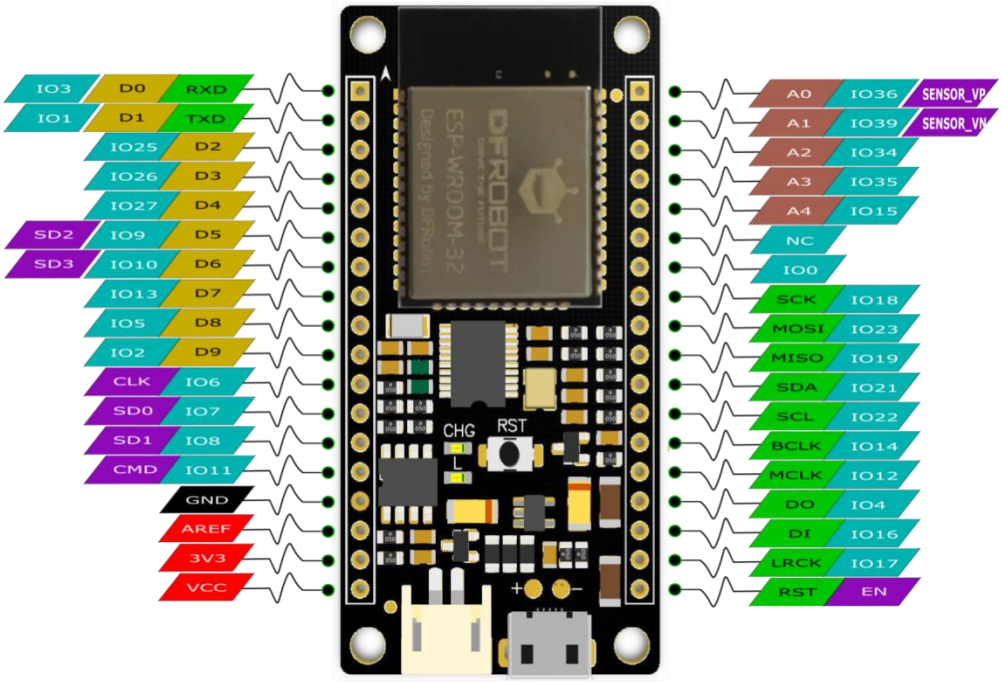


图 1 FireBeetle Board-ESP32 默认配置及硬件 IO 示意图

1.3 Arduino IDE for FireBeetle Board-ESP32

如果你是第一次接触 Arduino 开发平台，同样可以通过以下教程学习如何下载安装

Arduino IDE 软件，以及如何为 FireBettle Board-ESP32 主板编写程序。

开始之前，请确认你手边有一套 DF 的阿里云 IoT 入门套件，如图 2 所示，除此之外，你还需要一台运行 windows/Mac OS/Linux 操作系统并且有网络连接的电脑。

图 2 阿里云 IoT 入门套件

我们需要套件中的 FireBeetle Board-ESP32 主板及 USB 连接线，如图 3 所示：

图 3 USB 连接线&FireBeetle Board-ESP32 主控

Arduino IDE for FireBeetle Board-ESP32 就是让你的 Arduino IDE 支持 FireBeetle Board-ESP32 控制器，通过以下步骤，可以快速搭建开发环境。

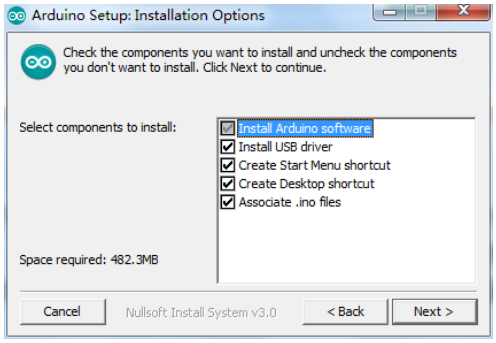
STEP1：下载 Arduino IDE 软件

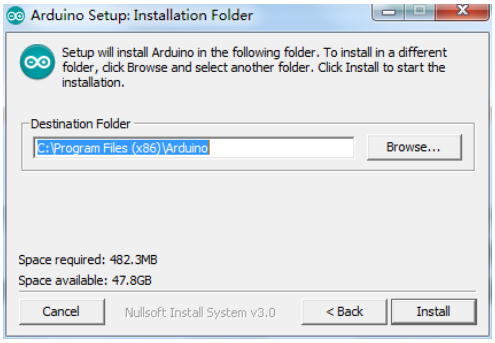
以下的步骤是基于 Windows 操作系统，如果你使用的是其他操作系统，可以将其作为参考。

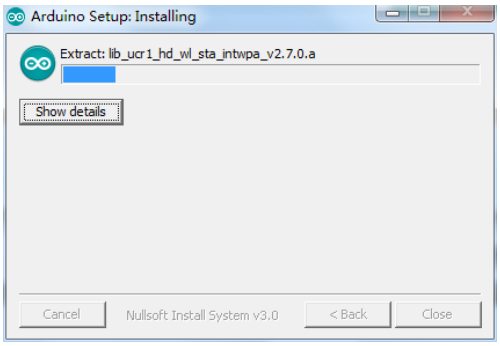
首先，你需要从官网下载最新版本的 Arduino IDE 软件。下载链接：https://www.arduino.cc/en/software

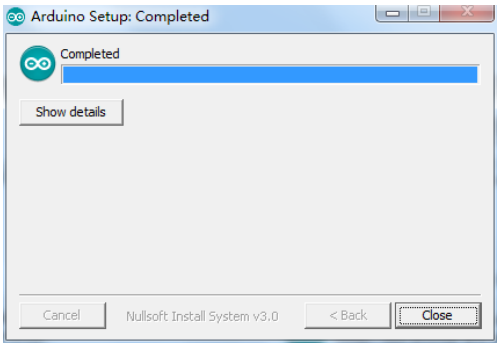
注意：FireBeetle Board-ESP32 主板建议使用 1.8.0 以上版本。

在下载页右侧的列表中选择下载对应的安装包。对于 Windows 系统用户既可以选择下载 **Windows installer**（推荐初次使用者下载），也可以下载 Windows ZIP 安装包（需要手动安装驱动）。若选择的是 Windows installer，你可以直接执行安装程序，并跟随安装向导完成配置，驱动会在程序安装完成后自动安装。





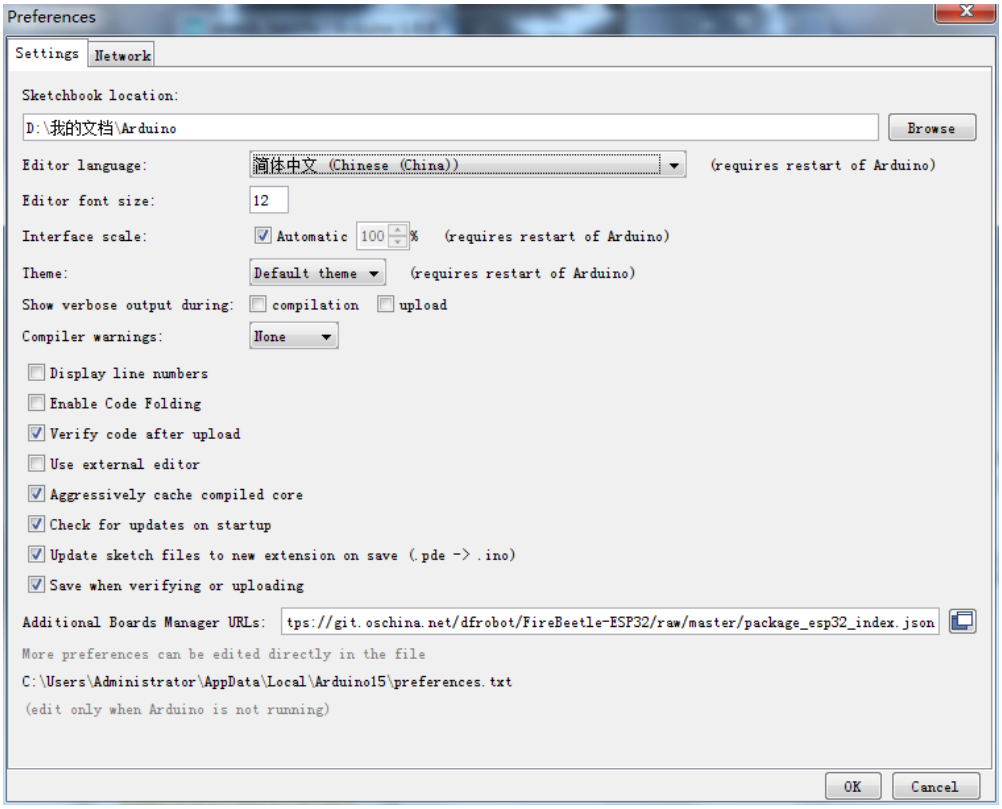




STEP2：Arduino IDE 汉化

Arduino IDE 本身支持多种语言（包括中文），我们只需设置为中文即可。

打开 **Fire->Preferences->Editor language**,选择**简体中文（Chinese（China））**，然后重启 IDE。



STEP3: 安装 FireBeetle Board-ESP32 开发板核心

Arduino IDE 安装包中不包含 FireBeetle Board-ESP32 开发板核心，需要手动添加。首先，要添加 FireBeetle Board-ESP32 支持，需要在 Arduino 开发板管理器里手动安装 FireBeetle Board-ESP32 开发板核心。

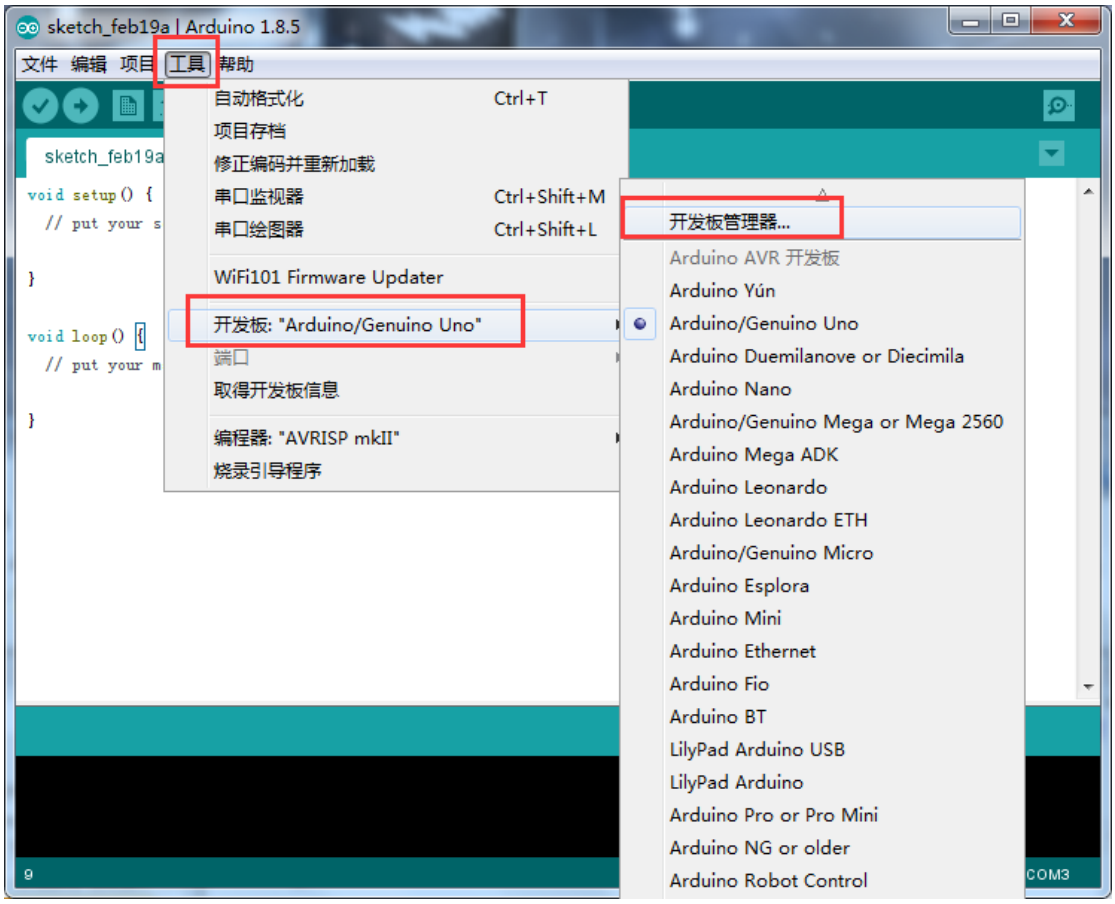
（1）打开**文件->首选项**，在**附加开发板管理器网址**中，将以下网址复制进去：

http://download.dfrobot.top/FireBeetle/package_esp32_index.json

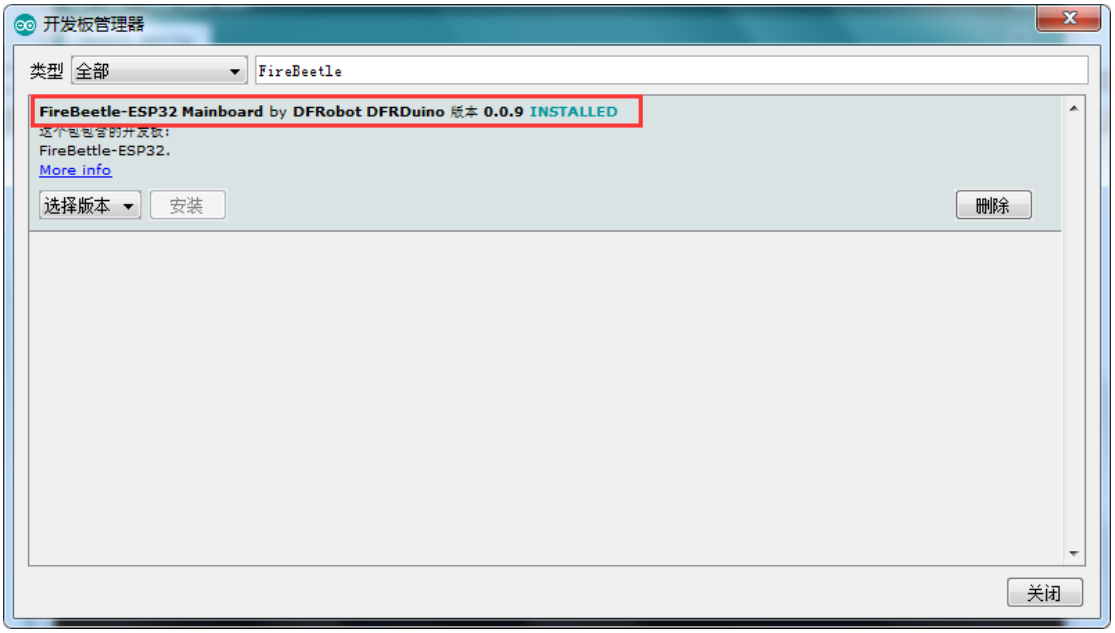
注：上述网址有可能会失效，导致板卡下载失败，若失败你可以前往 DF 官方 wiki 教程查看 FireBeetle ESP32-Board 的使用教程。

点击**好**，完成设置。

（2）打开**工具->板子->开发板管理器**



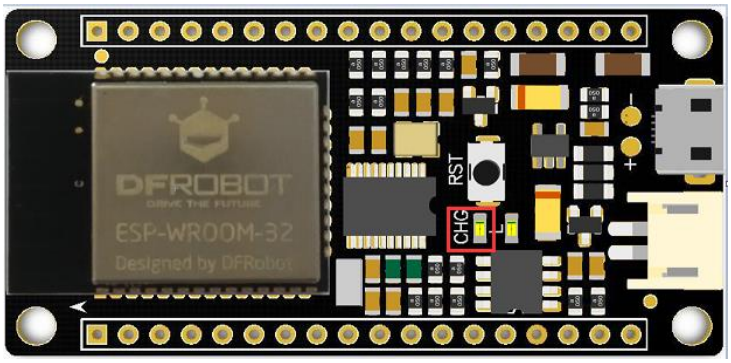
在打开的开发板管理器中，输入“FireBeetle”并等待信息加载完毕。版本号为 0.1.1（或者最新版本并点击“**安装**”后耐心等待安装完成。整个过程会因网络状况持续 5-10 分钟。安装完成后，开发板信息会被标注“INSTALLED”），如下图所示：



注意：如果发现安装过程中一直处于卡顿现象，可能是网络原因引起的，您可以尝试强制重启 Arduino IDE，重新执行之前的步骤，或者通过翻墙软件加速网络，直到安装完成。其次，在安装过程中，有些关键程序可能会被防火墙或者杀毒软件拦截，请选择允许更改并添加至白名单。

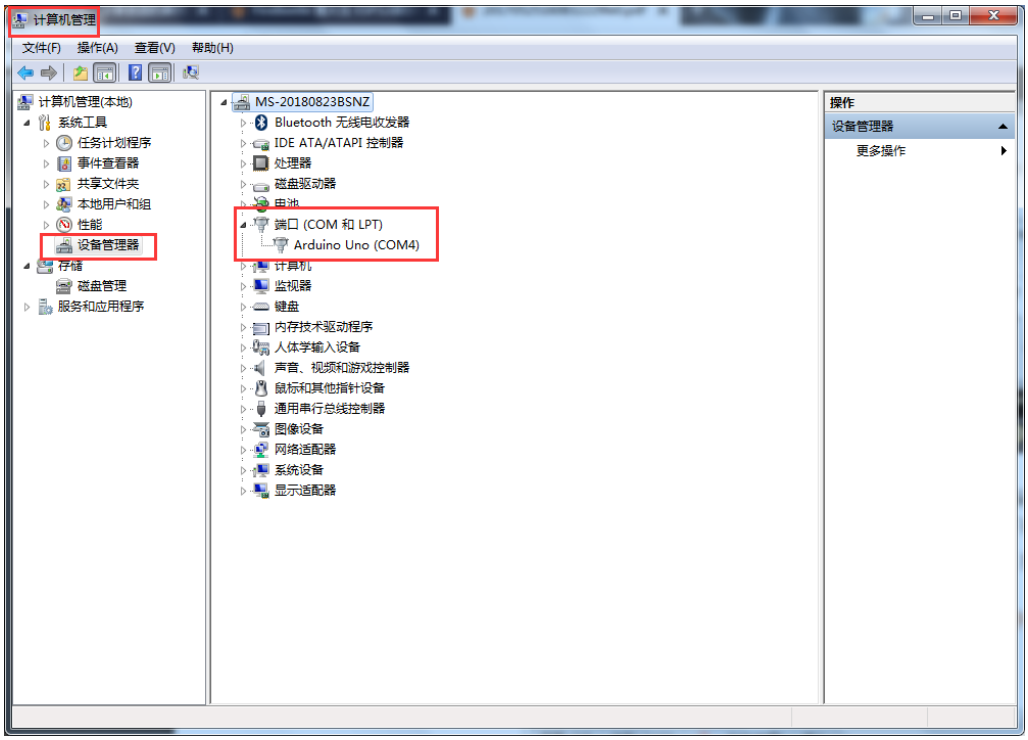
STEP4：连接 FireBeetle Board-ESP32 至电脑

正确安装完成 Arduino IDE 和 FireBeetle Board-ESP32 开发板核心后，即可将 FireBeetle Board-ESP32 通过 USB 数据线连接至电脑。正确连接时 FireBeetle Board-ESP32 的 CHG 电源指示灯会闪烁（这是在查询有没有接入锂电池）。



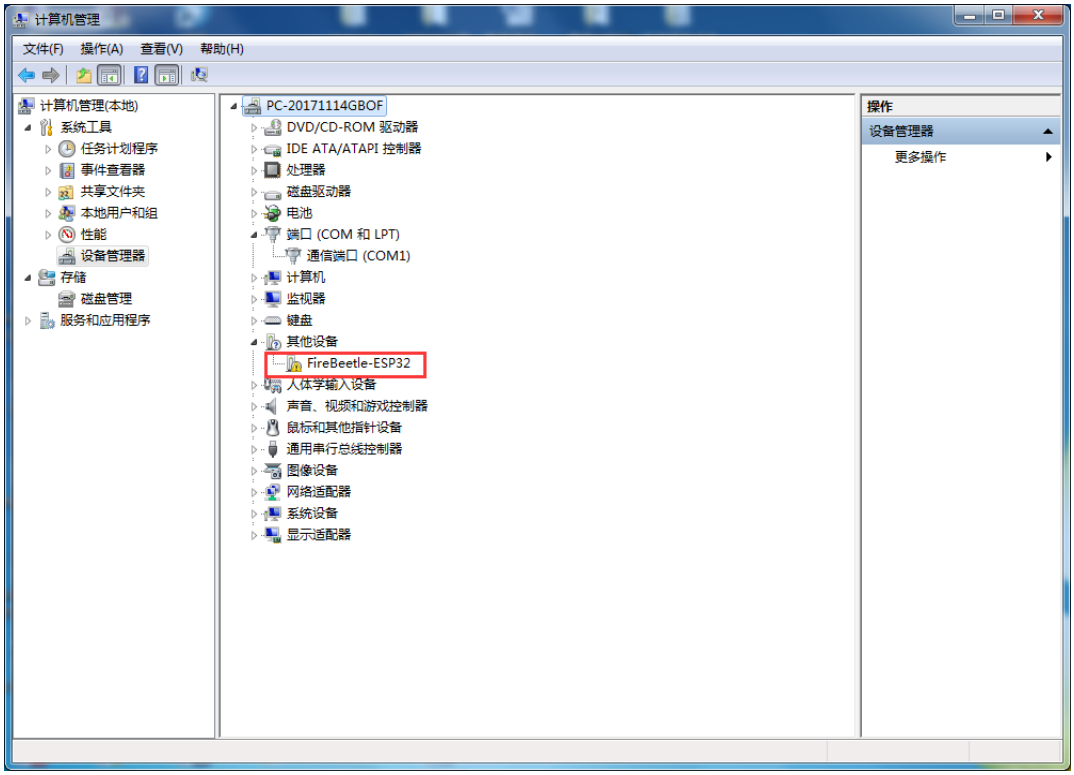
在编程之前，我们需要确保开发板被电脑识别，并找出连接了哪一个 COM 口（用于提供串口通信交互）。可以在接下来的步骤中确认。

首先打开“计算机管理”，打开“设备管理器”，点开“端口（COM 和 LPT）”。接上 FireBeetle Board-ESP32 的端口就会在列表中显示（这里是 COM4）。

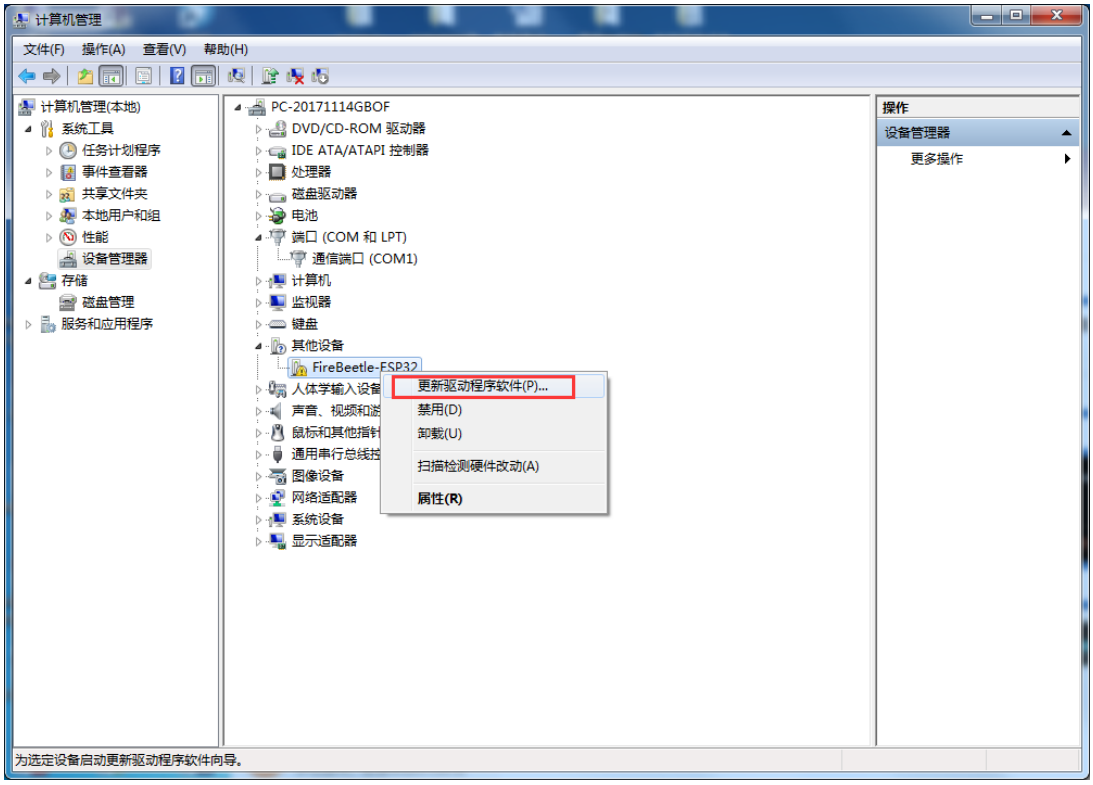


如果提示无法识别的设备，你需要下载 FireBeetle Board-ESP32 驱动到本机，并安装驱动，下载地址：https://github.com/Chocho2017/FireBeetle-Board-ESP32.git

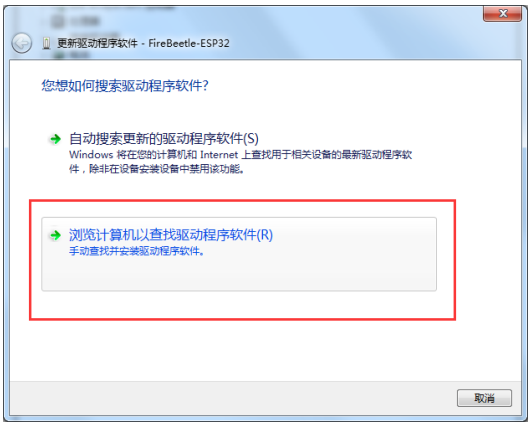
错误提示信息如下图所示：

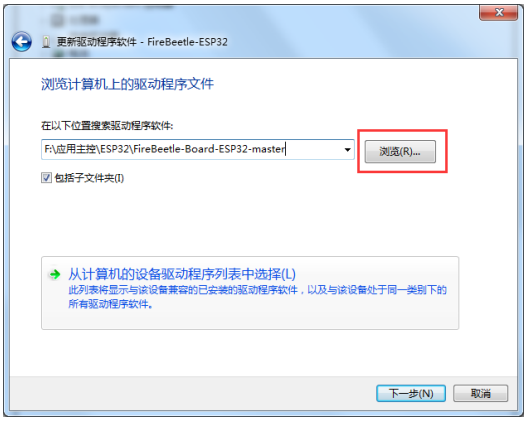


将下载后的 FireBeetle-ESP32.inf 驱动文件保存到你的电脑（任意位置都可），右键点击 **FireBeetle-ESP32**，选择**更新驱动程序软件**，如下图所示：



选择浏览计算机以查找驱动程序软件，在地址栏中输入你刚刚保存的 FireBeetle-ESP32.inf 文件目录，点击下一步，如下图所示：

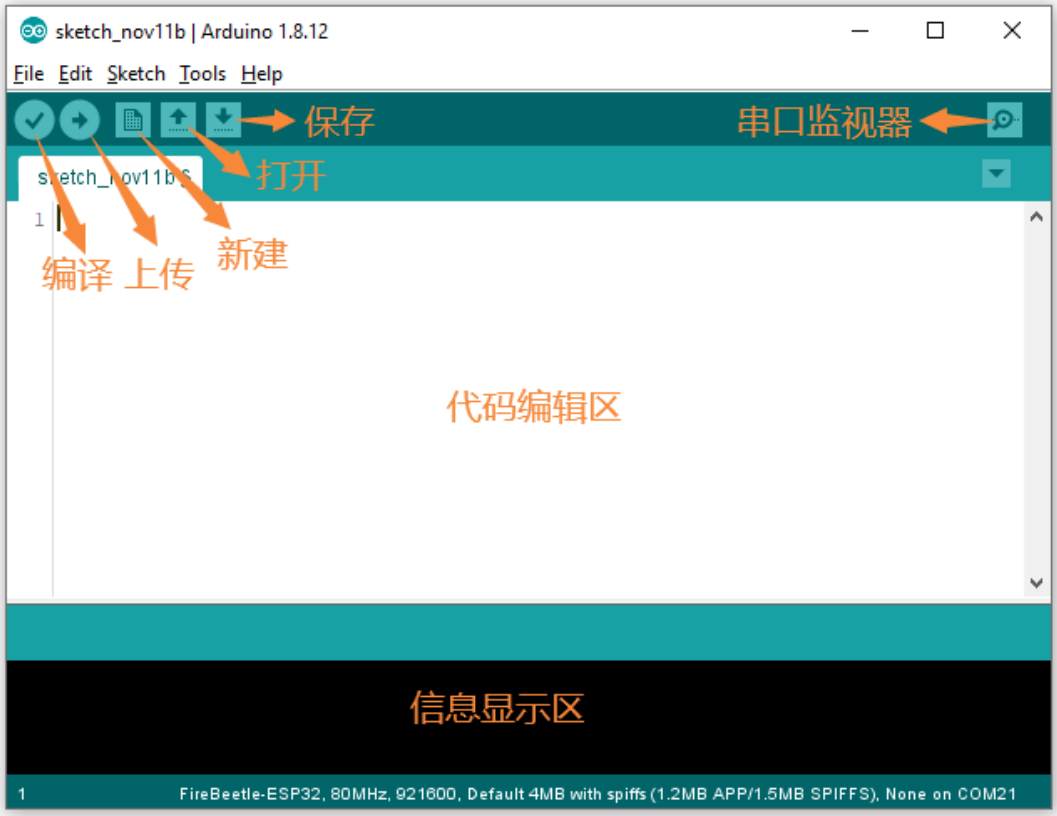




然后根据提示完成驱动文件的安装。

STEP5：在 Arduino IDE 中进行编程

Arduino IDE 软件安装完成后，运行软件打开编程窗口。你可以在这个窗口里编辑并上传代码到 Arduino 开发板上，或是使用内置的串口监视器与开发板通信。Arduino IDE 界面如下：



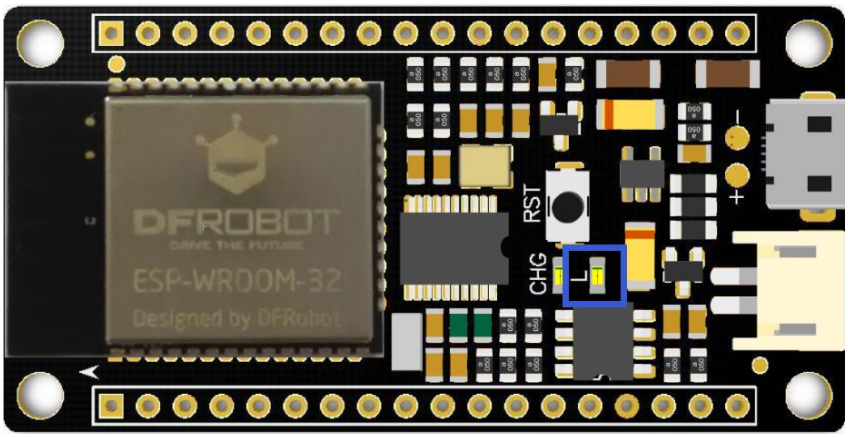
与常规 C 语言程序不同的是，一段用于 Arduino 的程序通常由 void setup()部分与 void loop()两部分构成。

“void setup()”用于放置初始化程序的代码，这部分代码在开发板上电后仅运行一次。需重复运行的代码需放置在”void loop()”中，这些代码会一直重复运行，使得开发板与外部进行交互。

STEP6：上传代码至 FireBeetle Board-ESP32 主板

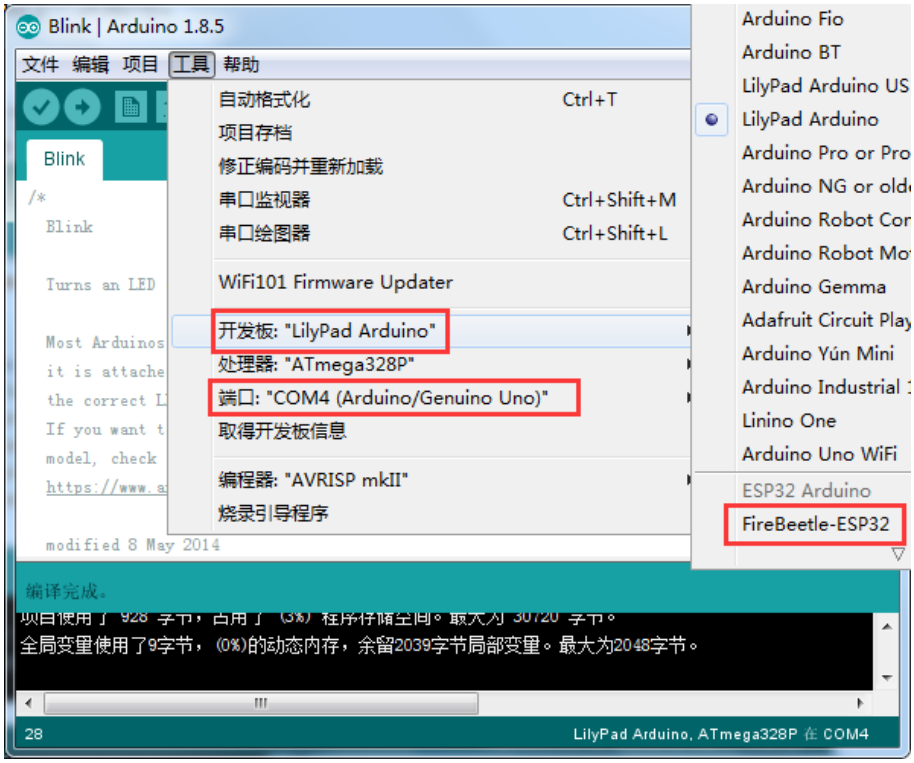
在这一步，我们将演示如何上传事例程序”Blink”到 FireBeetle Board-ESP32 主

板。”Blink”程序的功能是控制 D9 引脚上的 LED 灯间隔 1s 闪烁一次。FireBeetle Board-ESP32 主板与大部分 Arduino 相同，有一个板载的 D9 LED 信号灯，这意味着在本例中我们不需要连接其他的外设原件。LED 状态指示灯可在 FireBeetle Board-ESP32 主板上找到，如下图所示：

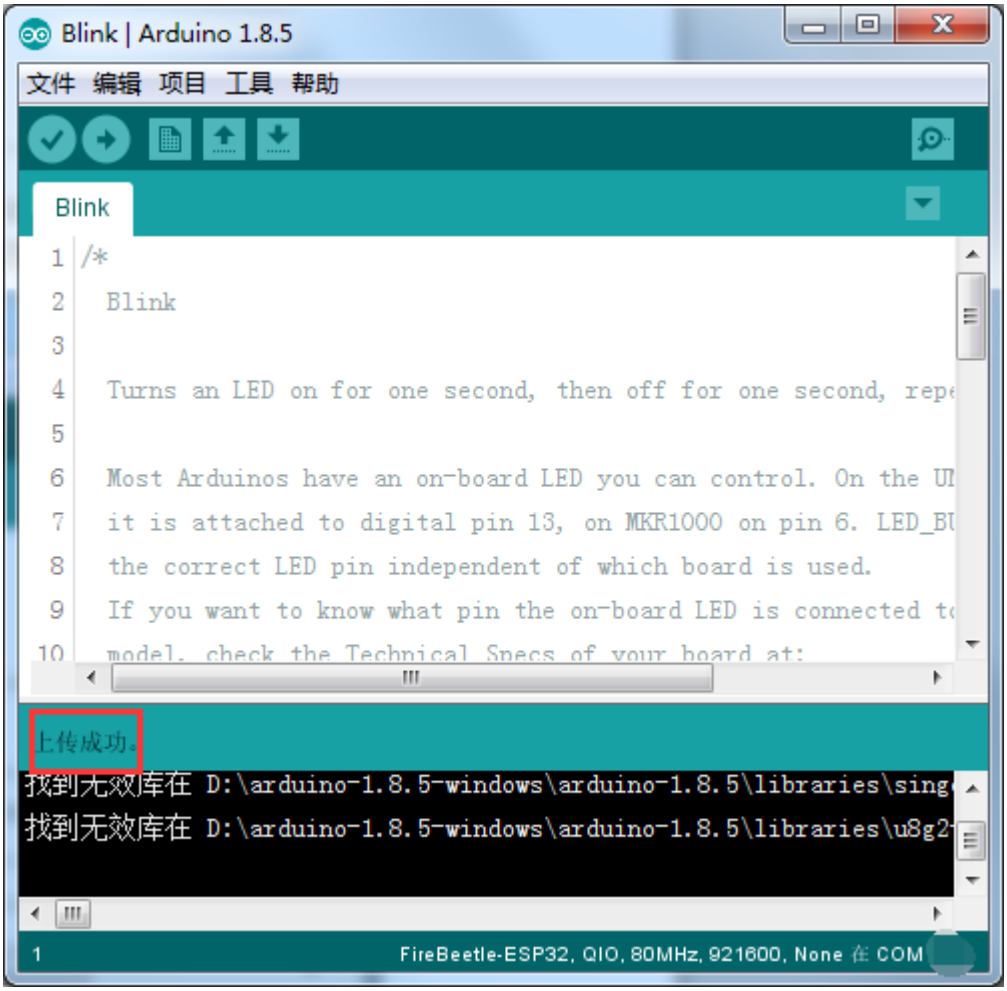


打开文件->事例->Basics->Blink，在上传之前，你应该首先确认代码中没有错误。点击“编译”确认。等待几秒，若没有错误，则会在信息窗口显示“编译完成”，表示编译成功。若出现错误可以返回检查程序是否完整。

编译成功后，选择工具->开发板->FireBeetle-ESP32，并根据 STEP4 中显示的 FireBeetle Board-ESP32 所占用的端口号，我们应该选择“COM4”作为通信端口。COM 和开发板选择完成后，开发板的信息和端口号就会在窗口右下角显示。



最后，点击“上传”烧写代码到 FireBeetle Board-ESP32 中。成功上传后，“上传成功”消息会出现在信息窗口。此时 FireBeetle Board-ESP32 板载的 D9 LED 灯会开始闪烁。



简而言之，为 Arduino 上传代码可以分为以下三个步骤：

· 编译代码；

· 选择开发板型号和端口号；

· 上传！

以上就是使用 FireBeetle Board-ESP32 主板在 Arduino IDE 上面的一些基本的方法。如果在使用中存在任何疑问或者建议，欢迎访问我们的论坛联系我们。

论坛链接：http://www.dfrobot.com.cn/community/forum.php

DF 官方链接：http://www.dfrobot.com.cn

1.4认识阿里云 IoT 入门套件

下面让我们了解一下阿里云 IOT 套件都有哪些模组，如图 4 所示：

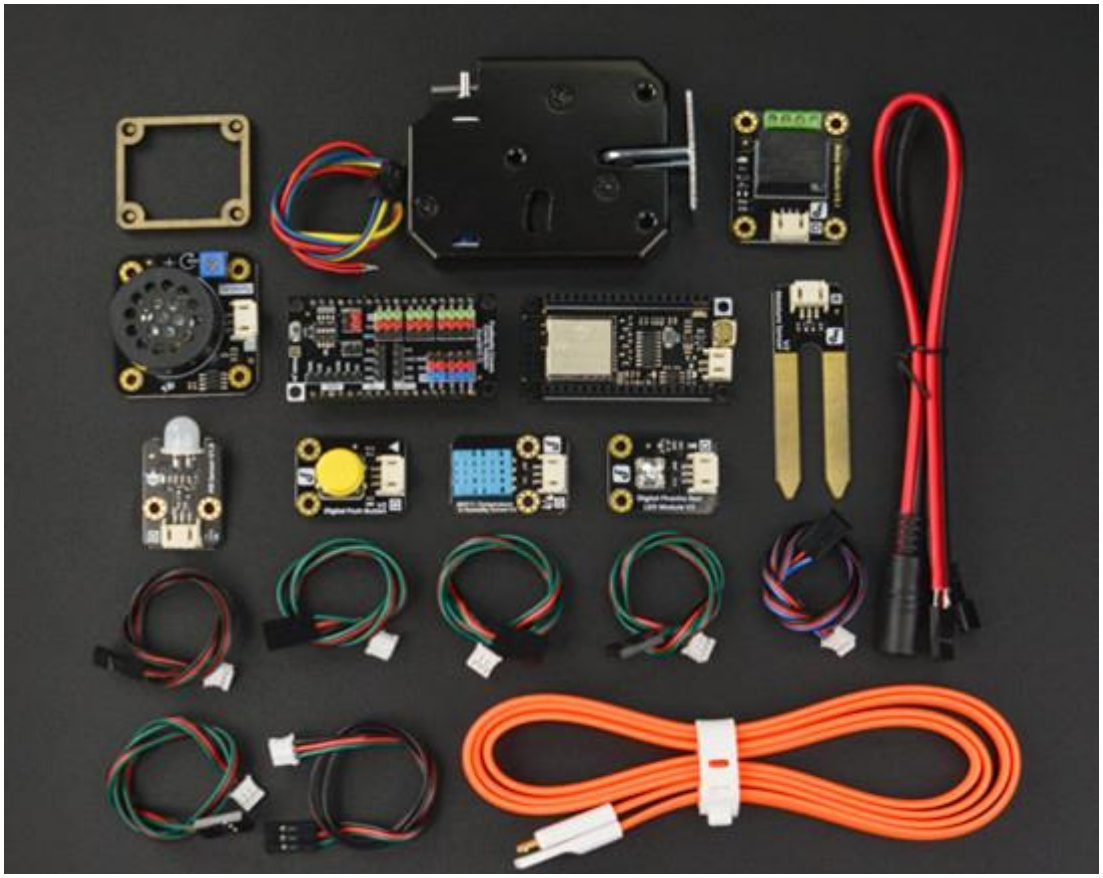


图 4 阿里云 IoT 入门套件模组

如上图所示，DF 为 FireBeetle Board-ESP32 搭配了不同的传感器，所有传感器都可连接到 FireBeetle Gravity 传感器扩展板上，FireBeetle Gravity 传感器扩展板引脚完全兼容 FireBeetle Board-ESP32 主控。

套件中具有模拟/数字传感器输入、输出模块，主要分为专业的传感器模组（如土壤湿度传感器，红外热释电传感器）和丰富的执行器模组（如按键，LED 灯等），具体分类如下表所示：

套件模组	属性	套件模组	属性
红色 LED 模块	执行器模组	土壤湿度模块	传感器模组
黄色按键模块	执行器模组	DHT11 温湿度模块	传感器模组
继电器模块	执行器模组	电磁锁	执行器模组&传感器模组

功放喇叭	执行器模组	FireBeetle Board-ESP32 主控	低功耗 IoT 主板
红外热释电	传感器模组	FireBeetle Gravity 扩展板	Gravity 扩展板

关于传感器模组和执行器模组的说明，请参考后续教程。

第二章 阿里云 IoT 平台

2.1 认识阿里云 IoT

阿里云 IoT 致力于实现万物互联的美好世界，为生态合作伙伴提供基于云端一体化、安全物联网基础平台等，在通过该平台高效连接，管理设备的同时，其开放能力使合作伙伴更高效、低成本地构建各种创新的物联网应用场景。

阿里云物联网平台为设备提供安全可靠的连接通信能力，向下连接海量设备，支撑设备数据采集上云；向上提供云端 API，指令数据通过 API 调用下发至设备端，实现远程控制。

此外阿里云 IoT 还提供了丰富的开发服务，用户可以直接在该平台上搭建 Web 可视化、移动应用、服务开发等开发服务，这降低了物联网项目开发的难度，有了它，用户无需任何专业的开发技巧也可开发自己的项目。

2.2 注册并登陆阿里云账号

打开谷歌浏览器，输入网址 https://www.aliyun.com 进入阿里云主界面，如图 5 所示，点击**登陆**，在密码登陆栏点击**免费注册**，进入阿里云账号注册界面(若已有账号可直接登陆)，如图 6 所示，按要求完成注册即可。

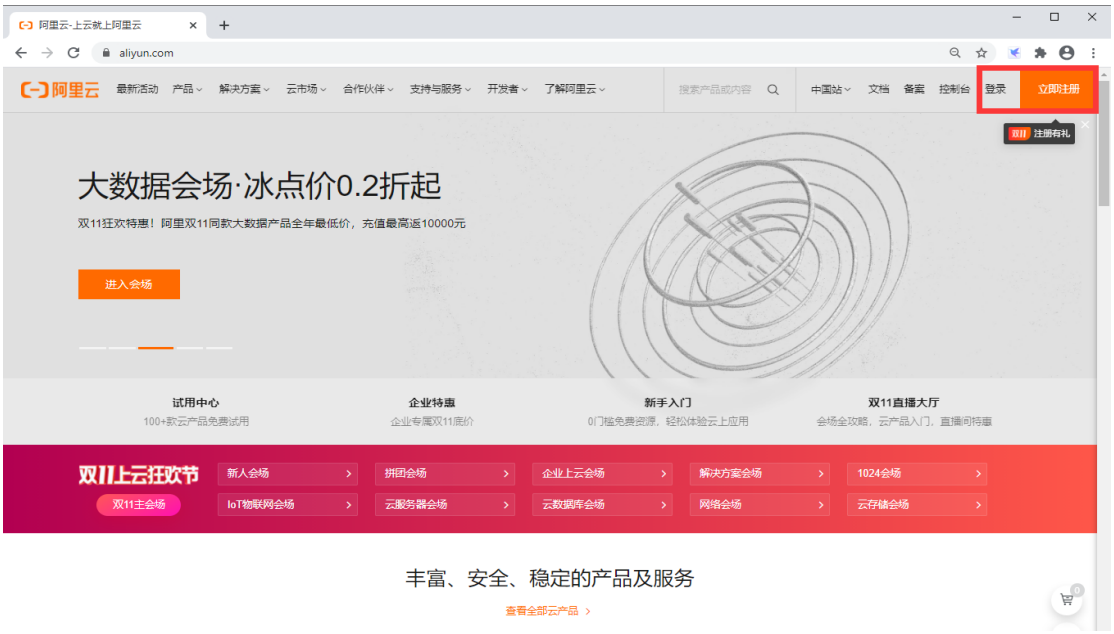


图 5 阿里云主界面

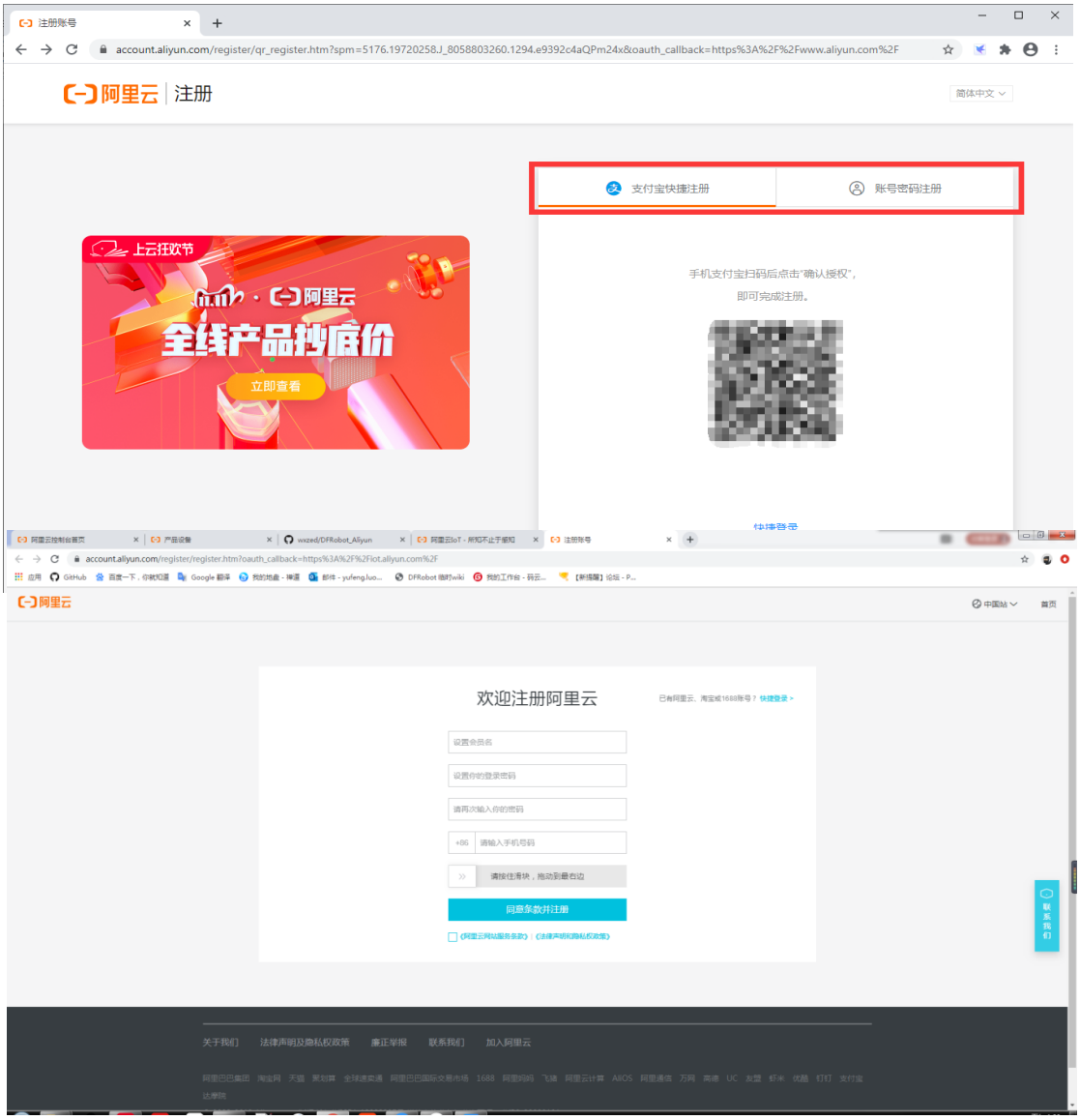
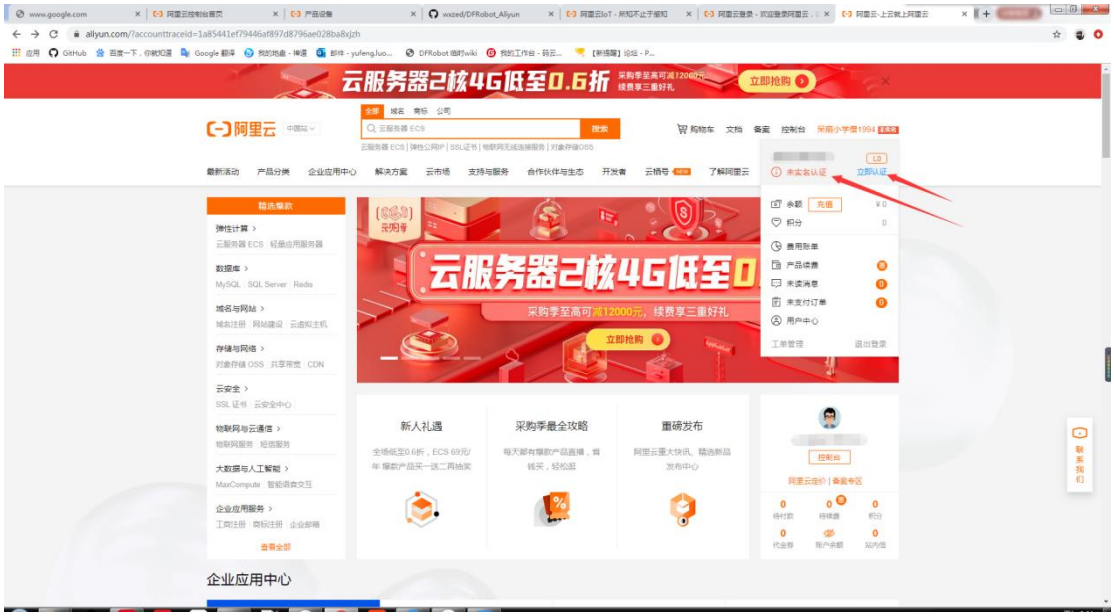
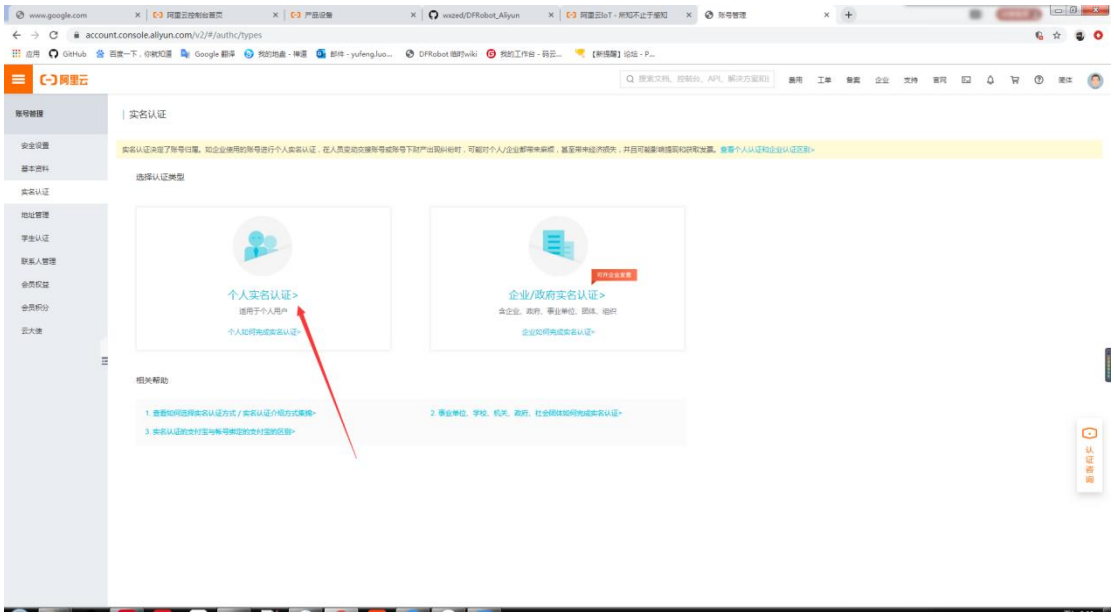


图 6 选择支付宝快捷注册或账号注册界面

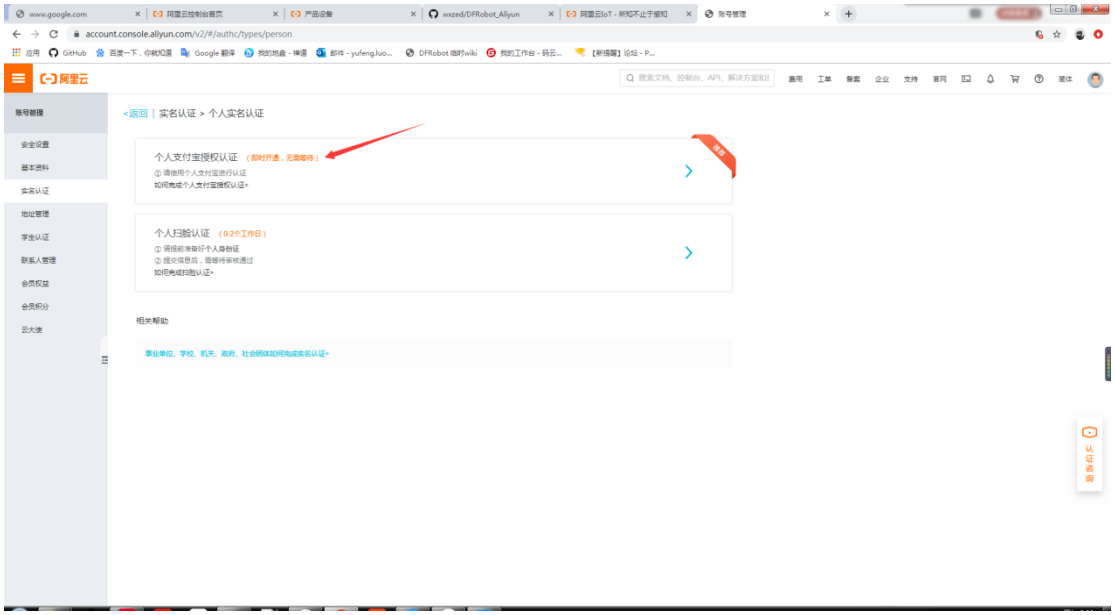
2.3 实名认证

第一次使用，需进行实名认证，在注册完成后，会回到主页面。在用户名栏点击实名认证，根据提示操作完成认证即可。

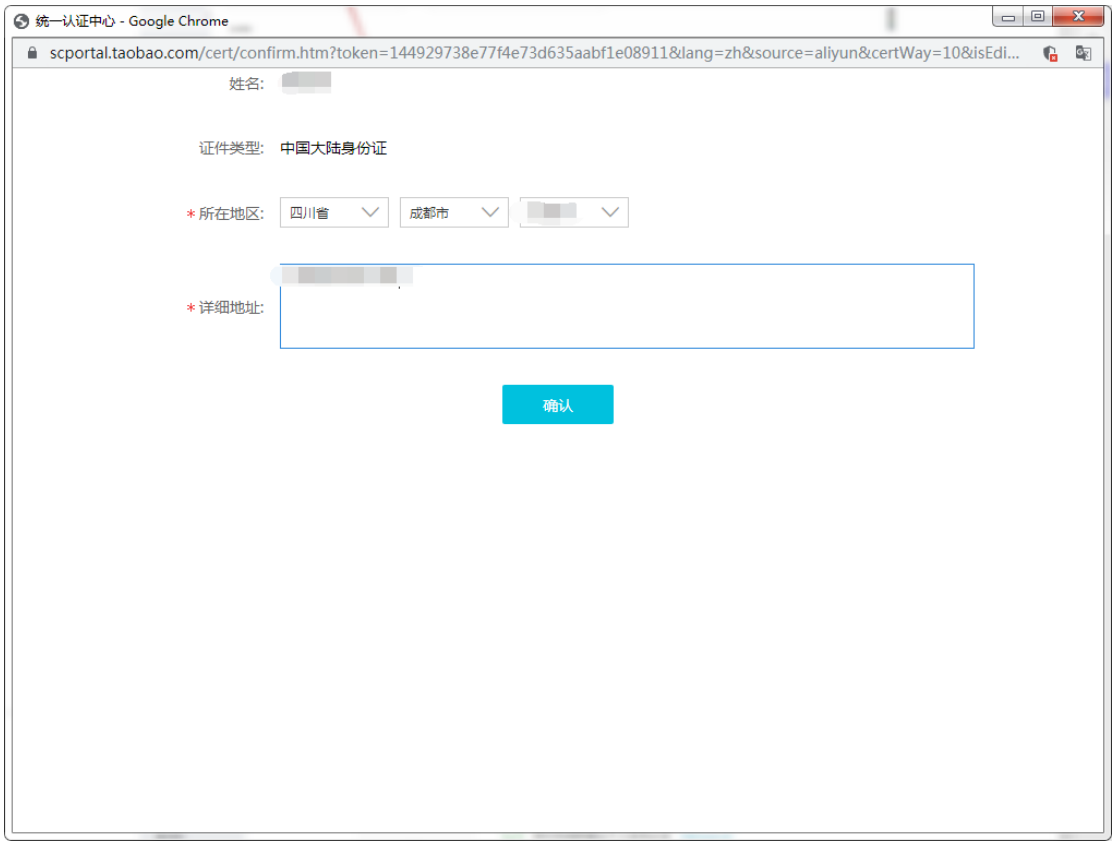


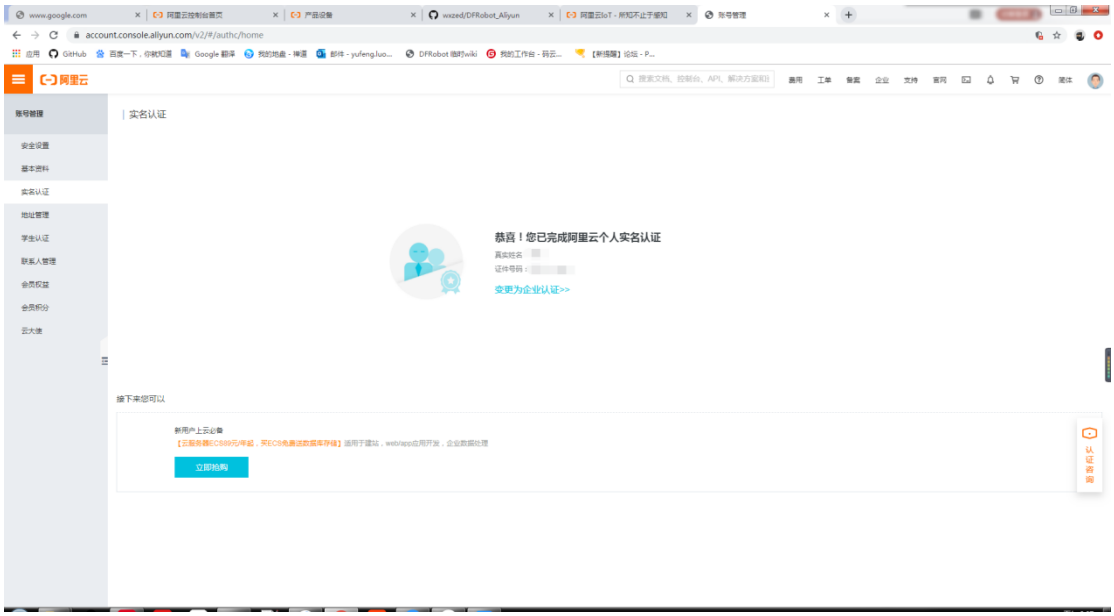


直接选择支付宝认证，如下图所示：



接下来需要填写真实个人信息和住址：





2.4 开通物联网服务

当我们实名认证成功后，是没有服务的。所以我们需要自己开通物联网服务。

1，点击首页的菜单键。

2，点击"产品和服务"选项。

3，选择"物联网 IOT"选择。

4，点击"物联网平台"。

选择开通物联网服务，按照提示步骤进行操作，直到提示服务开通成功。

2.5 登陆物联网平台控制台

物联网平台是阿里云 IoT 的控制台，阿里云上有关物联网的操作，都必须在该平台上进行，如本教程中提到的产品创建、设备添加、服务创建等，都需要在该控制台上进行。

打开谷歌浏览器，输入网址 https://iot.aliyun.com/，并登录。登录成功后，点击**物联网产品->物联网平台**，如图 7 所示，进入物联网平台页面，点击**管理控制台**(注：若是第一次进入该页面，应该是**立即开通**，根据要求开通即可，是免费的)，如图 8 所示，进入物联网平台控制台，如图 9 所示：

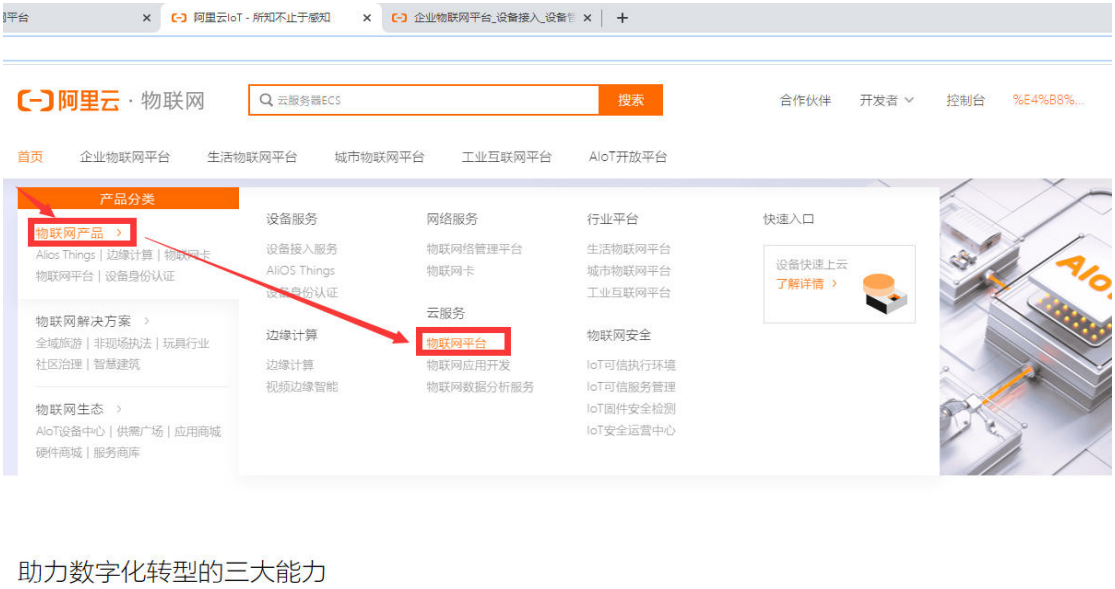


图 7 物联网平台接口示意图

用户可以在该控制台完成产品及设备的创建，以及相应服务开发，该教程的所有步骤都与该控制台相关。

图 8 管理控制台接口

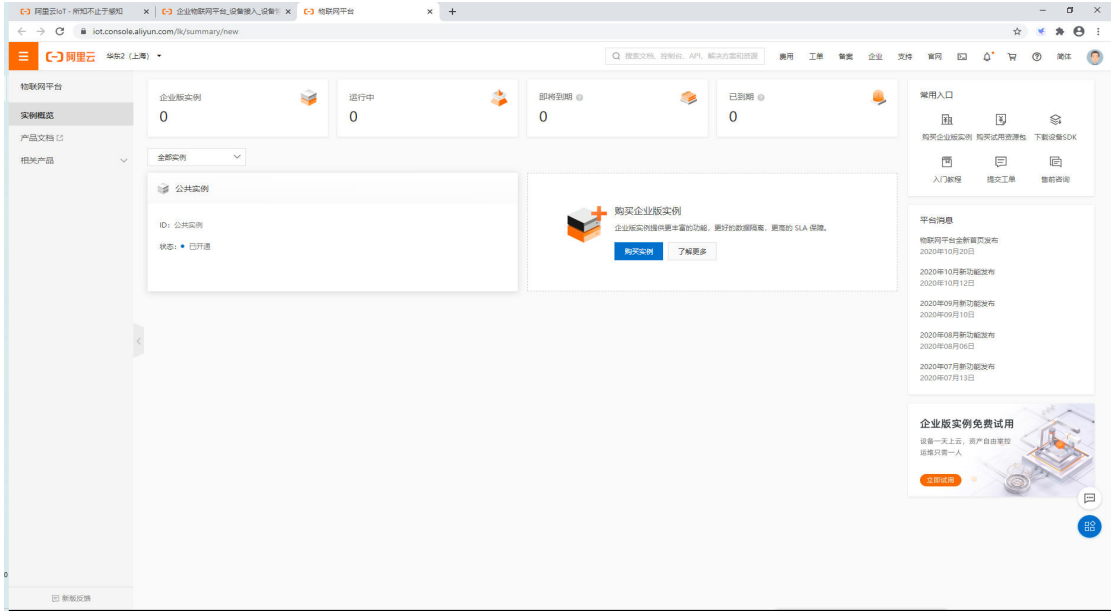


图 9 物联网平台控制台

2.6 使用物联网平台

STEP1 创建产品

登录并进入物联网平台控制台(可参考 2.3 小节)，在控制台找到**公共实例**并点击，跳转到新页面后，在左侧菜单栏点击产品，点击**创建产品**，弹出“新建产品”页面，如图

10 所示。目前开通的服务只有公共实例，便于原型调试或个人用户使用。企业实例需要额外付费。

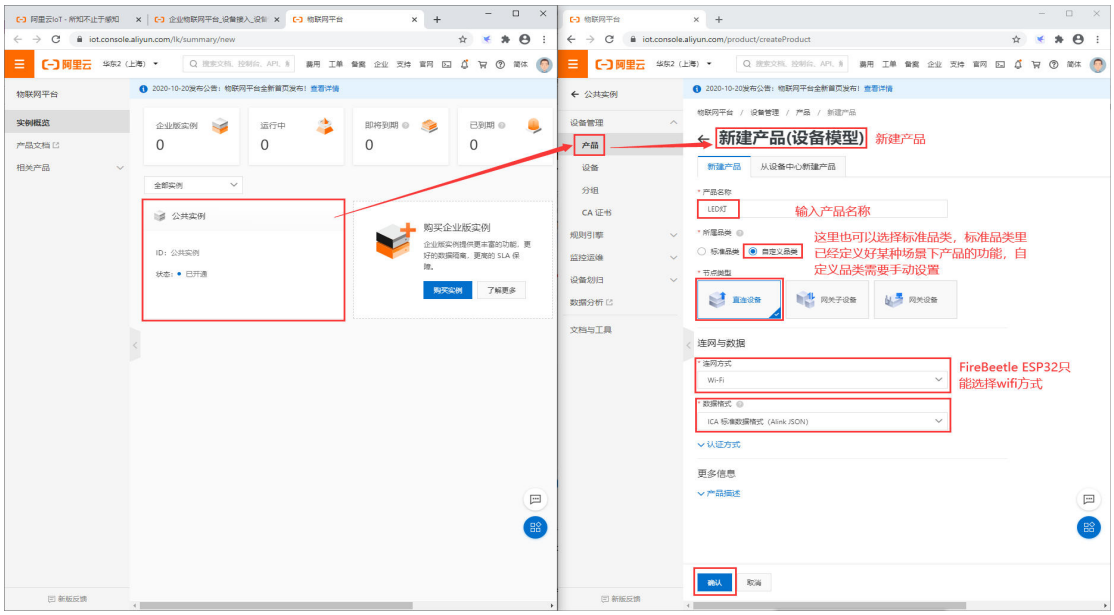
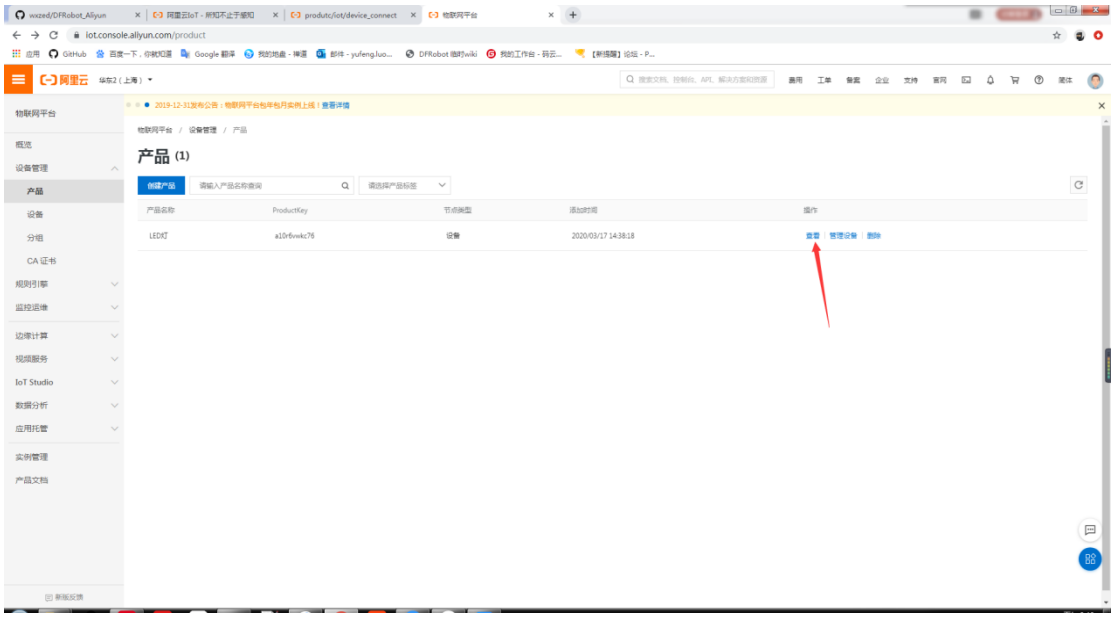


图 10 产品创建示意图

按照页面提示，填写信息，例创建一个名为 **LED 灯**的产品，单击**保存**。

产品创建完成后，会自动出现在产品列表中。接下来为产品定义相应的功能。



在产品列表中找到刚才创建的产品，单击该产品的**查看**按钮，进入产品详情页，点击**功能定义**，点击**编辑草稿**，在自定义功能栏，单击**添加功能**。按照页面提示，填写信息(注：要留意产品标识符，会用于 Arduino 与阿里云平台之间的连接，该产品标识符可在自定义功

能栏查看)，点击**完成**。具体设置参考图 12 所示：

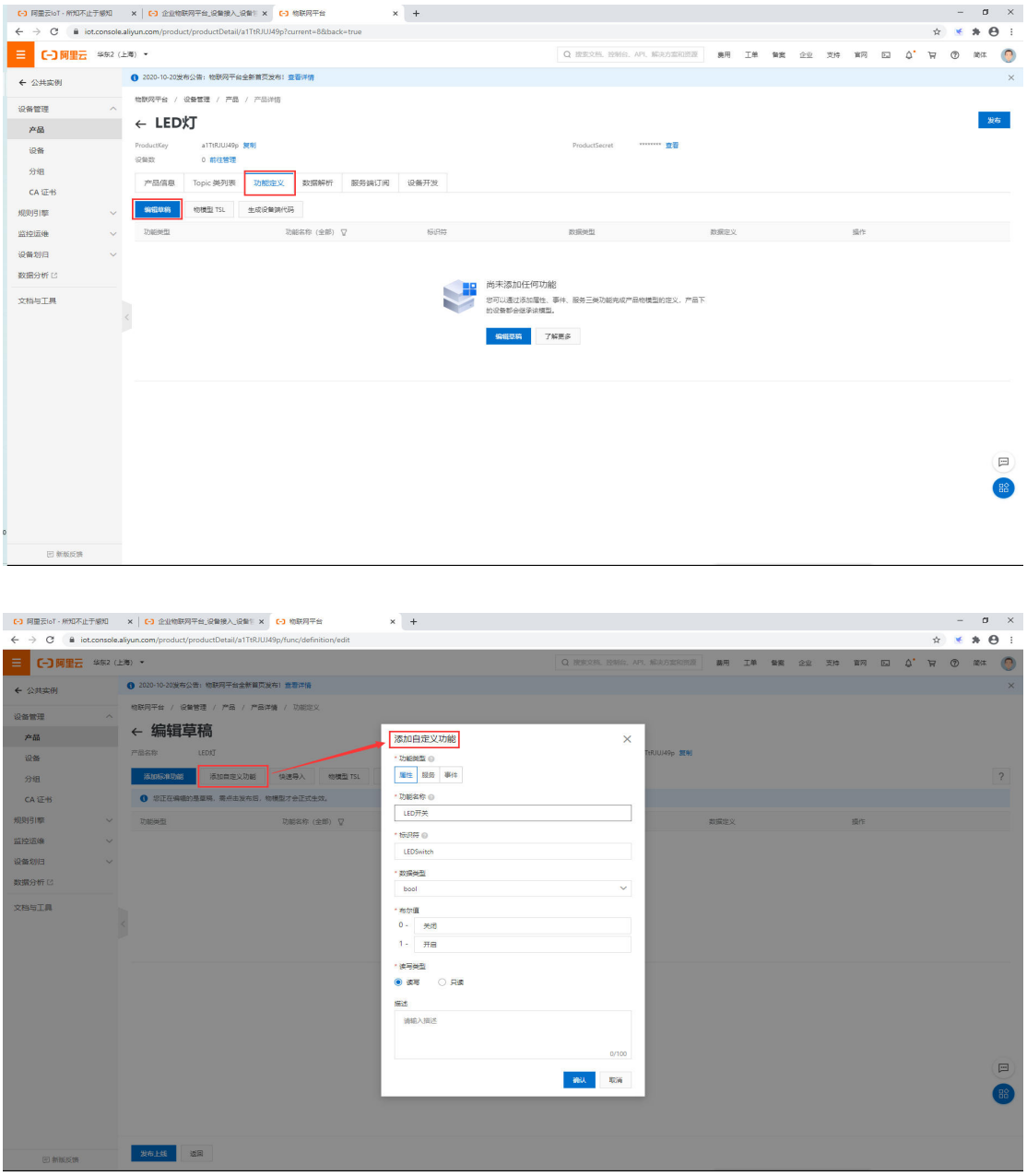


图 12 产品功能定义页

功能添加完成后，会自动出现在自定义功能栏，我们可以在这里查看产品的标识符，如图 13 所示：

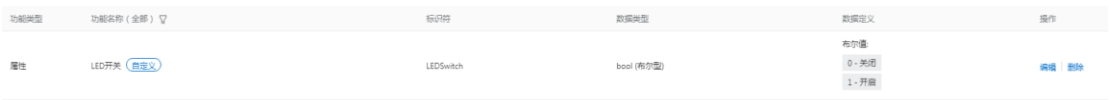


图 13 产品标识符查看栏

STEP2 创建设备

登录并进入物联网平台控制台的公共实例页面(可参考 2.5 小节)，在左侧导航栏选择**设备**，单击**添加设备**，选择一个已创建的产品。选择后，新建的设备将继承该产品定义好的功能和特性。填入 DeviceName。如果不填，系统将自动生成，用以标识设备(用于标识同一产品下的多个设备)。具体设置步骤参考图 14:

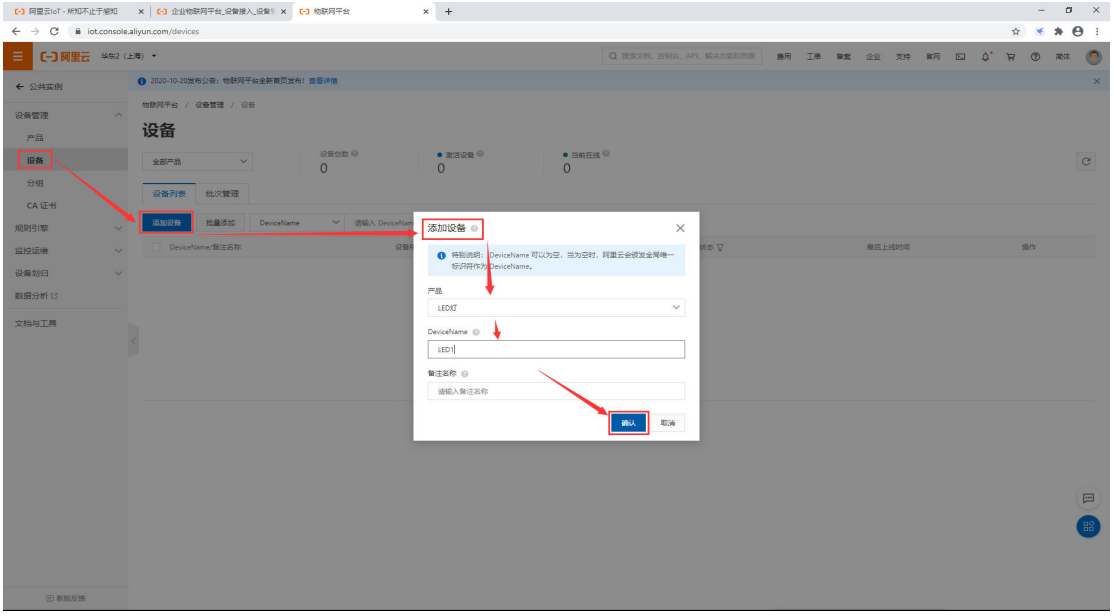
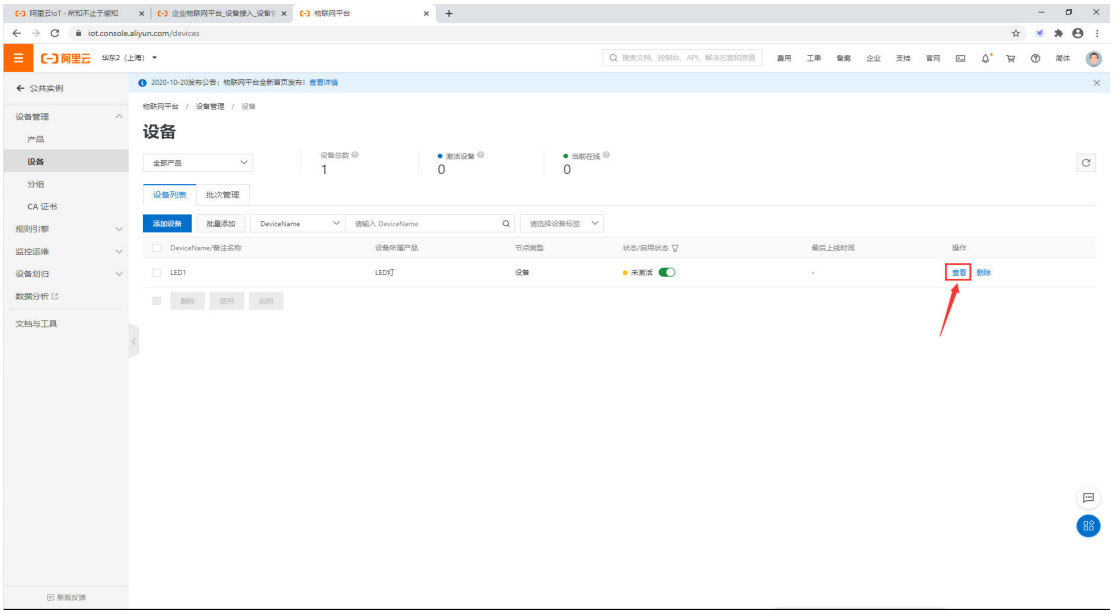


图 14 设备添加

点击**确认**，完成设备创建，点击查看可以查看设备信息。



点击查看设备证书：

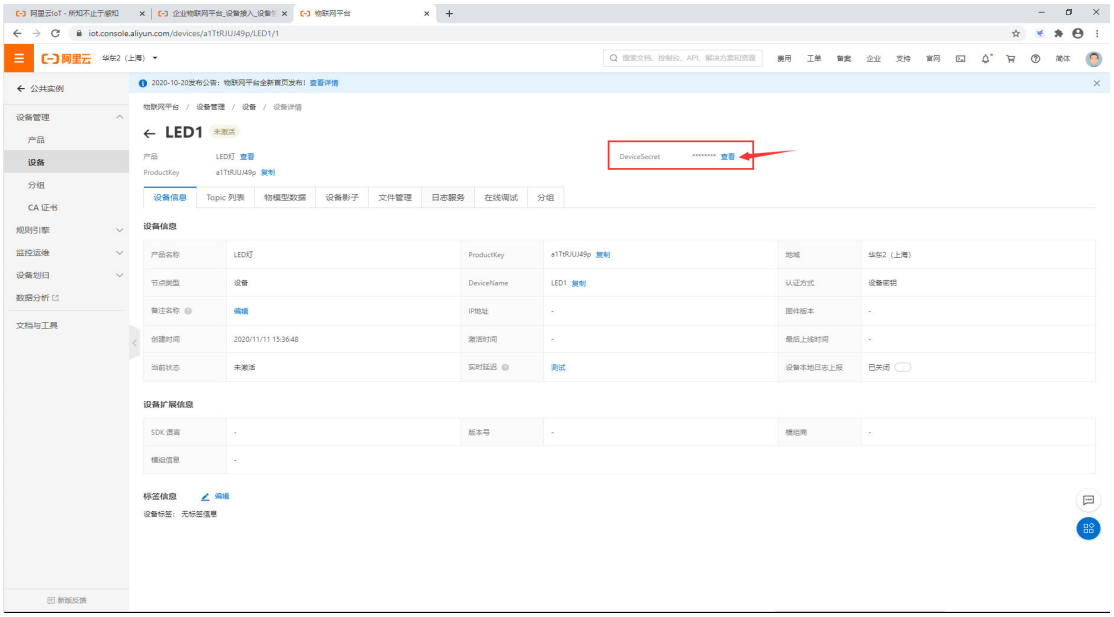


图 15 设备信息

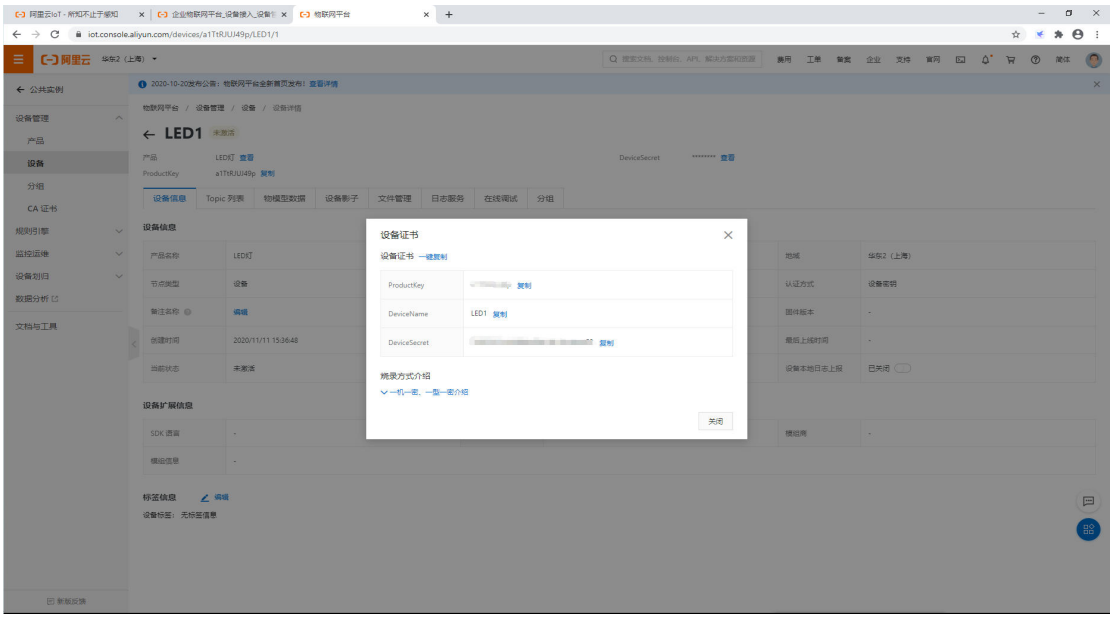


图 16 设备证书

你可以查看、复制设备证书信息。设备证书又名设备三元组，由设备 ProductKey、DeviceName、和 DeviceSecret 组成，是设备与物联网平台进行通信的重要身份认证，建议您妥善保管。

·ProductKey：物联网平台为您创建的产品颁发的全局唯一标识符。

·DeviceName：设备在产品内的唯一标识符，用于设备认证和通信。

·DeviceSecret：物联网平台为设备颁发的设备秘钥，用于认证加密，需与 DeviceName 成对使用。

此外在设备详情页，单击 **Topic 列表**，即可查看设备的上报及订阅 TOPIC，如图 17 所示。该信息用于设定设备与物联网平台进行通信时，是设备信息上传，还是云端数据下发，或二者兼之。

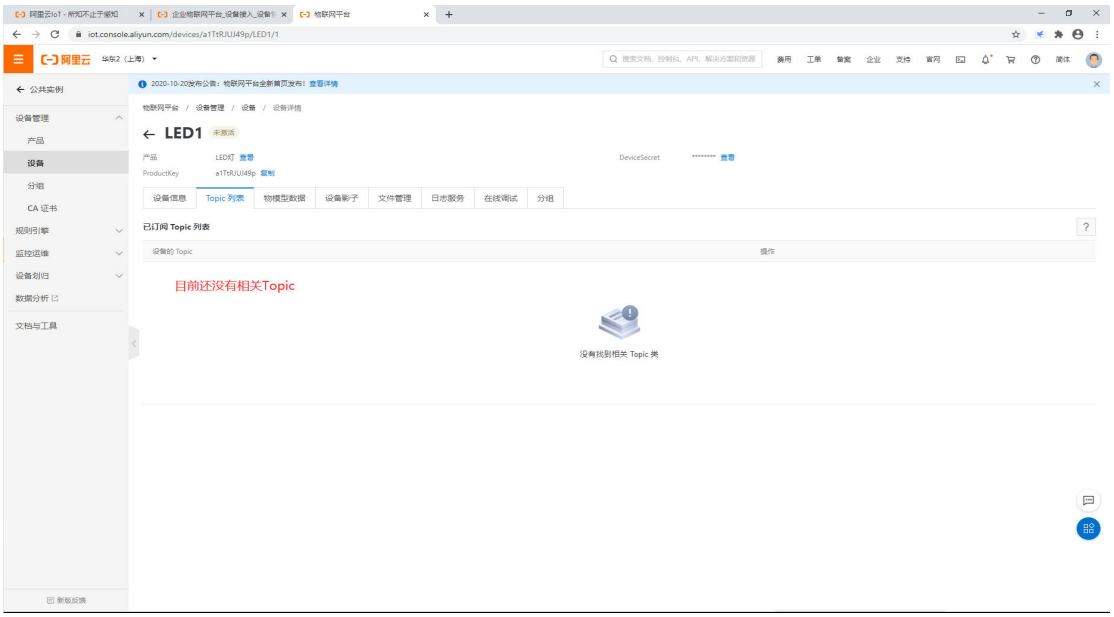


图 17 设备上报和订阅 TOIC

STEP3 物联网开发服务

物联网开发服务是阿里云针对物联网领域提供的开发服务，可覆盖各物联网行业核心应用场景，旨在解决物联网开发领域开发链路长、技术栈复杂、协同成本高、方案移植困难等问题。开发服务提供了 Web 可视化开发、移动可视化开发和服务开发工作台等便携的物联网开发工具。

无论是哪种服务开发，都必须先建立一个项目，然后再在该项目下创建服务。在这里我只简单的介绍如何创建项目，导入产品，以及如何进入服务创建页面，至于具体的服务拖建说明请参考物联网开发服务文档，那里有详细的说明。简而言之，开发服务的搭建可分为以

下 3 个步骤：

·新建项目；

·导入产品；

·服务创建及开发；

● 准备工作

在服务开发之前，我们先创建一个项目，并将已建好的产品导入该项目中。

项目创建

登录并进入物联网平台控制台(可参考 2.5 小节)，在左侧导航栏选择**相关产品->IoT Studio**，单击**项目管理**，单击**新建项目**，选择**创建空白项目**。根据提示，填写信息，如图 18 所示：

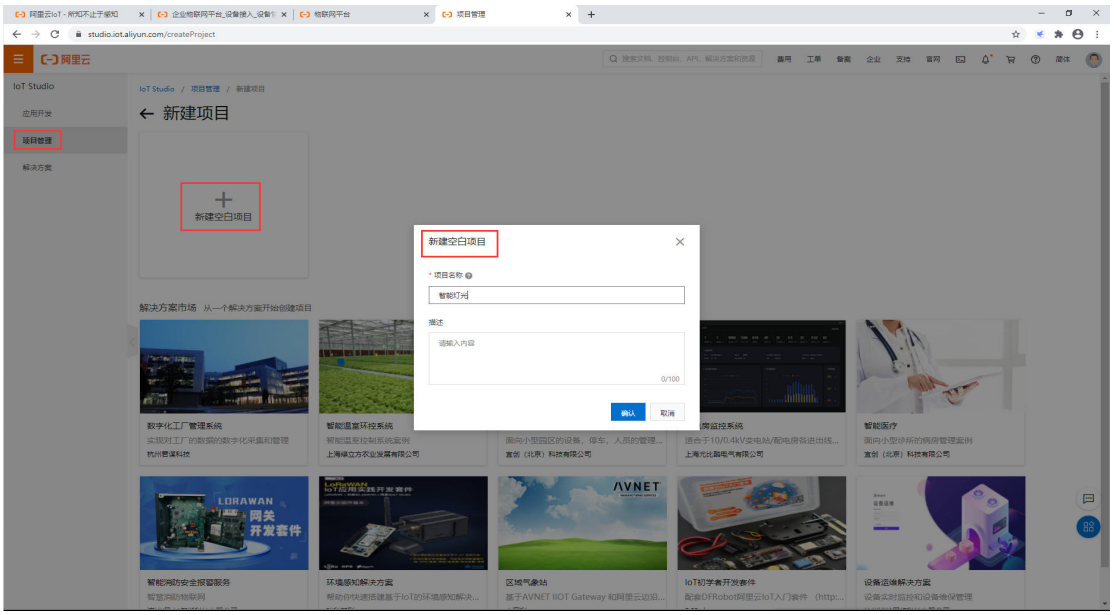
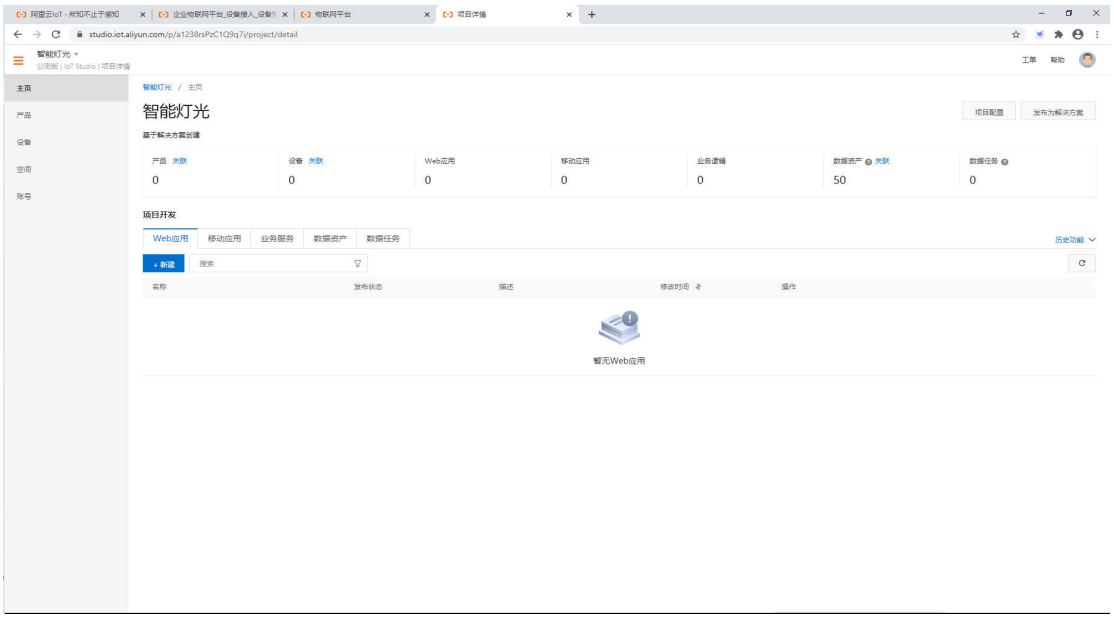


图 18　项目创建框

单击**确认**，项目创建完成后，会自动进入到项目中,如下图所示：



产品导入

在项目列表中找到刚才创建的项目（智能灯光），单击，进入该项目概览页。我们可以在该页面下导入已经创建好的产品(注：已经在项目中的产品，无法再导入其他项目)，点击概览页中的**产品**，再点击**关联物联网平台产品**，弹出关联产品对话框，选择要导入该项目的产品，如图 19 所示，。

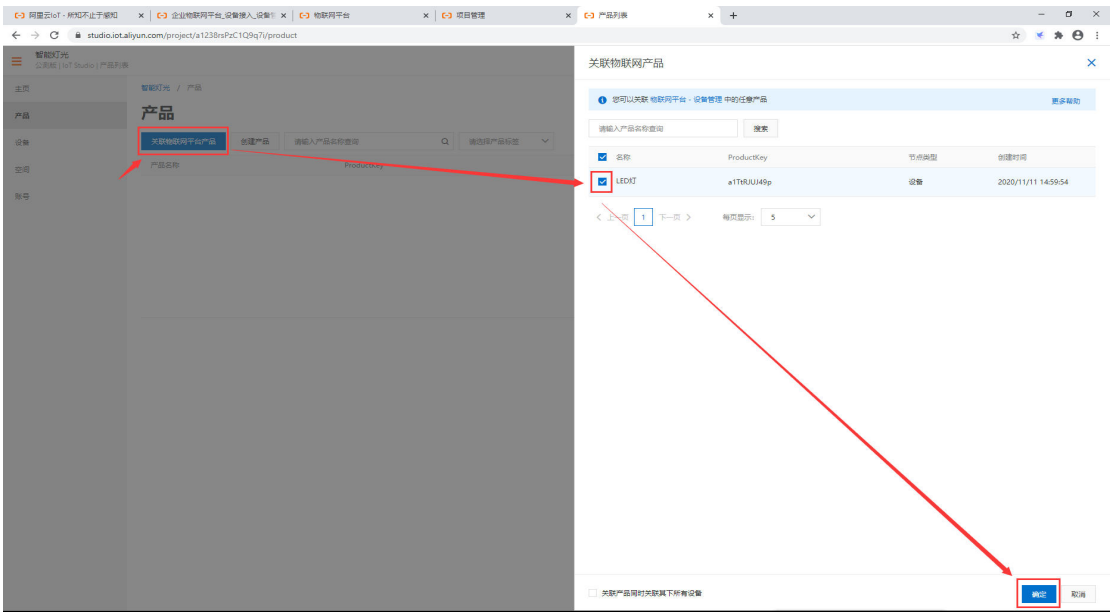
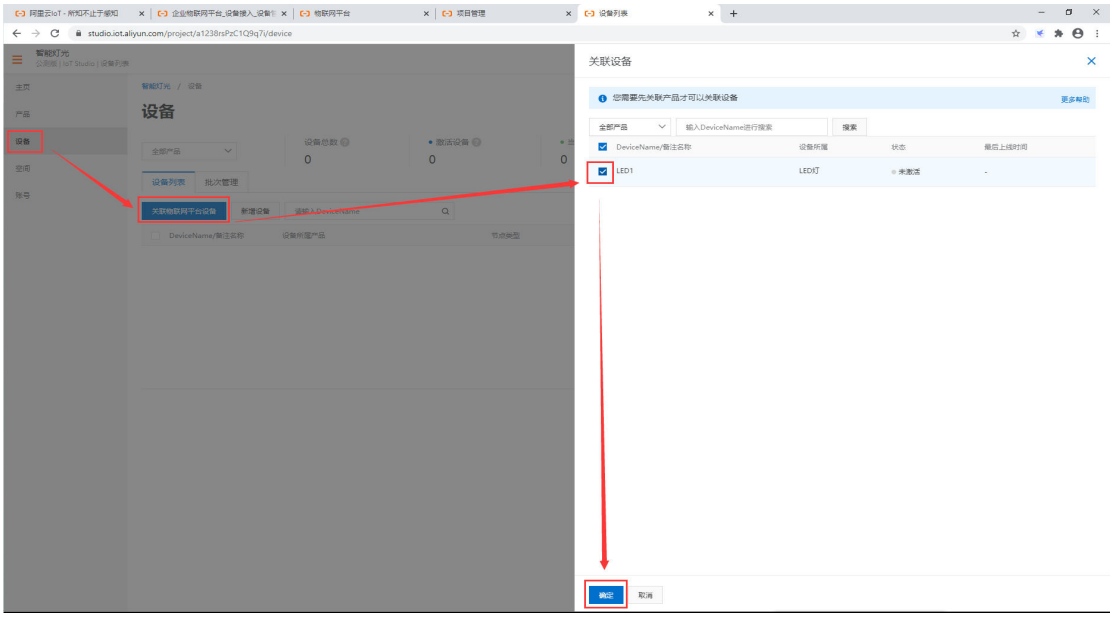
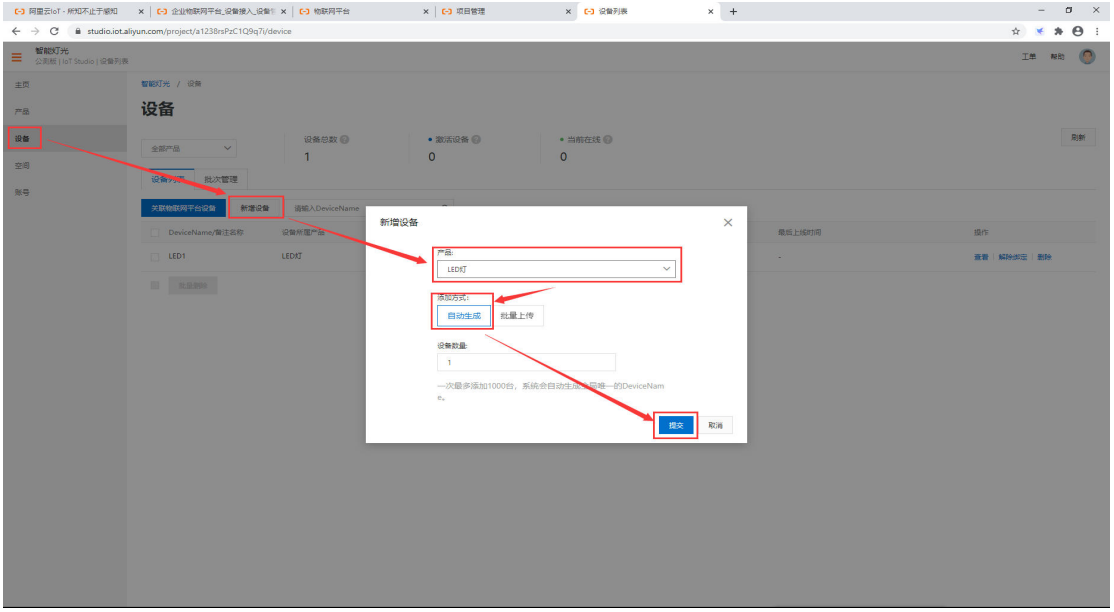


图 19 导入产品示意图

产品导入成功后，该页面的**关联产品**数量就会发生变化。这个时候，我们选择关联之前创建的**设备**，或者在该产品下新建一个设备，如果是关联设备的话，操作如下图所示，点击设备，选择关联设备，然后选择刚才导入产品下挂载的设备导入即可。



在项目下随机生成一个设备的方法。**以下的应用都以随机生成设备的方式进行操作，不再使用导入的方式进行制作。如需要外部导入设备的，请参照以上的步骤！**

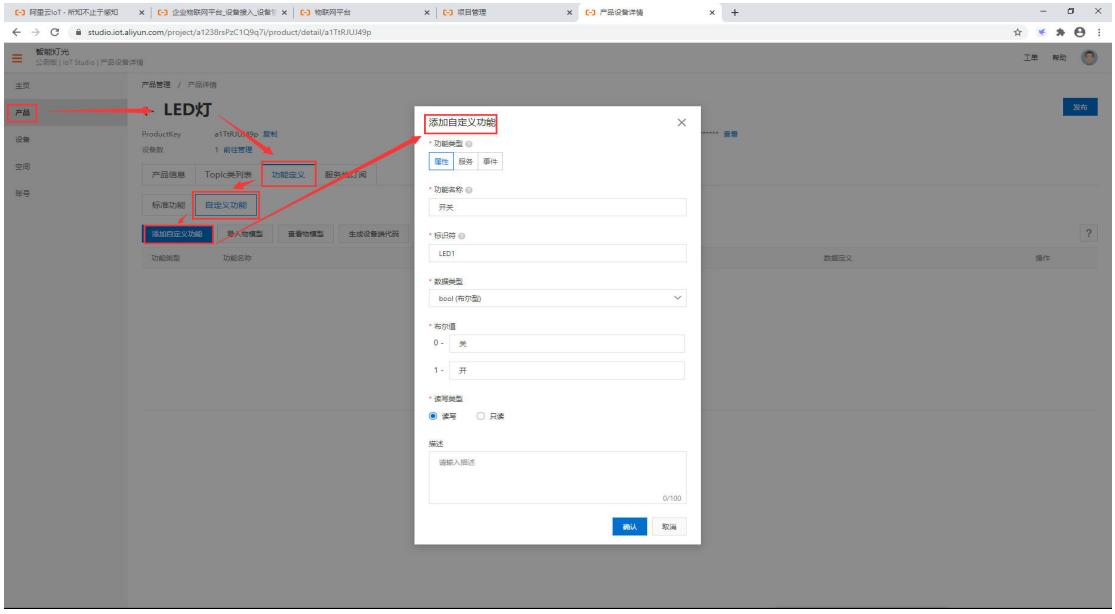


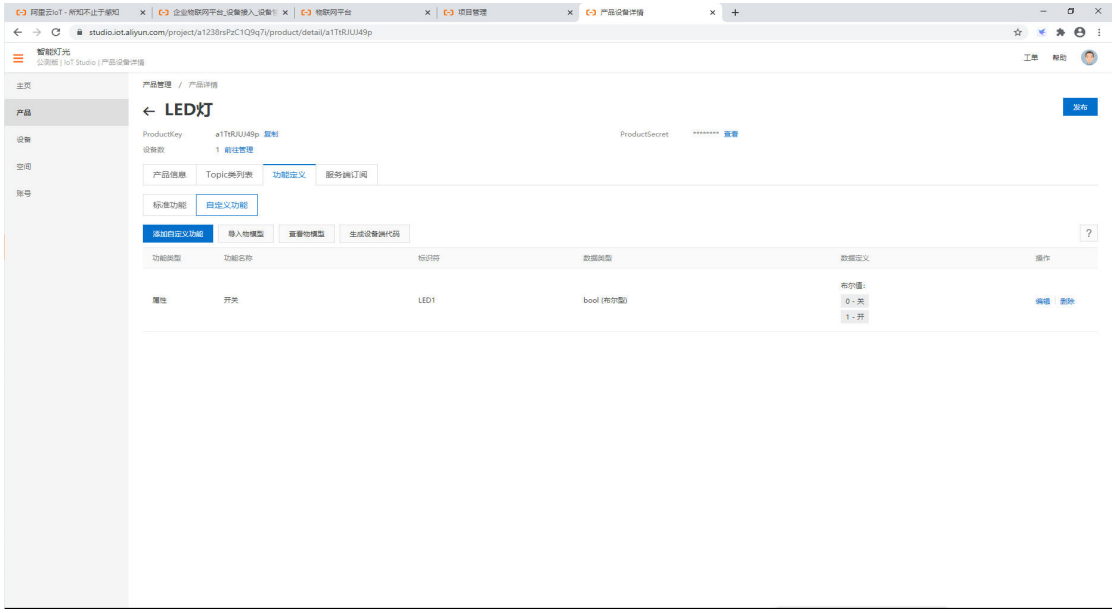
在IoT Studio页面中，可以选择对信息进行查看，选中左侧导航栏的**项目管理->选择正在搭建的项目->产品**，即可产看导入的产品。

点击左侧导航栏的**项目管理->选择正在搭建的项目->设备**，即可查看产品中挂载的设备。用户可以通过此方法查看相应产品和设备的信息；用户也可在项目下直接创建产品和设备，不过需要注意的是在这里创建的设备的名字是由系统自动生成的，无法用户自定义。

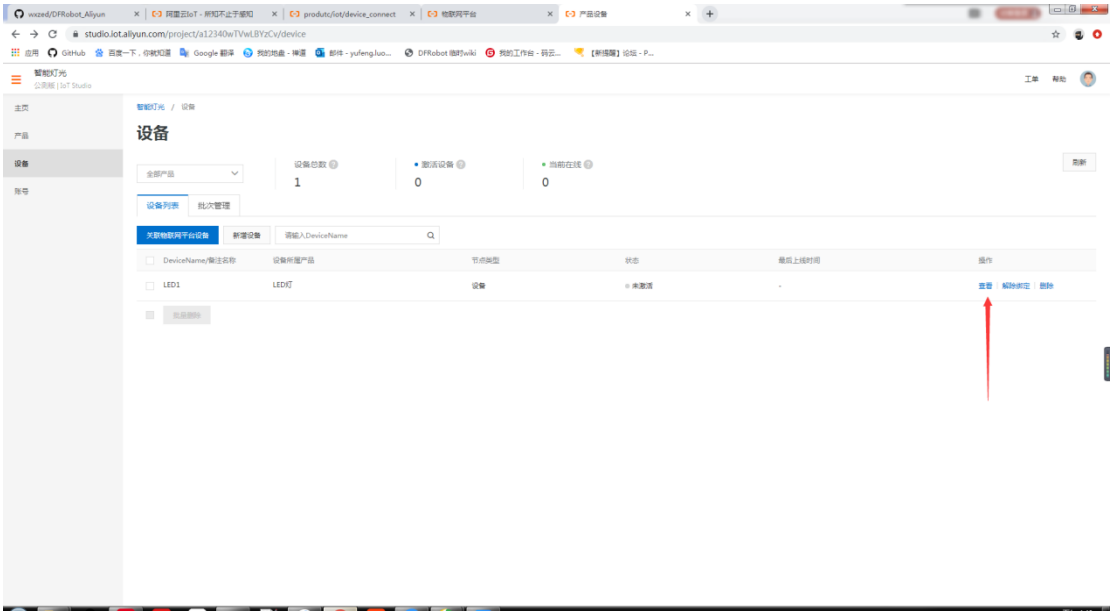
添加功能

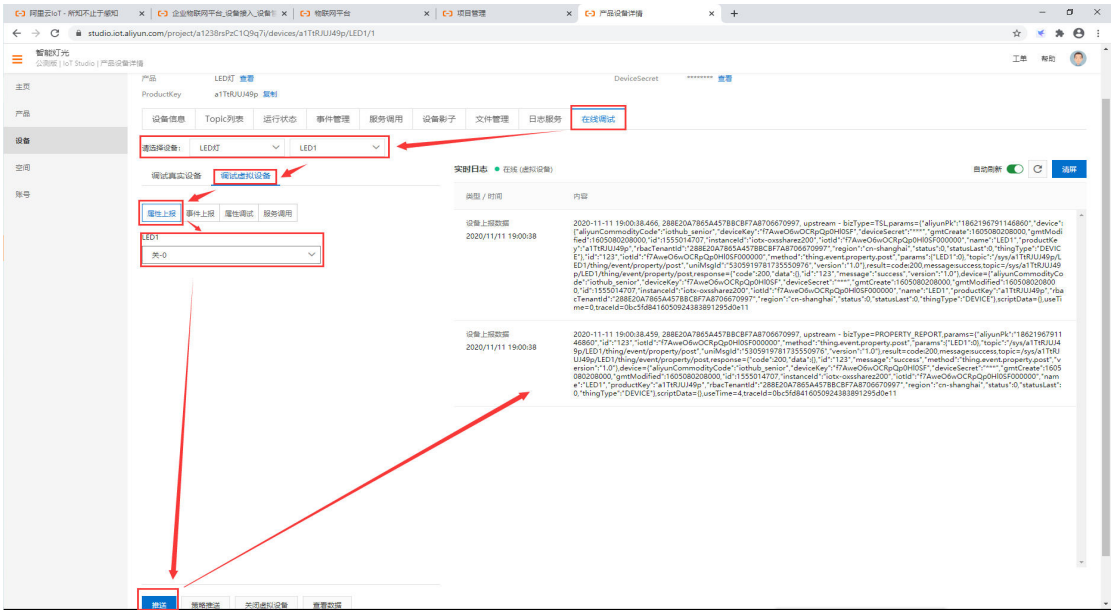
产品和设备搭建好后，需要添加功能才可以正常运行。**项目管理->选择正在搭建的项目->产品**，点击**查看**数，点击**功能定义->自定义功能->添加自定义功能**，按照提示完成输入。





功能创建完成后，进行参数格式的验证，操作步骤如下图所示：





以上步骤都完成后，接下来就可以进行开发服务了，目前支持的开发服务有三种：

- **Web 可视化开发；**

- **业务逻辑开发；**

三种服务开发可通过点击左侧导航栏的 **应用开发**->WEB **可视化开发**或**业务逻辑**，如图 20 所示：

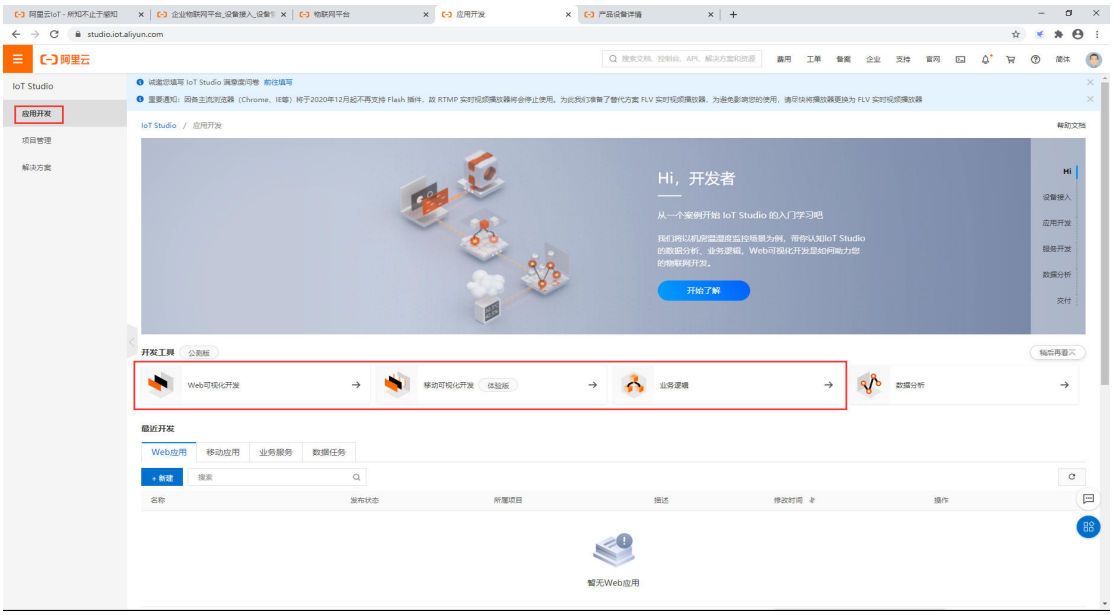


图 20 各开发服务进入接口

由于本套件涉及最多的是移动应用开发和 Web 可视化开发，下面简单的介绍一下如何进行 Web 可视化开发和业务逻辑开发。

• Web 可视化开发

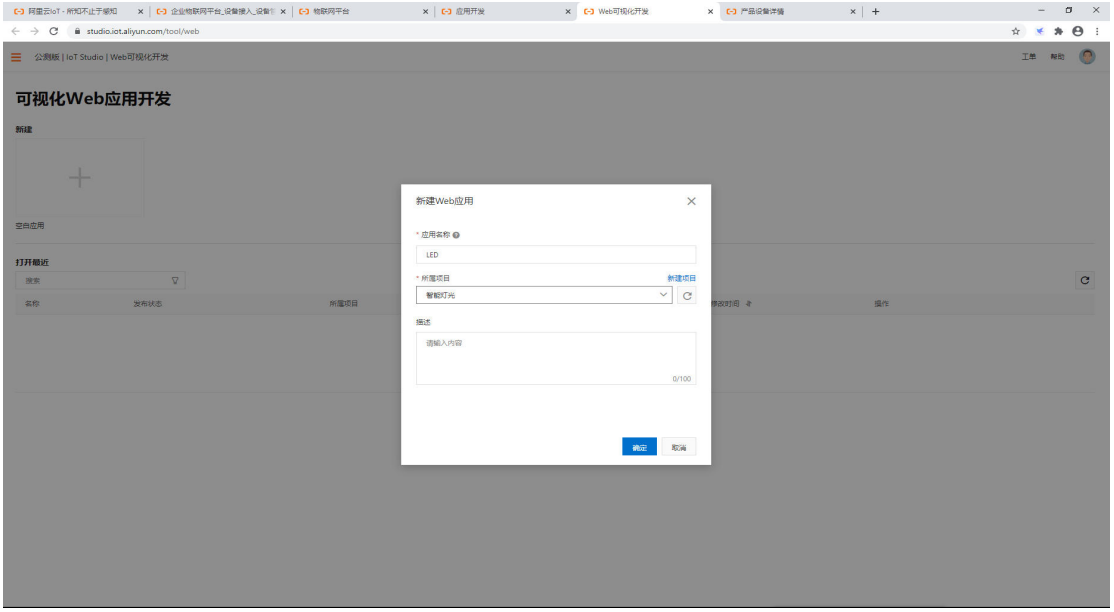
Web 可视化开发操作方式跟移动应用开发类似，大致可以分为以下 5 个步骤：

- 新建 Web 可视化开发；
- Web 页面编辑；
- 可用账户设置（可选）；
- Web 可视化发布；
- Web 查看

新建 Web 可视化开发

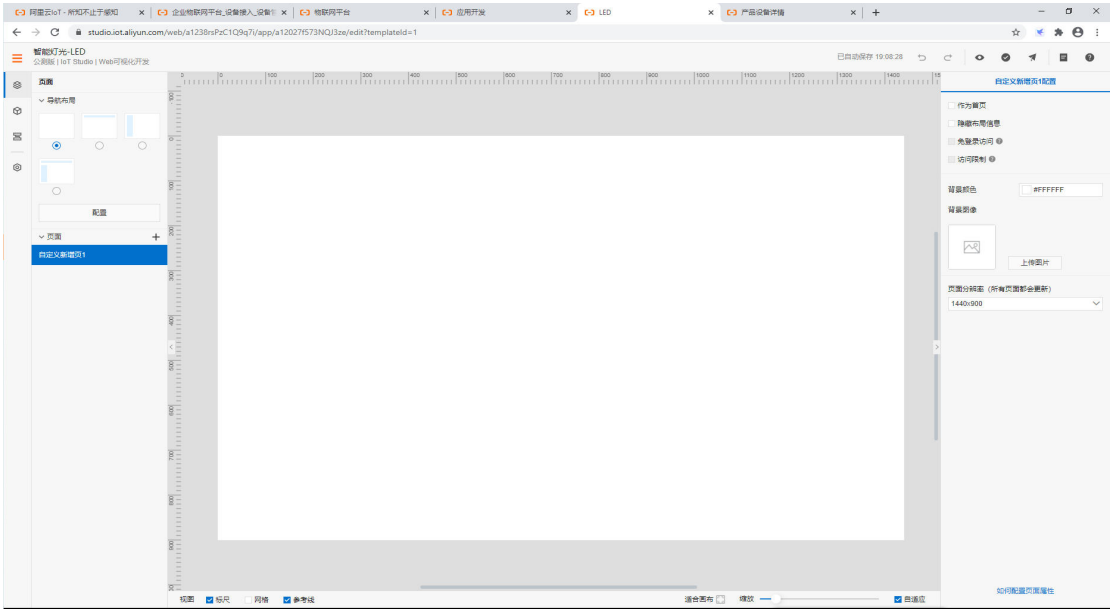
选中 **Web 可视化开发**选项，点击**新建**，即可进入 Web 可视化配置界面。

在弹出的移动可视化开发页面，填写**应用名称**信息，单击**完成**。该名称即为 Web 的名称。



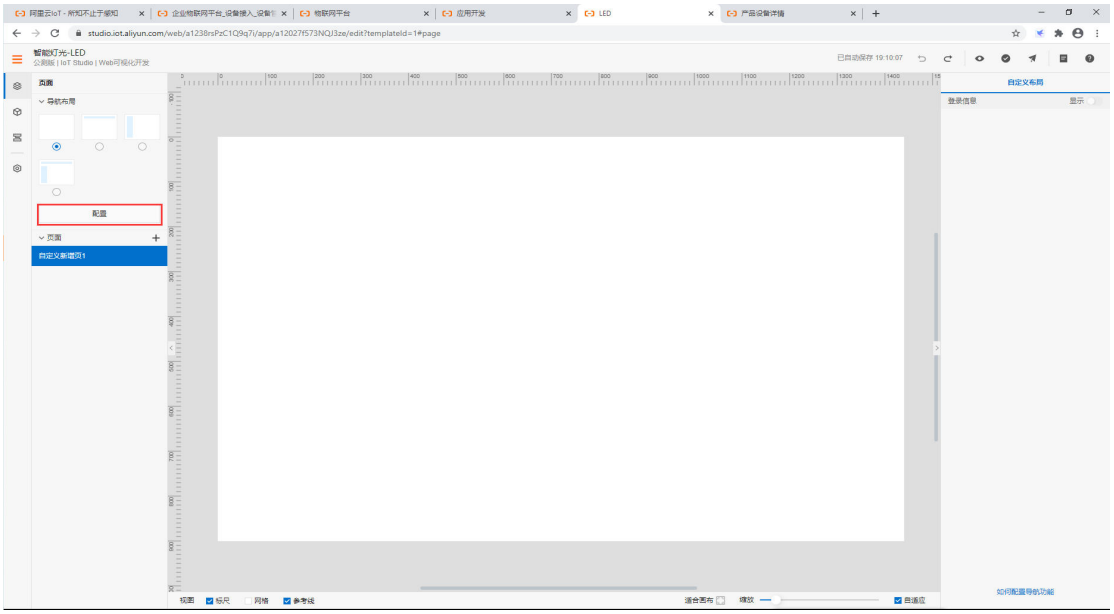
Web 页面编辑

Web 页面编辑跟 App 页面编辑是类似的，明显的不同之处在于 Web 页面没有首页模块，组件种类更多，操作方法上也是差不多的，所以我这里就不过多赘述了。



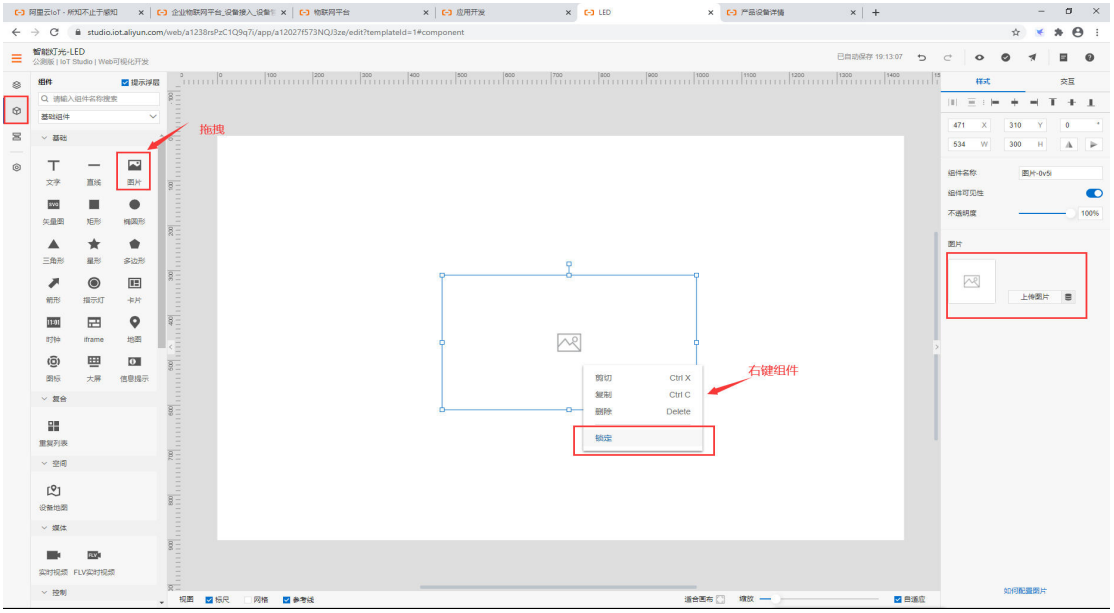
下面就进入 Web 的页面编辑

Step1:将自定义新增页 1 改为 LED

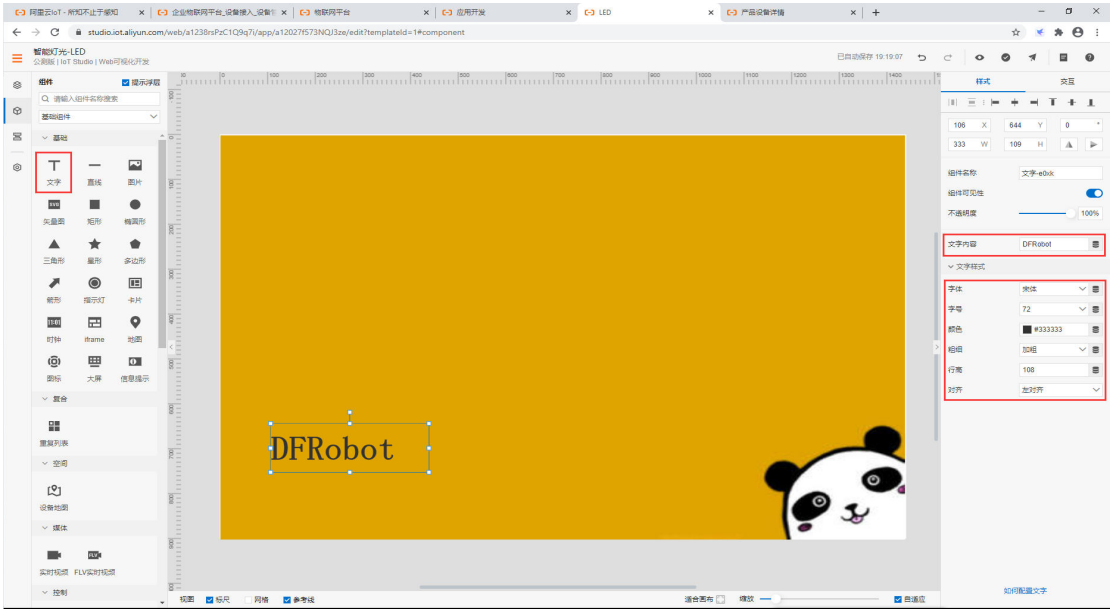


Step2:添加组件。

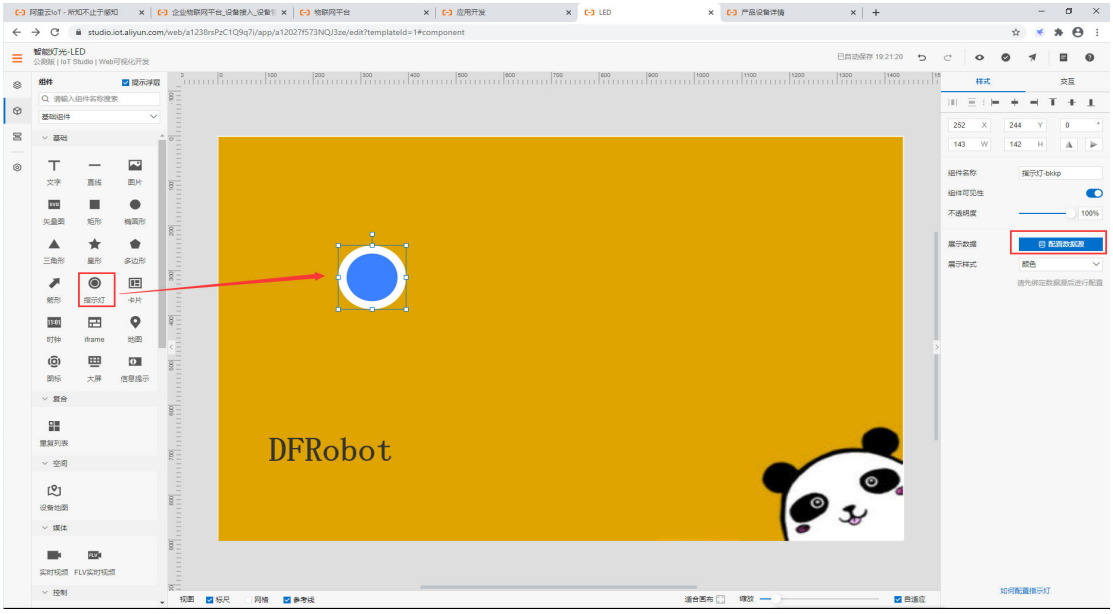
为了美观，我们放上一张背景图，所以先添加一张**图片**，将图片拖到画布中，设置为**锁定**。然后上传一张比较好看的图片。背景图原图如下。

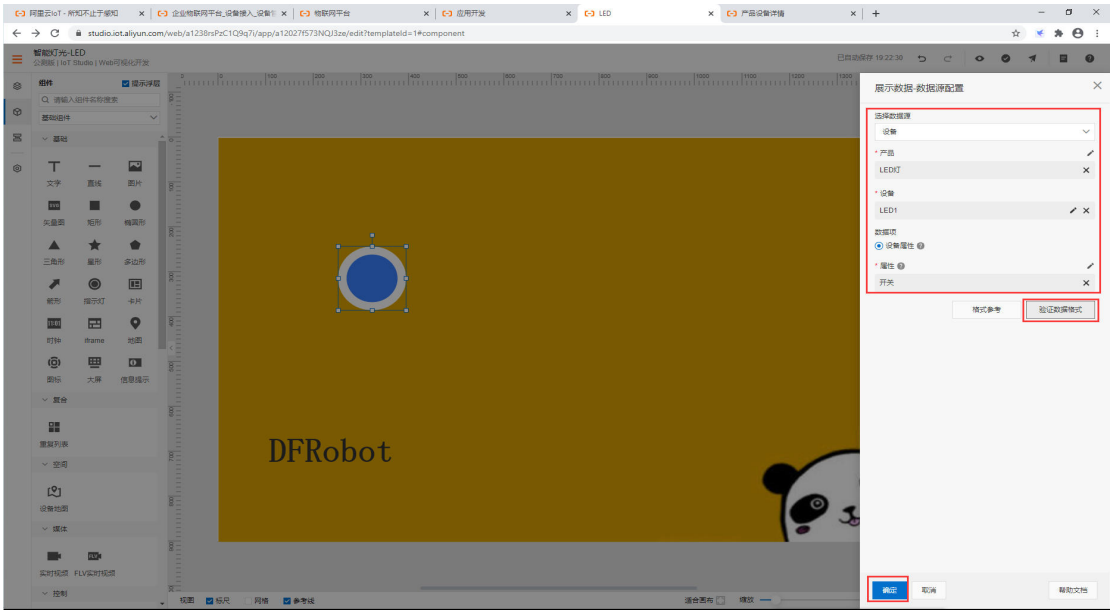


添加文字。将**文字**拖到画布中，改为 **DFRobot**。

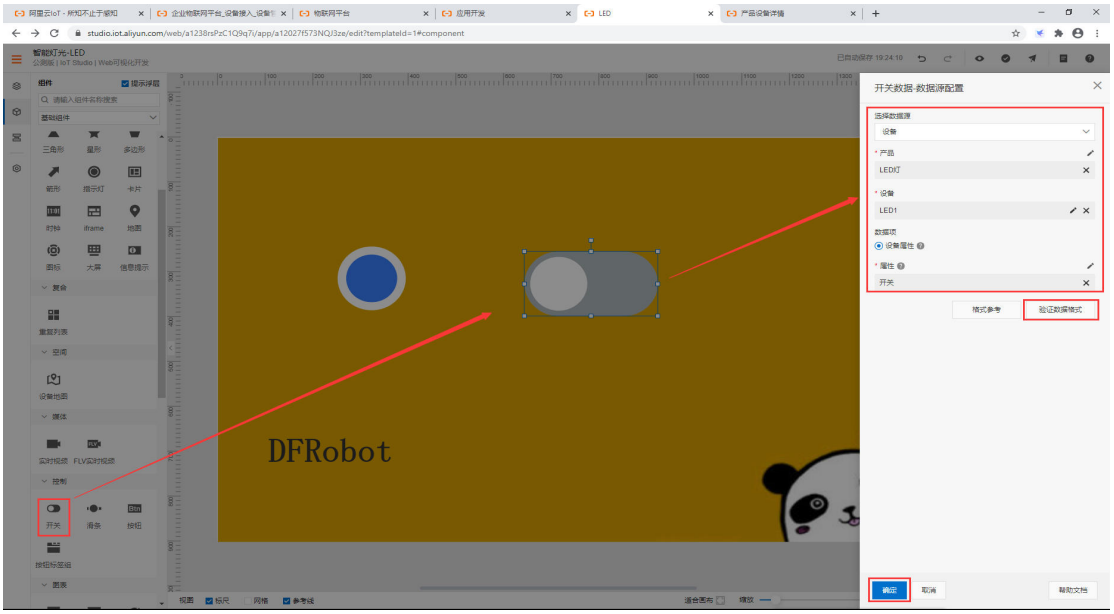


添加指示灯。将**指示灯**拖到画布中，然后**配置数据源**，产品配置为 LED 灯,设备配置为刚才在项目中创建的设备,属性为工作状态，点击**验证数据格式**，再点击**确定**，这就配置完成了。

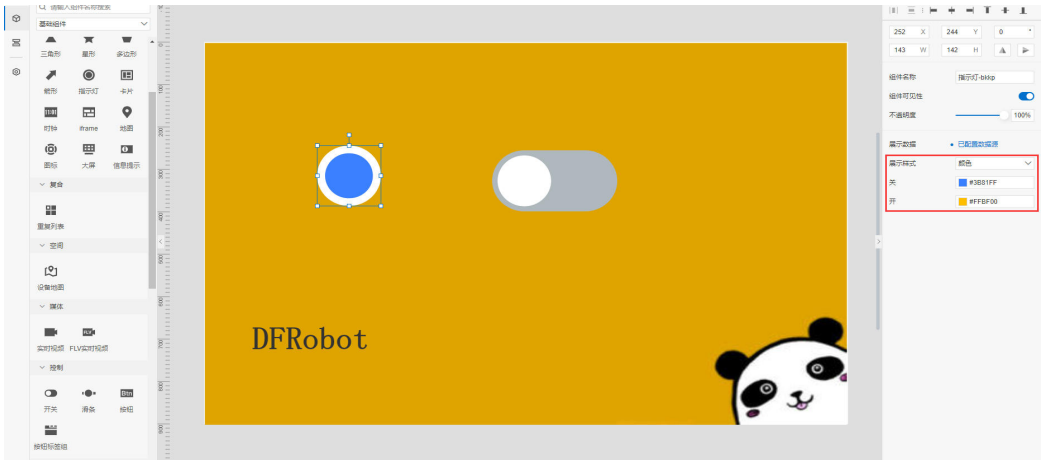


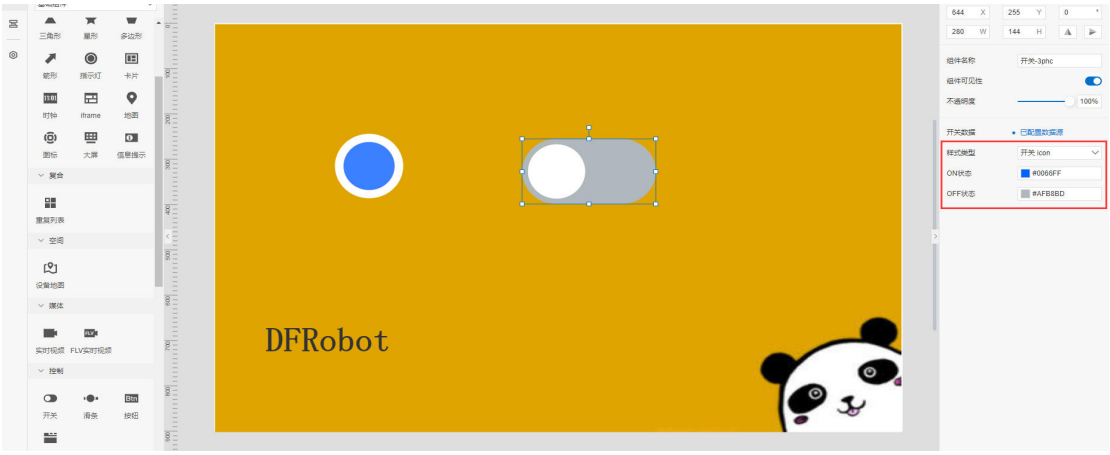


添加开关。将**开关**拖到画布中，然后**配置参数**，产品配置为 LED 灯，设备配置为刚才在项目中创建的设备,属性为工作状态，点击**确定**。



将指示灯和开关的**颜色**进行修改，符合背景图的颜色。配置完成后如下第二张图。

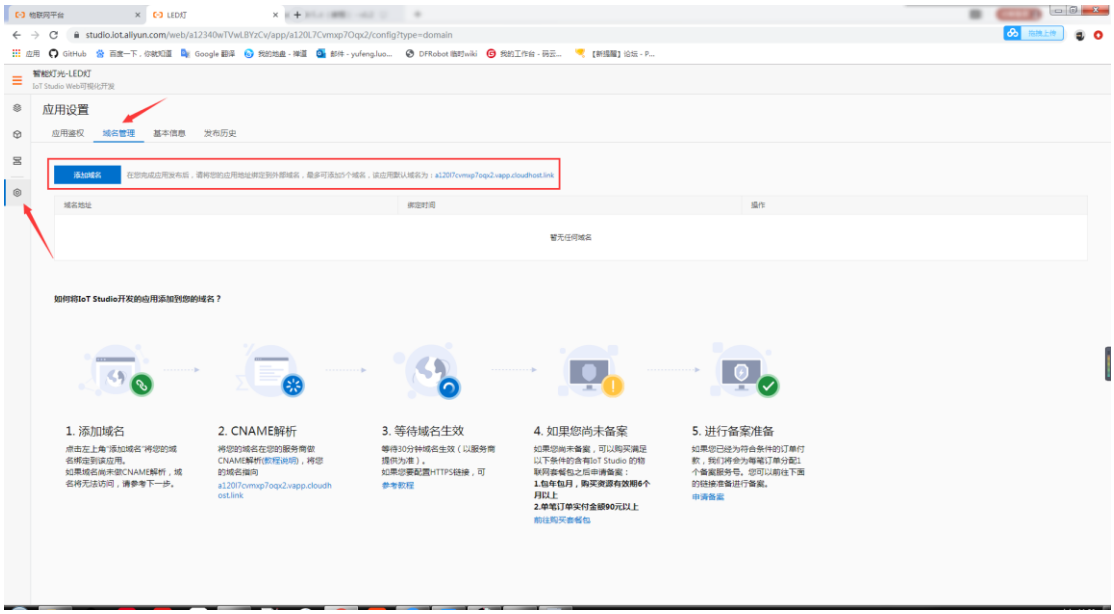




可用账户设置（可选）

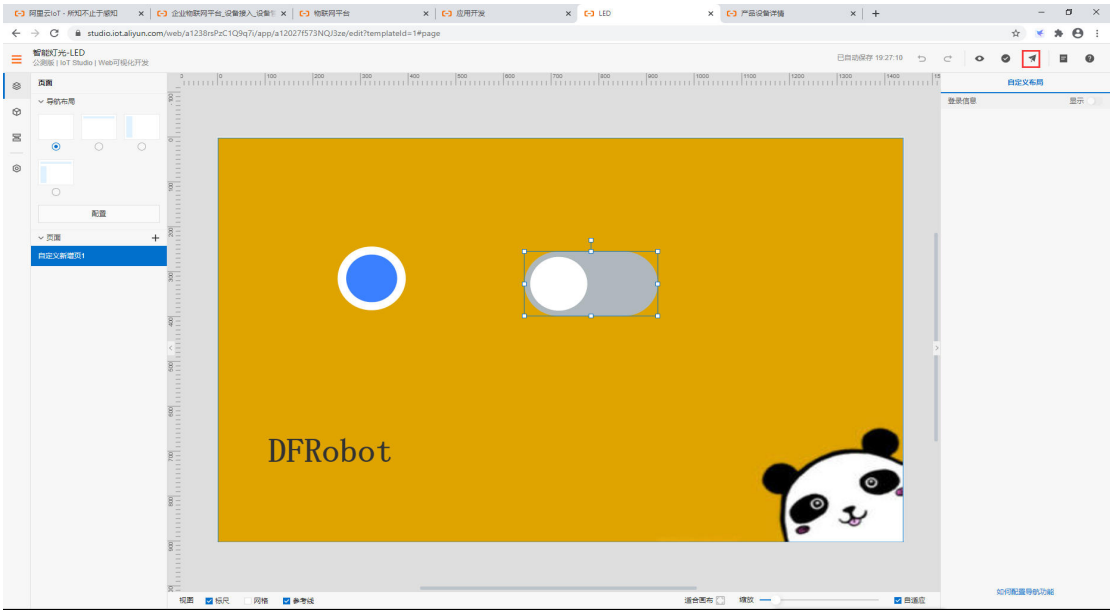
Web 可视化如果要移植使用，需要绑定域名，为了安全需要，绑定域名使用阿里云设置了

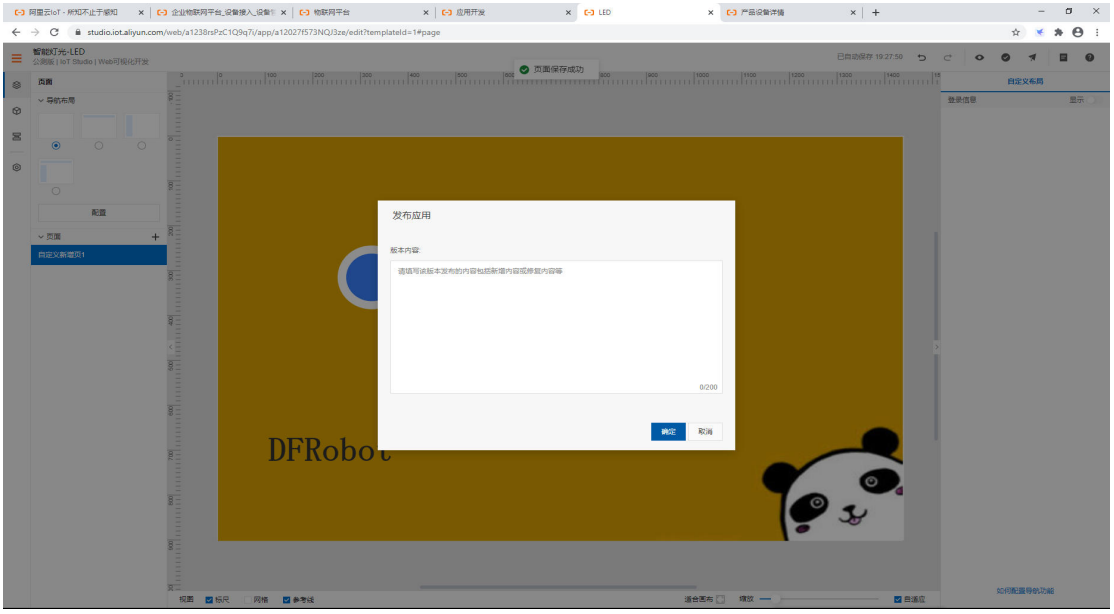
付费。如果需要移植使用的，可参考**设置->域名管理**，**CDN** 付费原则。如果只是个人需要不必添加域名，默认系统的域名即可。

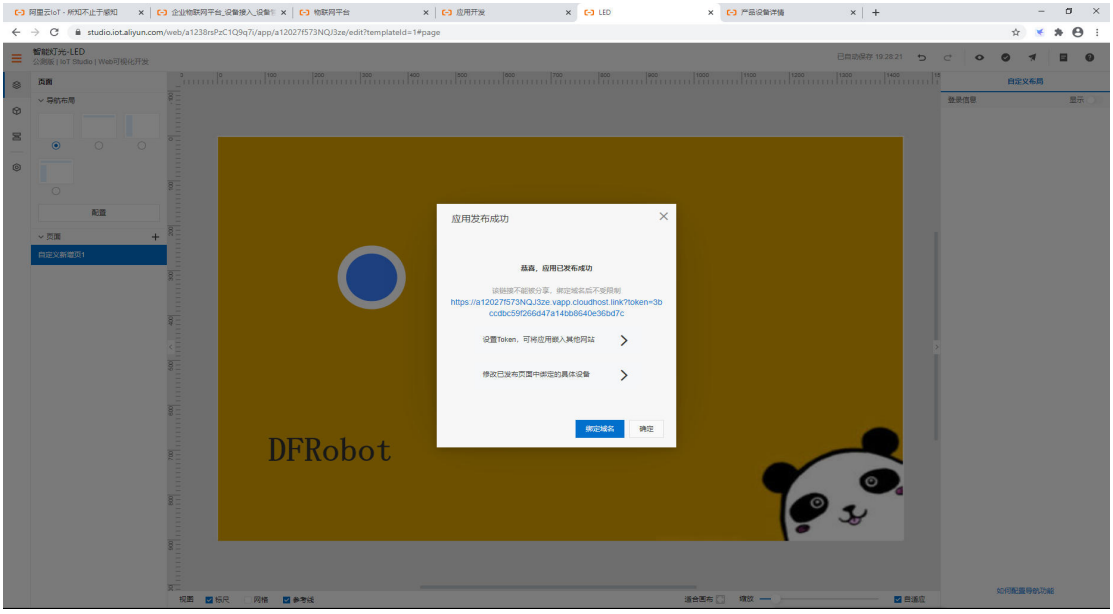


Web 可视化发布

点击右上角的**发布**选项，然后跳到第二张图，备注网页信息，**不用填写**，直接点击**确定**，然后得到网页链接。注意：如果不需要绑定域名，直接点击网页链接，不要点击其他选择。

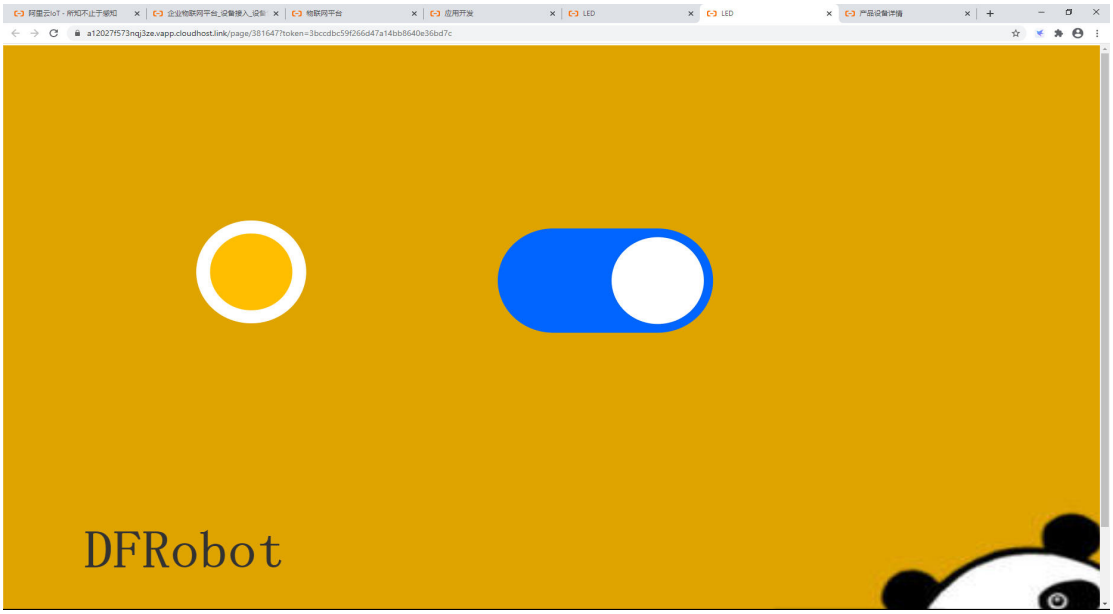






Web 查看

Web 的页面已经生成，可直接在网页上点击开关按钮，实物 LED 就会执行相应的命令。



至此，一个 Web 可视化开发就完成了。

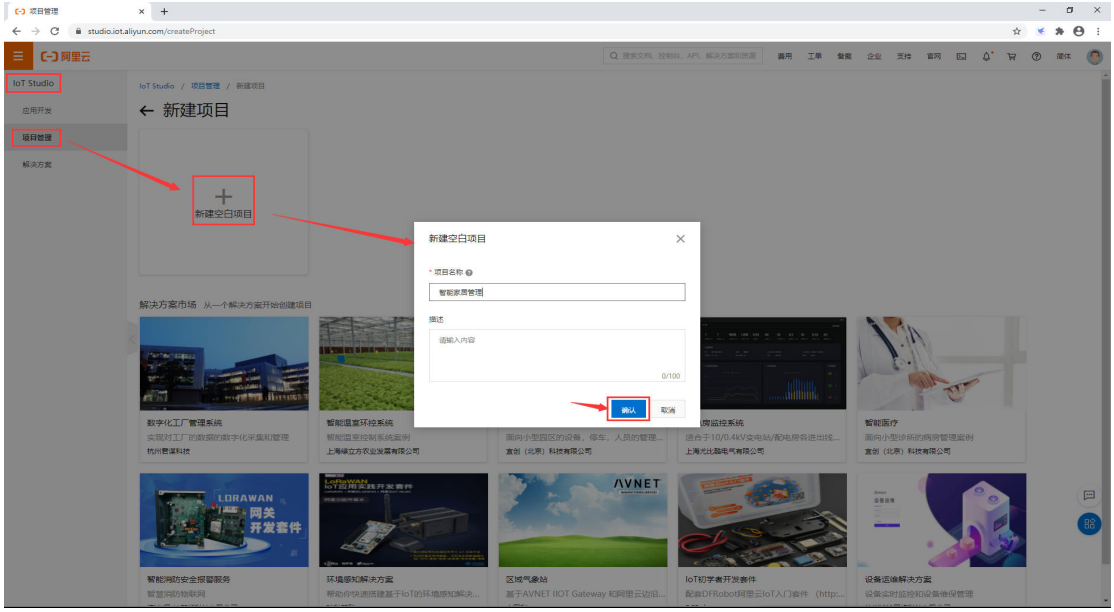
第三章 IOT 应用

准备工作

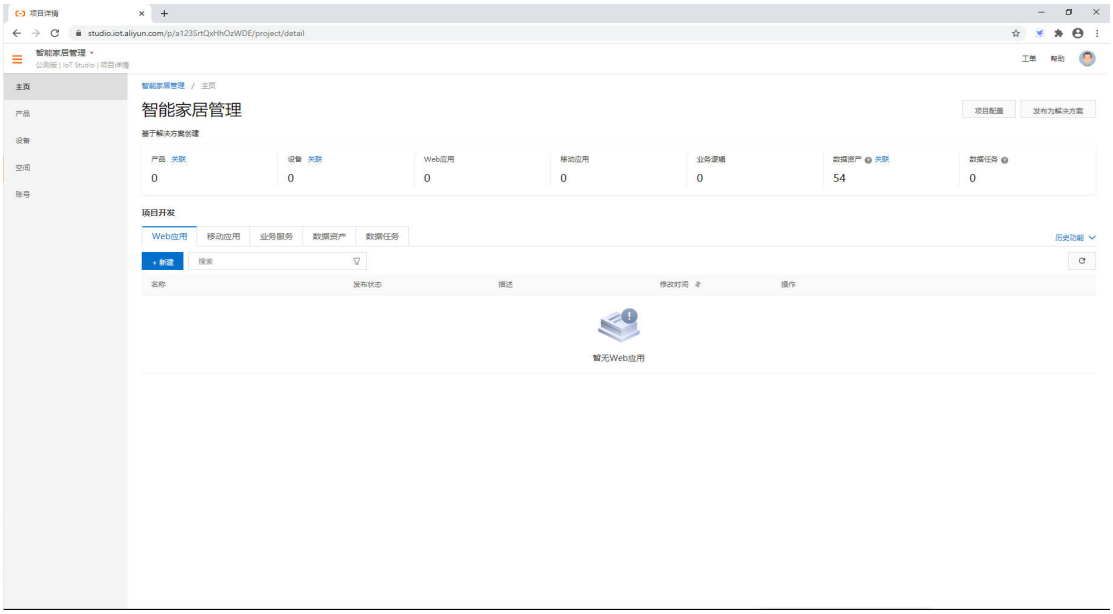
在应用开始之前，我们需要在物联网平台上创建一个开发服务的项——**智能家居管理**，来管理产品，设备，和服务。

创建一个名为“**智能家居管理**”的项目。

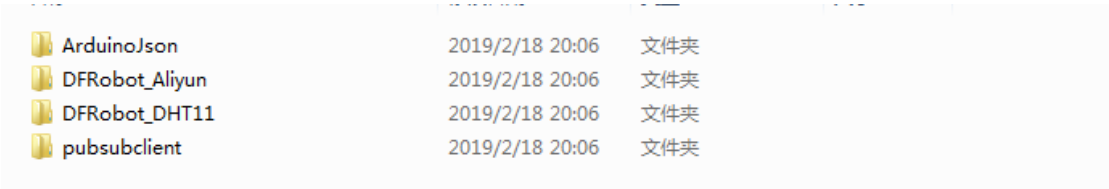
登录并进入物联网平台控制台在左侧导航栏选择 **IoT Studio**，进入项目管理页面，单击**新建项目**，在项目名称框编辑智能家居管理，填写信息，如下图所示:



点击**确定**，若项目创建成功，会自动弹出项目管理页面。在这里，我们可以管理此后应用中的所有产品，设备，和服务。“智能家居管理”项目创建好了，我们可以在该项目下开发有趣味的物联网项目啦！



此外，我们还需要把以下**库文件**（如下图所示）存放到 Arduino IDE 安装目录下的 libraries 文件中，如下图所示



DFRobot_Aliyun 文件中存放了下面 8 个应用的程序例程，我们只需要修改程序其中的某些内容即可编译上传代码至 ESP32。这些内容包括：

● WiFi 名称和密码的配置；

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

●设备证书信息配置；

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

●产品标识符配置

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

●上报和订阅 TOPIC 配置

```
/*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
```

这些配置信息中，除了 WiFi 名和密码需要用户连接自己的 WiFi，其他的配置信息都需要从阿里云 IoT 平台上获取。

好了，接下来就让我们制作有趣的应用项目吧！

3.1 应用一：智能灯光

在对 Arduino 平台和阿里云 IoT 平台有了基础的了解，我们就可以开始制作我们的第一

个应用了——智能灯光。

应用目标：通过 Web 可视化或者手机 App 随时随地监测、控制 LED 灯的工作状态。

智能灯光的核心就是能随时随地实现对灯的控制与监测。比如，你晚上起来用手机一键打开过道上的灯，又比如你外出后，不确定是否关掉卧室的灯，此时你可以打开手机监测卧室灯的状态，若显示灯是开着的状态，则关灯。

那么我们如何制作该应用呢？

所需元件

红色食人鱼 LED 发光模块 x1

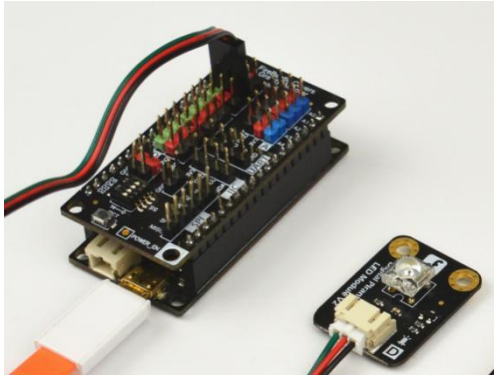
FireBettle Board-ESP32 x1

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，注意扩展板不要插反，有白色丝印的那个角要对齐。再把红色 LED 模块接在扩展板的 D2 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。

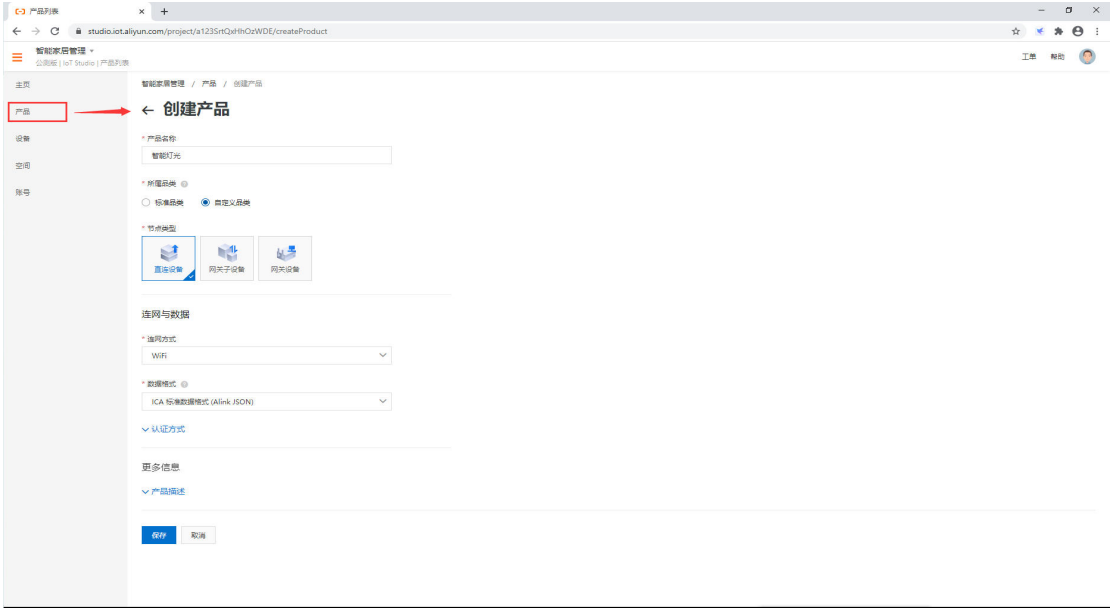


元件连接好后，使用 USB 线连接 FireBettle Board-ESP32 主板和电脑。

硬件准备好后，接下来就需要我们在阿里云平台上创建相应的产品，设备，服务来建立物理设备和阿里云 IoT 平台之间的联系了。

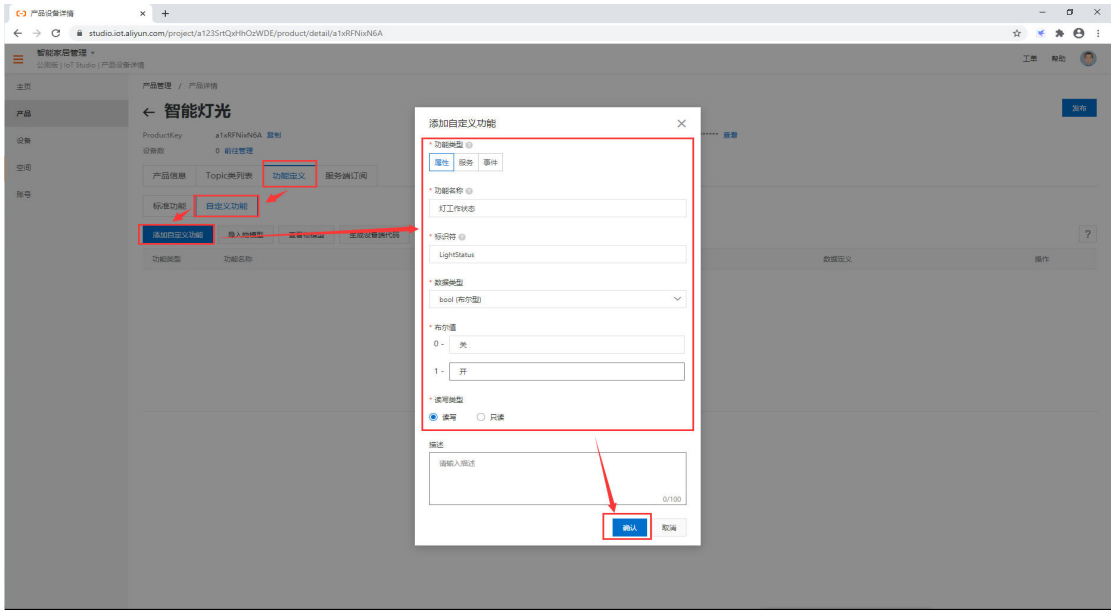
制作步骤

产品创建。在我们之前创建的智能灯光项目下创建名为**智能灯光**的产品，如下图所示：



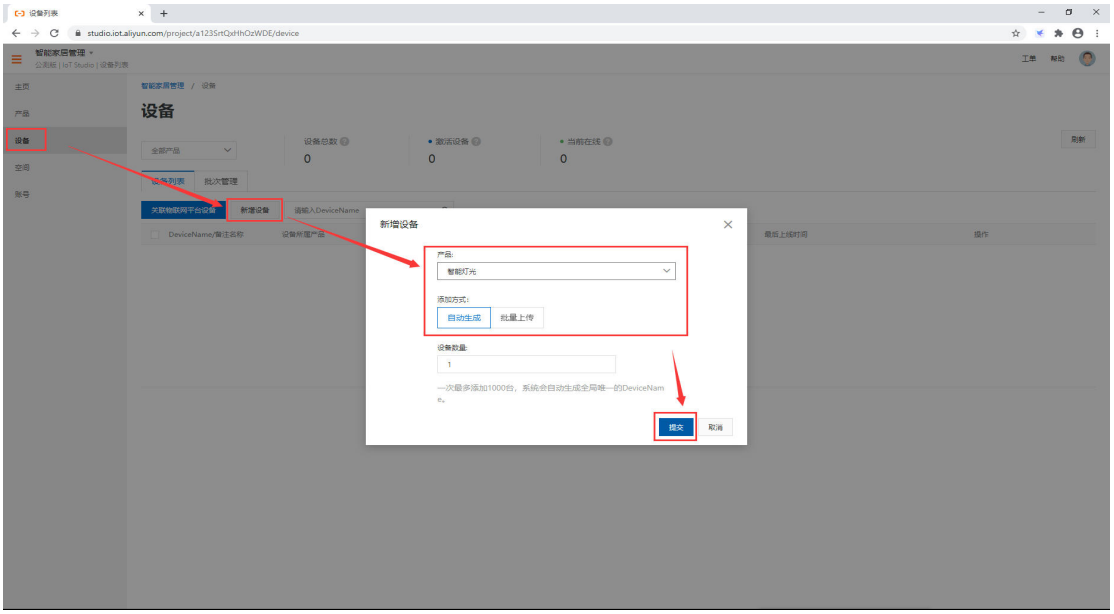
点击**保存**，则会在产品列表出现一个名叫智能灯光的产品，点击**查看**。

定义产品功能。选择**功能定义**后再点击**添加自定义功能**为智能灯光产品定义一个**灯工作状态**的属性，灯有两种工作状态：亮灭，可用布尔型表示，其中 0 代表灯灭，1 代表灯亮。如下图所示：

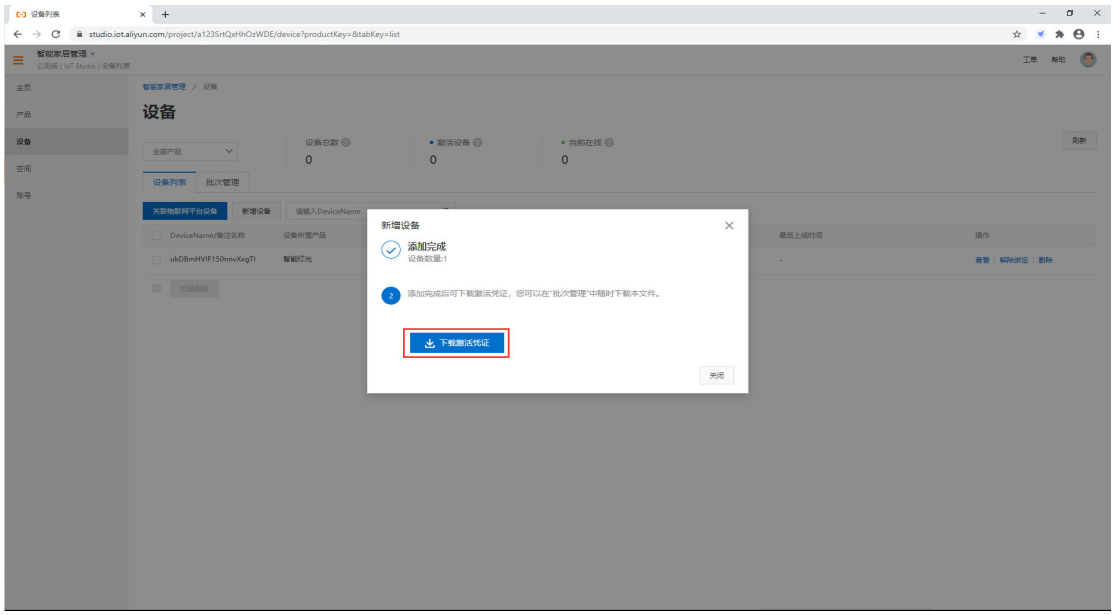


点击**确认**，则会在产品详情页的自定义功能栏看到所定义的功能及标识符。

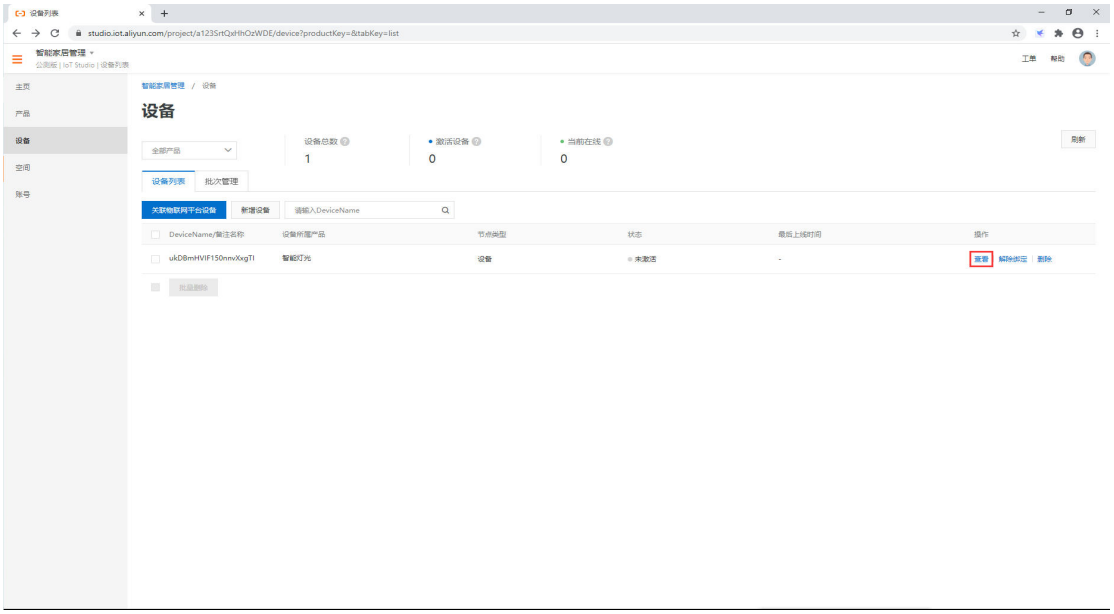
为产品添加设备。为智能灯光生成一个设备，如下图所示：



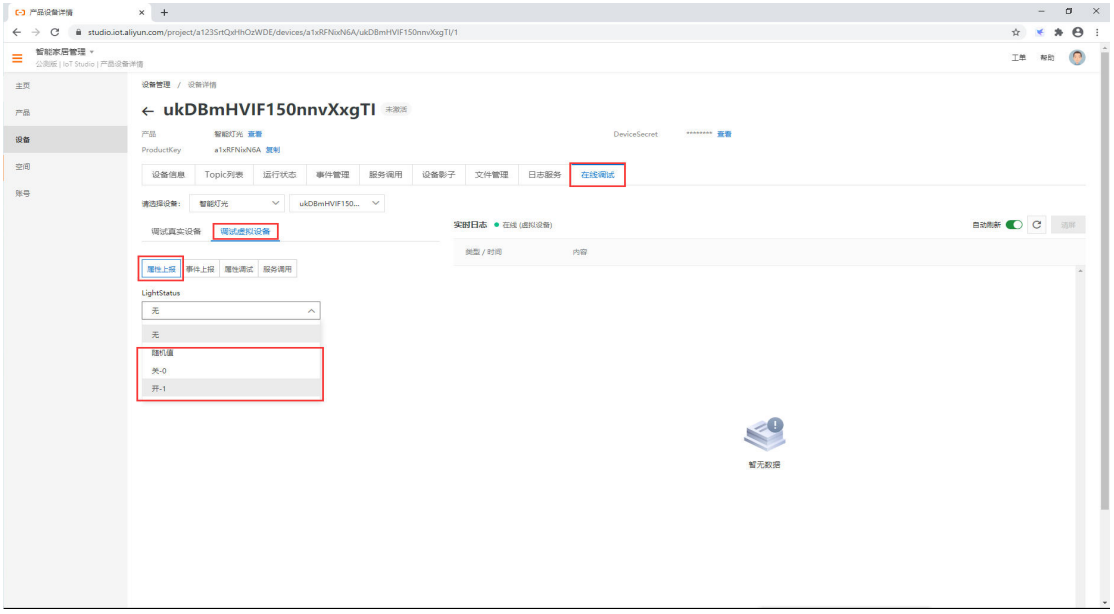
点击**提交**，会弹出设备激活凭证下载，点击下载（激活凭证是我们连接阿里云的关键参数），关闭后，该设备会自动出现在设备列表中。由于某个产品中的设备会继承该产品的所有属性，故设备具有灭、亮的工作状态属性了。



在线调试参数，当设备生成成功后我们可以选择在线调试参数。这一步的目的主要是为了生成移动应用时可以配置数据。点击查看，进入到设备详情页。再点击在线调试，进入到调试页面。



按照下图所示的方式进行调试推送即可。

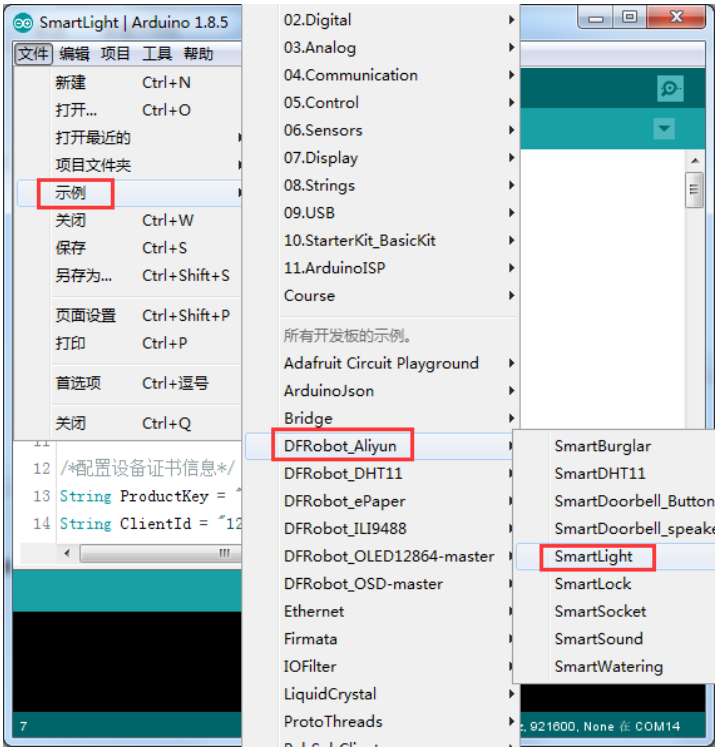


将下拉框的所有选项均推送一次。

进行到此步，即可开始 Arduino 程序的修改，完成真实物理设备同阿里云 IoT 平台的连接。

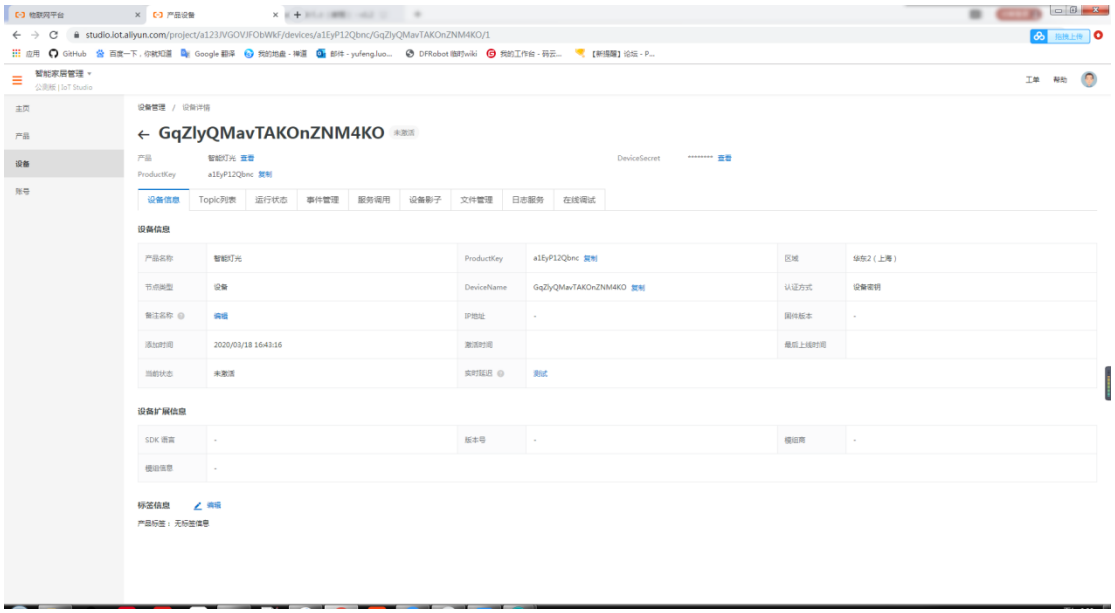
Arduino 程序修改。

打开 Arduino IDE，点击**文件->示例->DFRobot_Aliyun->SmartLight** 项目，如下图所示



修改程序，WiFi 名及密码，直接将其修改成能使用的 WiFi 即可，就不多说了，这里主要介绍如何获取产品标识符、设备证书信息(ProductKey、DeviceName、DeviceSecret)、及订阅和上报 TOPIC。

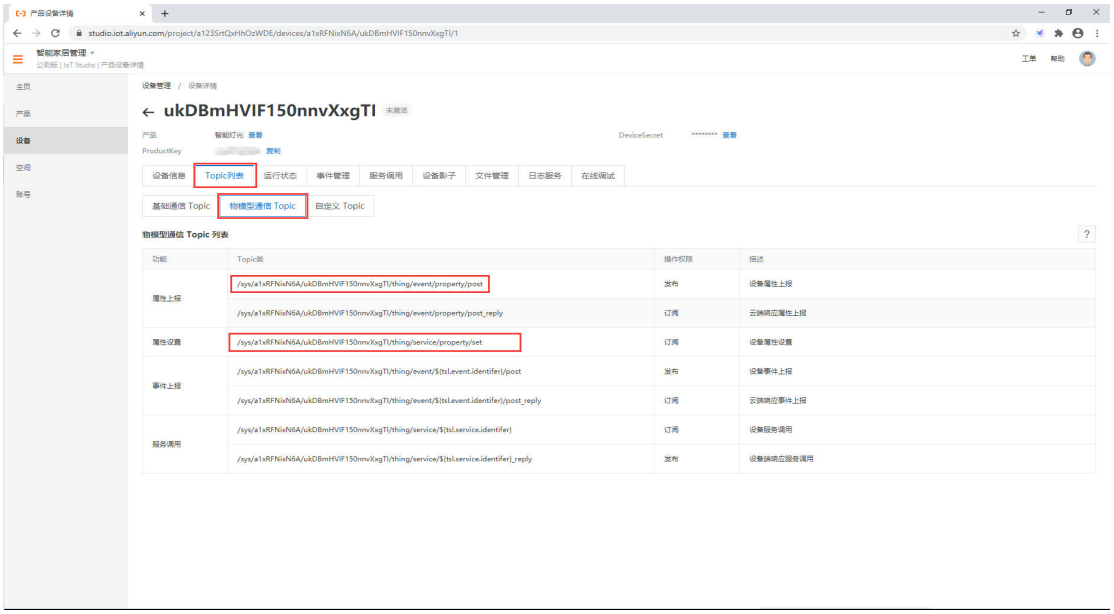
设备信息证书。进入智能家居管理项目，在左侧导航栏选择**设备管理->设备**，在设备列表中找到我们刚才创建的设备，单击其后的**查看**，即可看到该设备的设备证书信息，如下图所示：



将该设备证书信息分别复制粘贴到 SmartLight 程序的 ProductKey、DeviceName、DeviceSecret 变量值的双引号内，如下图所示：

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

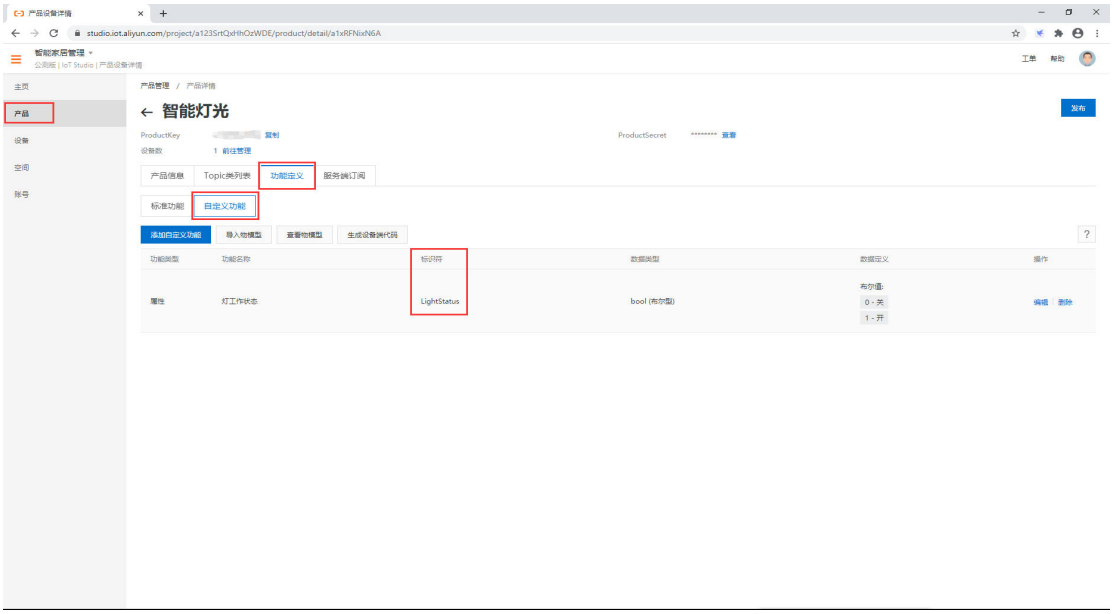
设备上报和订阅信息。点击查看，进入 Light 设备详情页，点击 Topic 列表，即可看到设备的发布和订阅信息，如下图所示：



将设备的发布和订阅信息复制粘贴到 Arduino 程序中的相应位置，如下图所示：

```
/*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
```

产品标识符。在左侧导航栏选择**产品**，在设备列表中找到名为**智能灯光**的产品，单击**查看**，进入产品详情页，单击**功能定义**，即可在**自定义功能栏**查看该产品的**标识符**，如下图所示：



复制粘贴该产品标识符到程序代码相应位置，如下图所示：

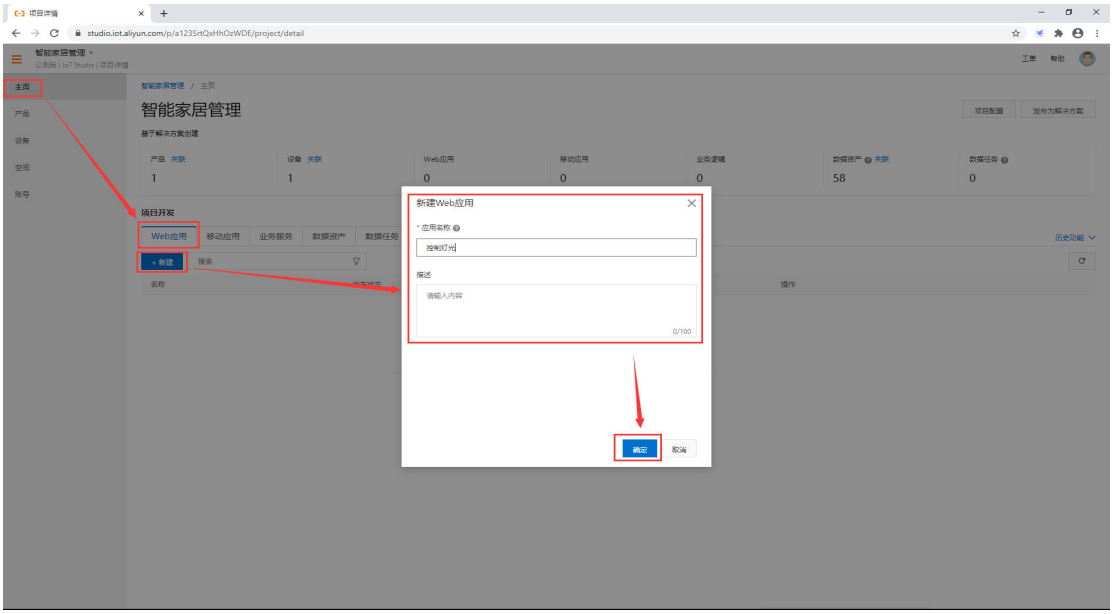
```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

至此，智能灯光的 Arduino 程序修改完毕，此时编译上传至 ESP32 即可。执行端已经准备完毕。

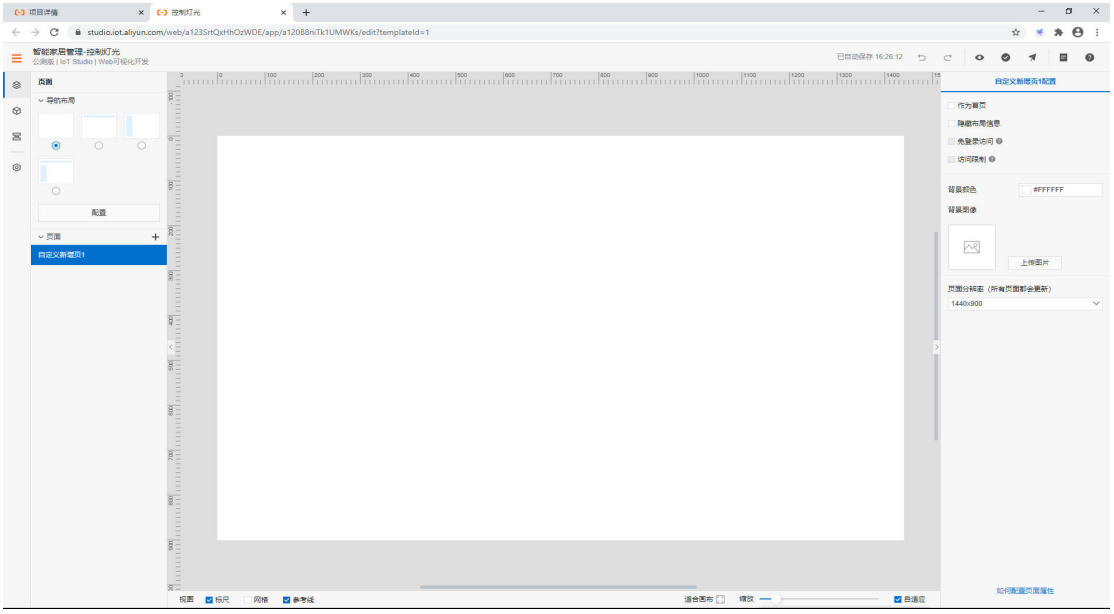
现在来介绍如何创建一个 **Web 可视化开发**来实现对 LED 灯的控制和监测了。

首先是 **Web 可视化开发的应用**

如下图所示在**项目管理界面**中点击**主页**然后点击**新建**。输入名称后点击**确认**即可。

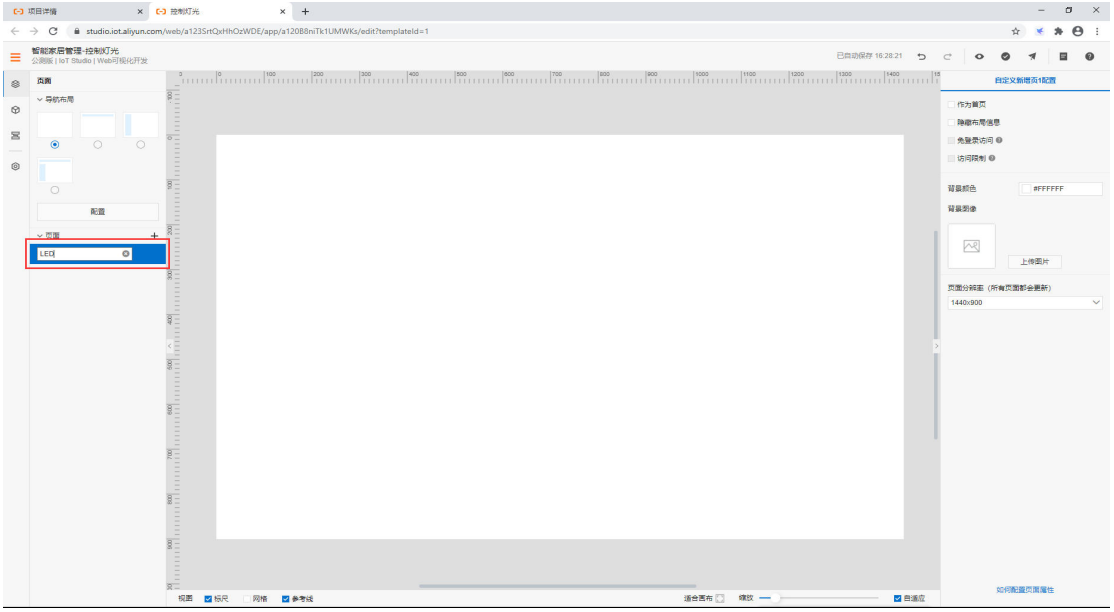


Web 页面编辑



下面就进入 Web 的页面编辑

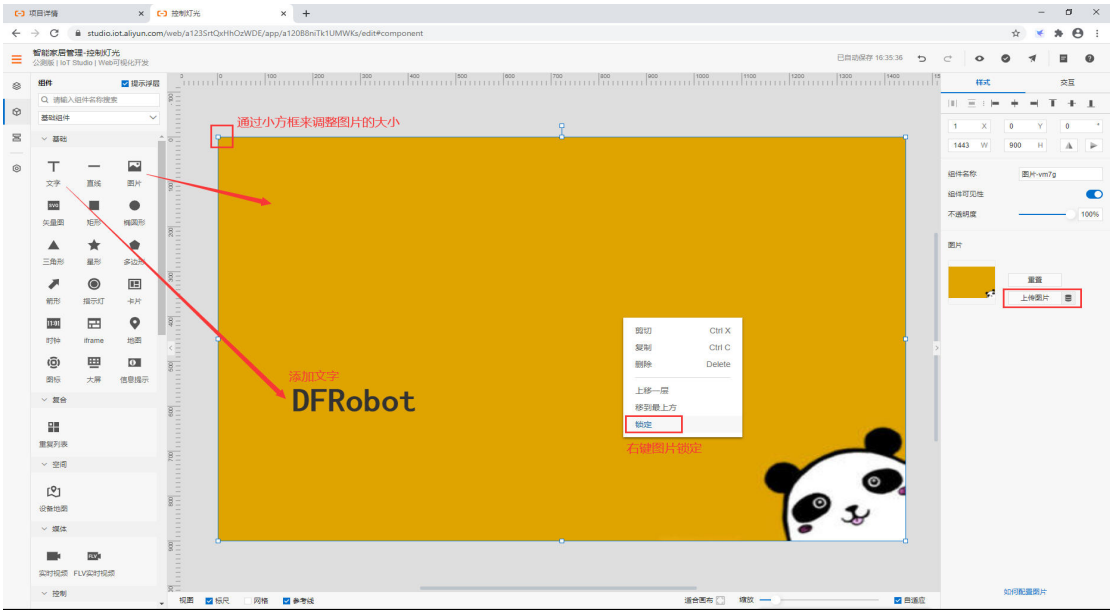
Step1:点击配置，然后将自定义新增页 1 改为 LED



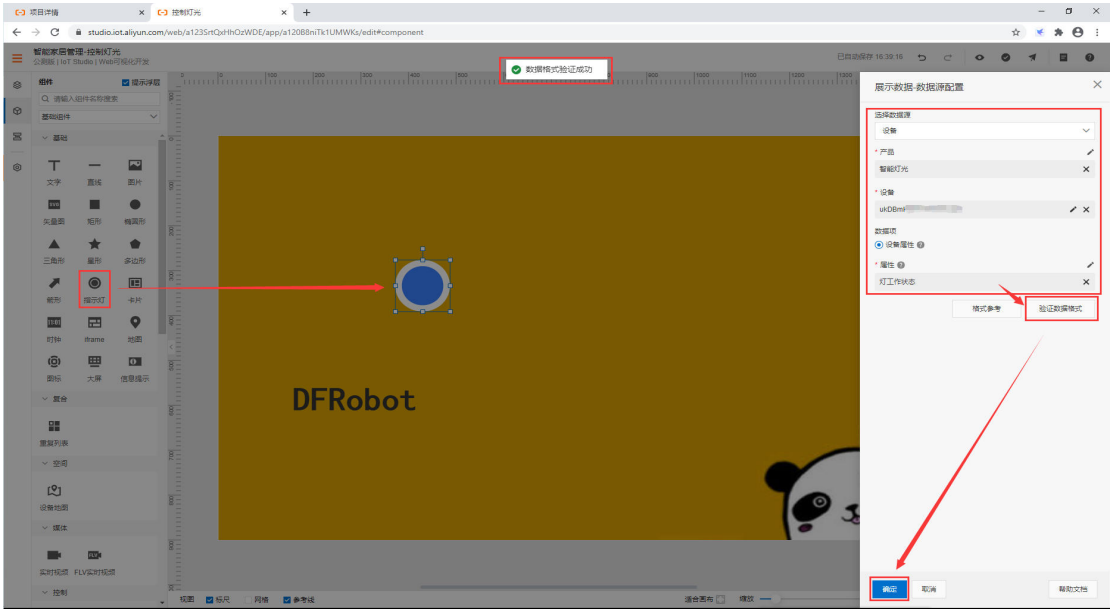
Step2:添加组件。

为了美观，我们放上一张背景图，所以先添加一张**图片**，将图片拖到画布中，设置为**锁定**。然后上传一张比较好看的图片。背景图原图如下。

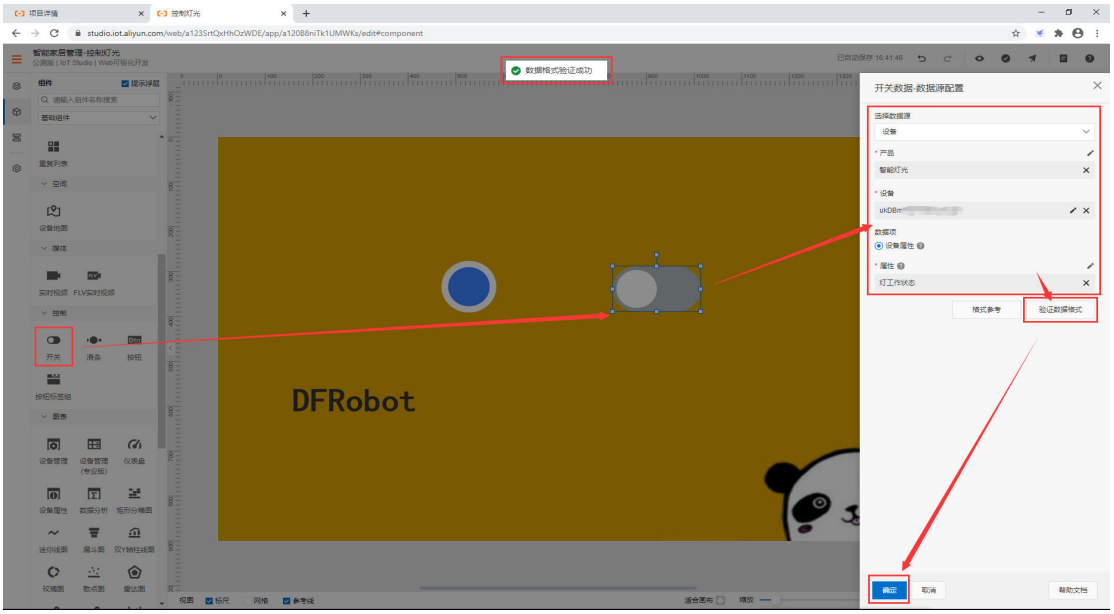
添加文字。将**文字**拖到画布中，改为 **DFRobot**，可以将字体颜色改为贴合背景色的颜色。



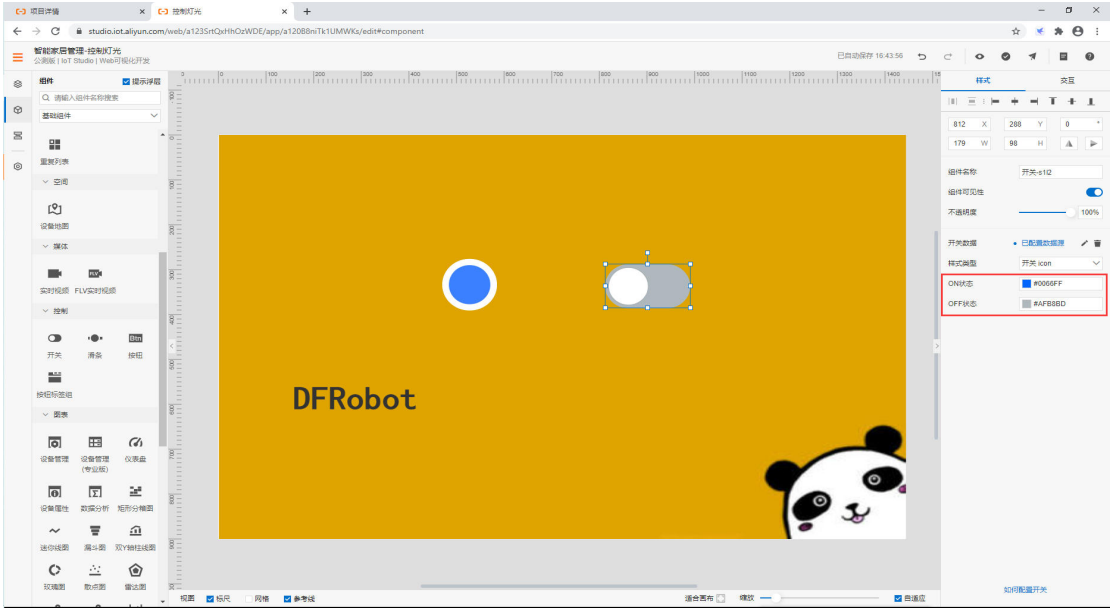
添加指示灯。将**指示灯**拖到画布中，然后**配置参数**，产品配置为智能灯光,设备配置为刚才创建的设备 ,属性为灯工作状态，点击**验证数据格式**，再点击**确定**，这就配置完成了.



添加开关。将**开关**拖到画布中，然后**配置参数**，产品配置为智能灯光,设备配置为之前创建的设备,属性为灯工作状态，点击**验证数据格式**，再点击**确定**。

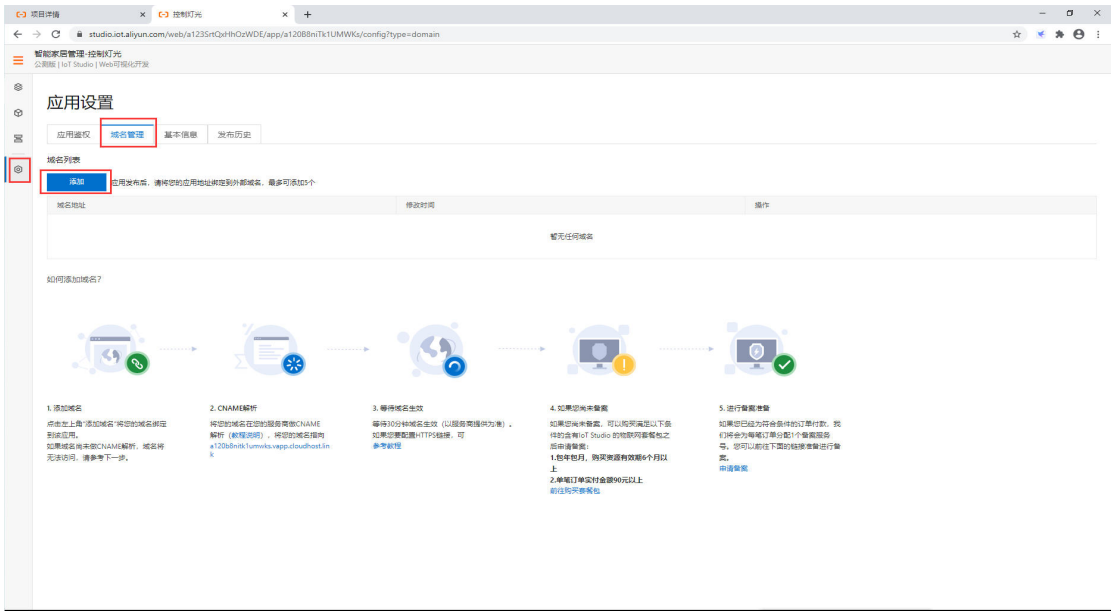


将指示灯和开关的**颜色**进行修改，符合背景图的颜色。配置完成后如下第二张图。



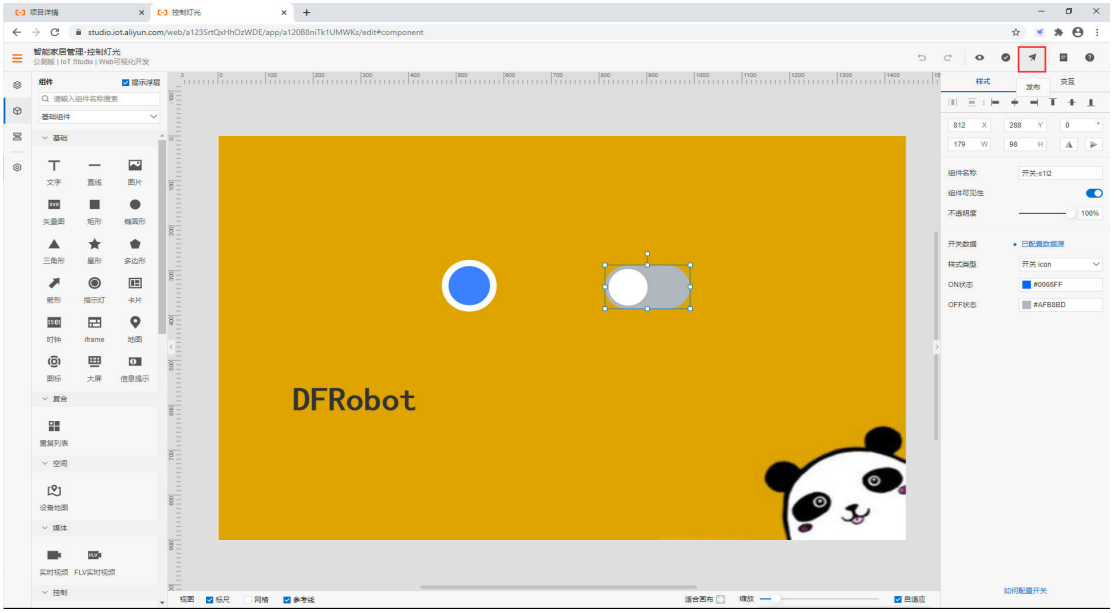
可用账户设置（可选）

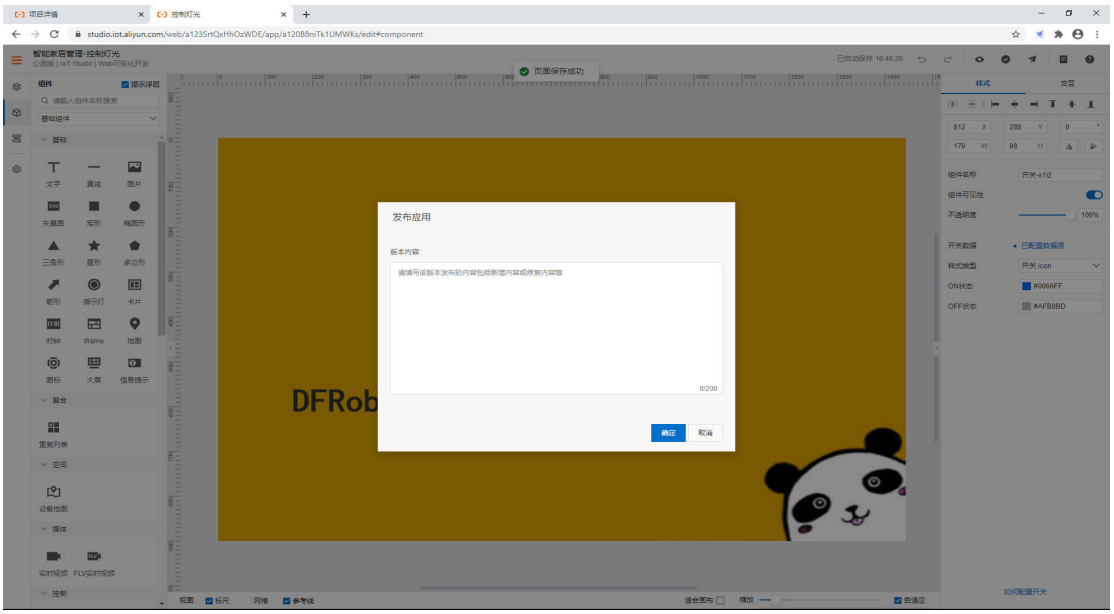
Web 可视化如果要移植使用，需要绑定域名，为了安全需要，绑定域名使用阿里云设置了付费。如果需要移植使用的，可参考**应用设置->域名管理**，**CDN** 付费原则。如果只是个人需要不必添加域名，默认系统的域名即可。

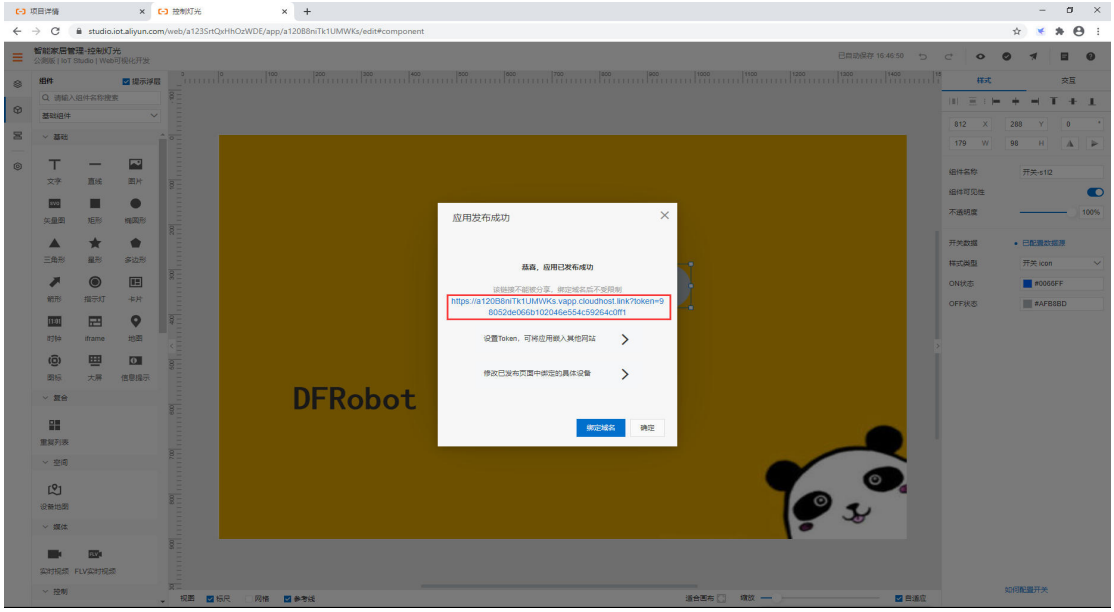


Web 可视化发布

点击右上角的**发布**选项，然后跳到第二张图，备注网页信息，**不用填写**，直接点击**确定**，然后得到网页链接。注意：如果不需要绑定域名，直接点击网页链接，不要点击其他选择。


DFRobot


发布应用
确定
取消



Web 查看

Web 的页面已经生成，可直接在网页上点击开关按钮，实物 LED 就会执行相应的命令。

3.2 应用二：智能插座

通过前一小节的学习，我们大概了解了物联网项目的移动应用开发流程。而这个项目同上个项目在原理上大同小异。

应用目标：通过手机 App 控制继电器的开关。

所需元件

继电器模块 x1

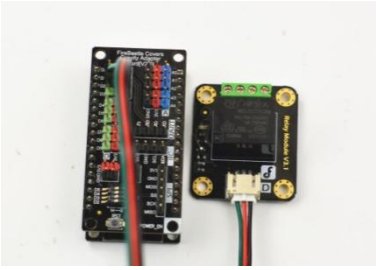
FireBettle Board-ESP32 x1

FireBettle Gravity 扩展板 x1

智能手机 x1

硬件连接

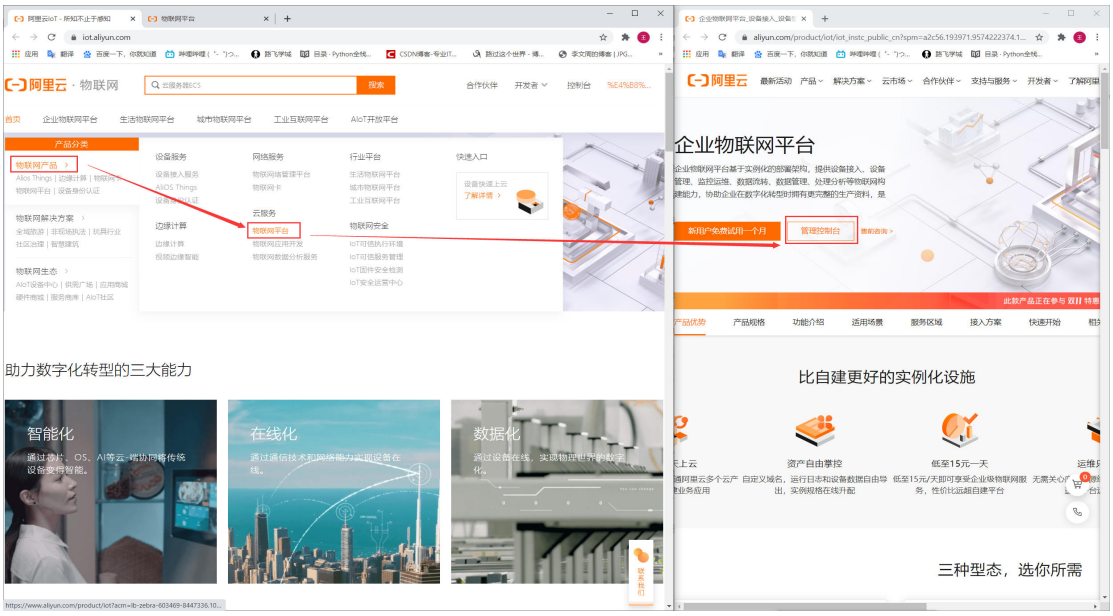
把 FireBettle Gravity 扩展板直插在 FireBettle Board-ESP32 主板上，再把数字继电器模块接在扩展板的 D2 数字引脚上，确保电源引脚、接地引脚和信号引脚都连接正确。



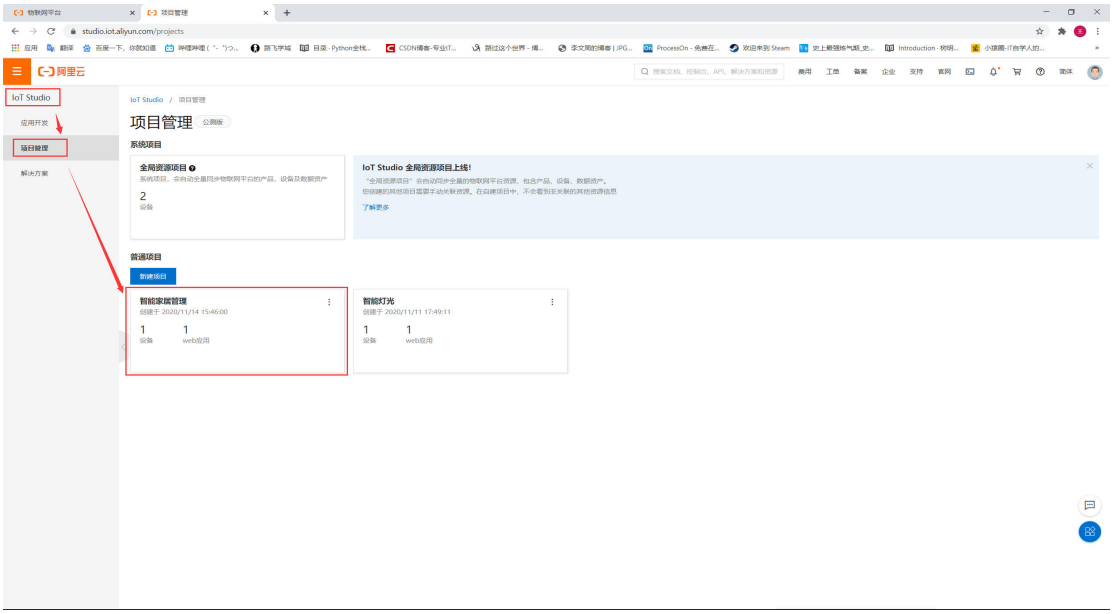
应用分析：通过对智能灯光项目的总结，可以知道，DFRobot 和阿里云 IoT 平台已经为物联网项目开发的连接部分完成了，用户只需在阿里云平台上完成产品、设备、服务的创建和配置，并获取他所创建的产品的标识符、设备证书、设备订阅或上报 Topic 信息，并将它在程序中做出相应的修改，即可搭建物联网项目。

制作步骤

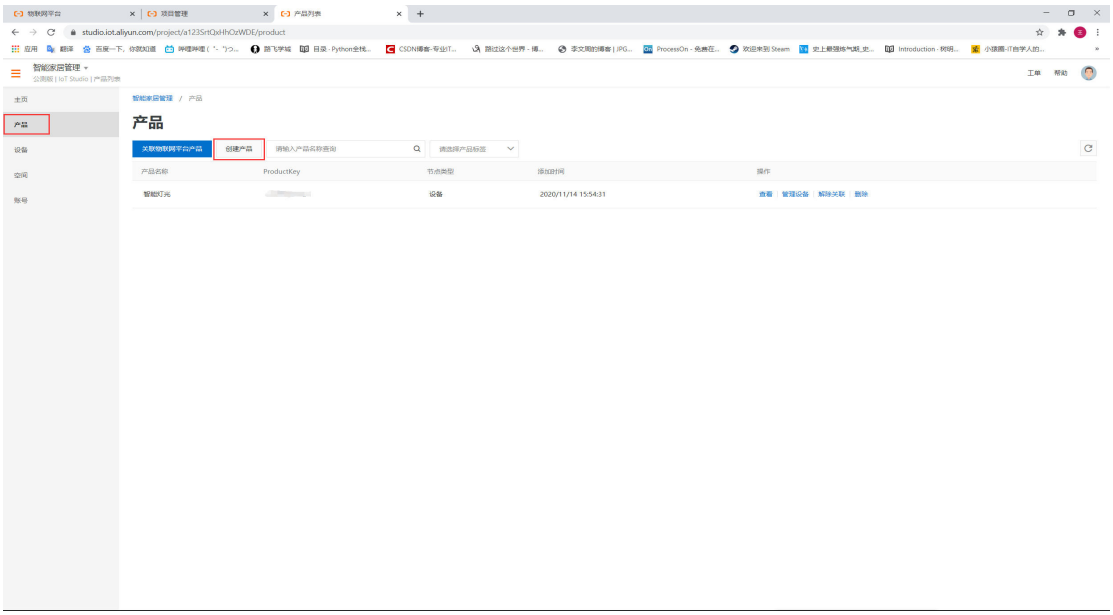
1.打开浏览器，输入网址 https://iot.aliyun.com ，登录并进入**物联网平台控制台**；

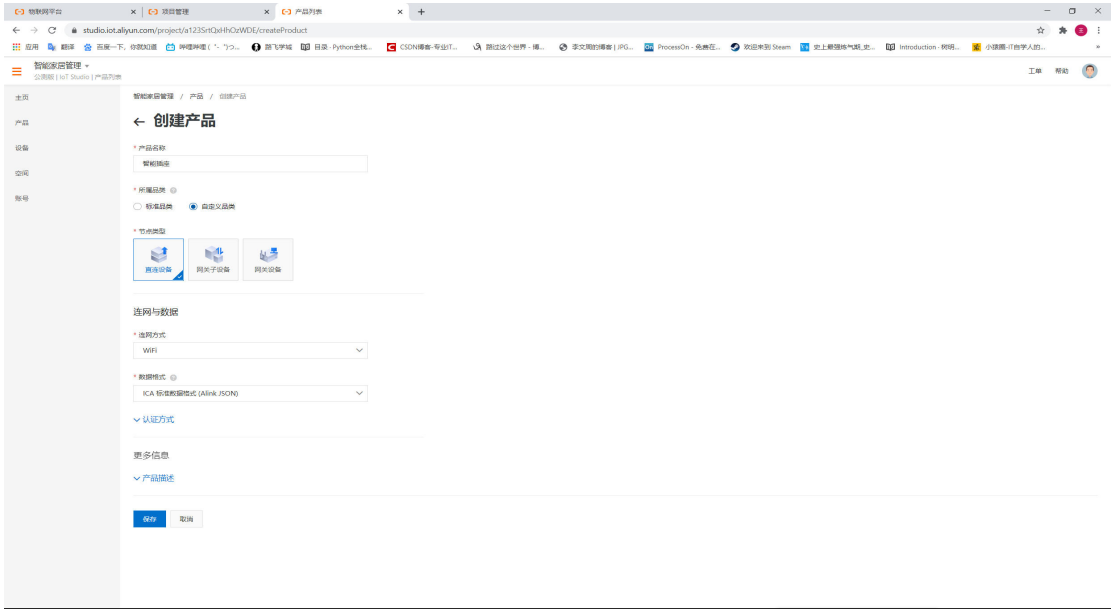


2.在左侧边栏的 **相关产品** 下点击 **IoT Studio** ，进入 IoT Studio 页面，点击 **项目管理** 选项，然后选择我们之前的智能家居项目。

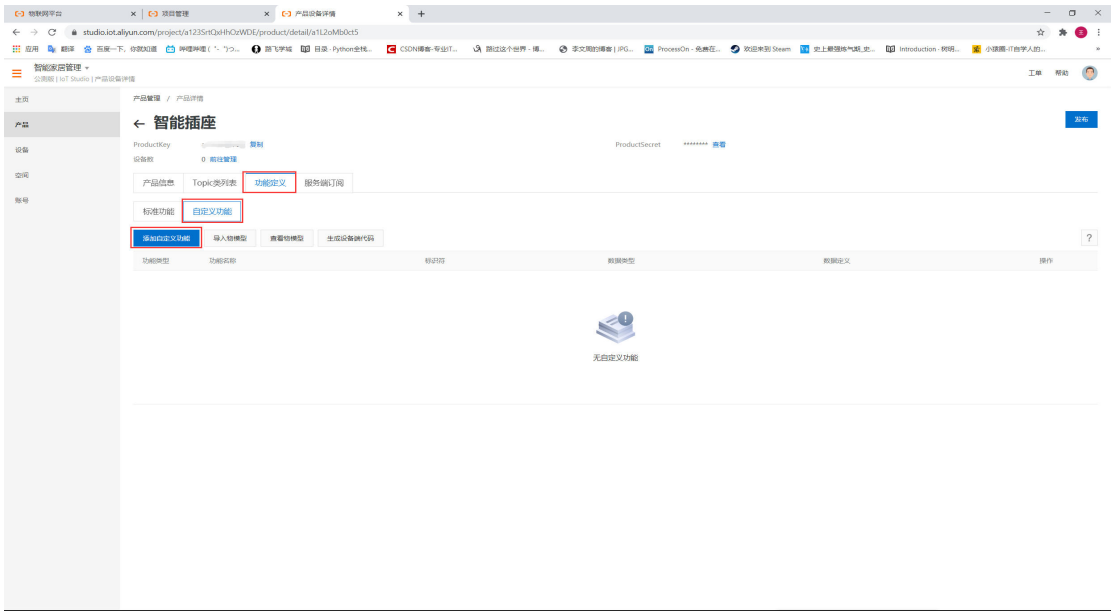


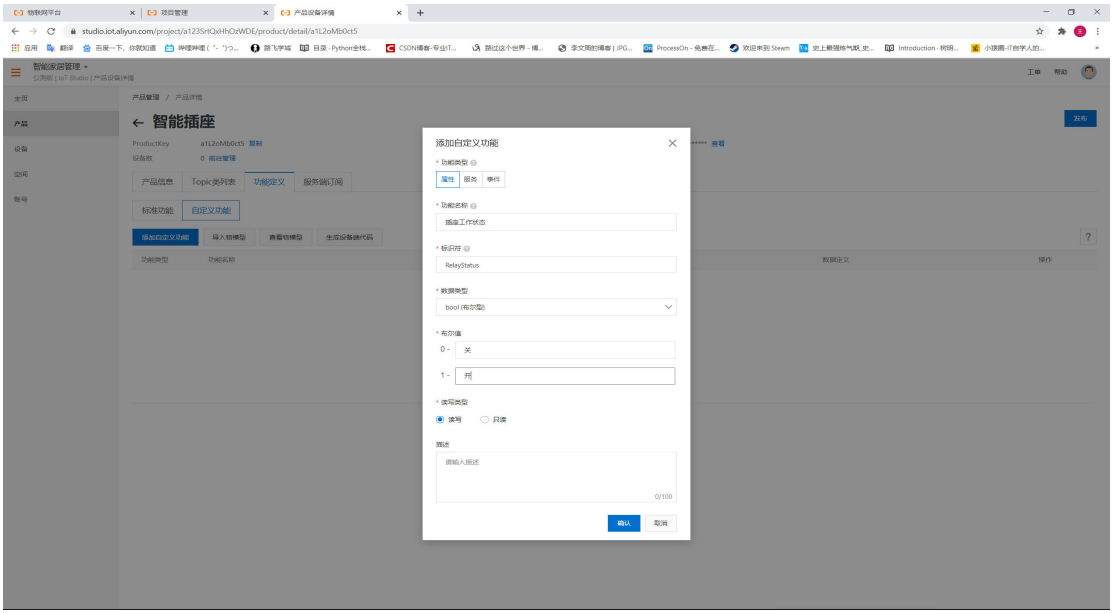
3，在该项目下创建一个产品。点击**产品**，选择**新建产品**，如下图所示创建一个对应的产品。



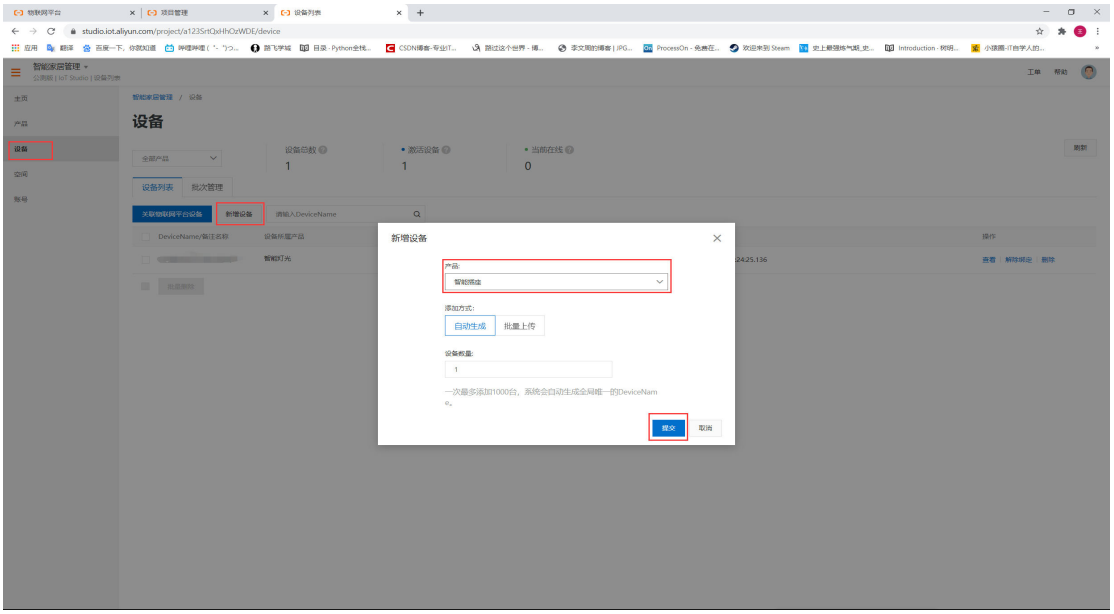


4，点击**查看**，为产品添加功能。选择**产品**，点击**功能定义**。再点击**添加功能**。定义一个功能名为“**插座工作状态**”的属性，自定义一个产品标识符 RelayStatus，插座工作状态有 2 种：开，关，定义 0 为关，1 为开。如下图所示：



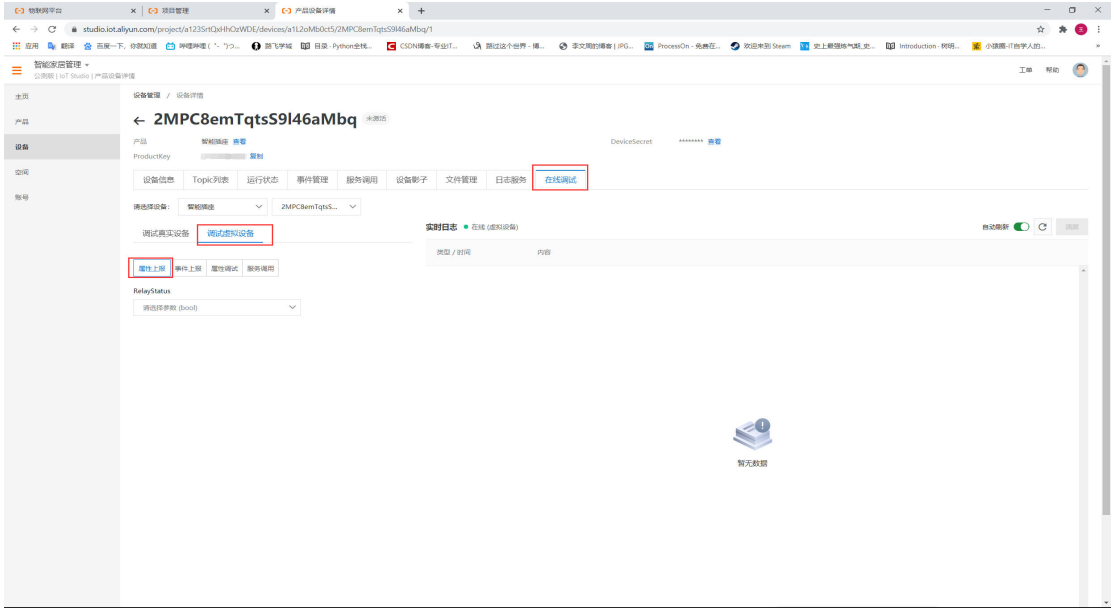


5，为**智能插座**添加一个设备，如下图所示，点击确定。下载好**激活凭证**后（激活凭证是我们连接到阿里云的关键参数），点击**关闭**。



6， 设备创建成功后，点击产品进行在线调试。点击**设备**，再点击**查看**，再选择**在线调试**。

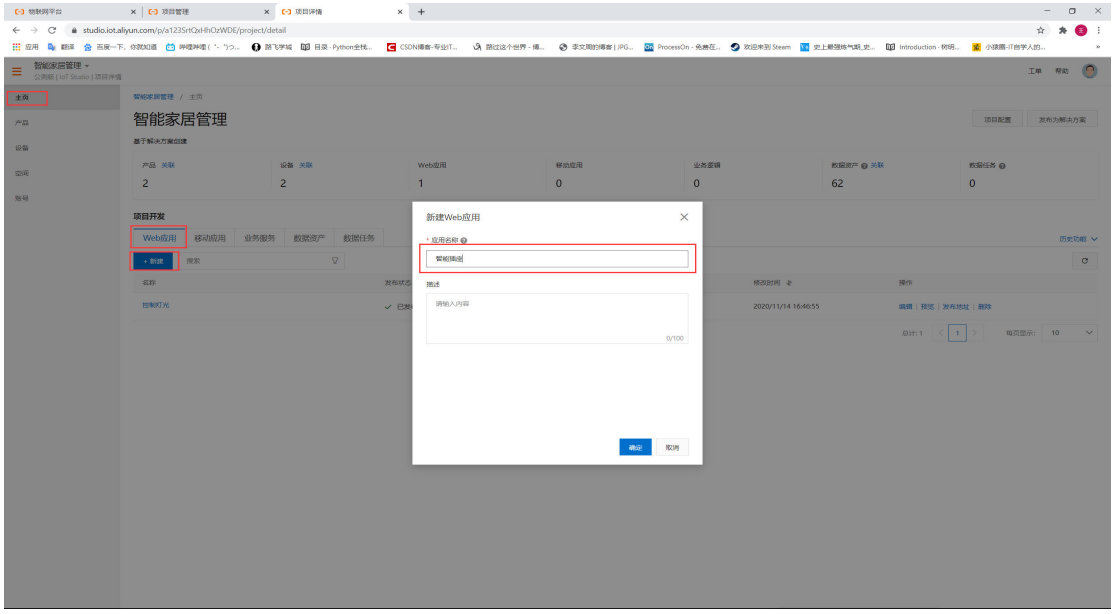
按照下图所示操作



将下拉框的所有选项均推送一次：

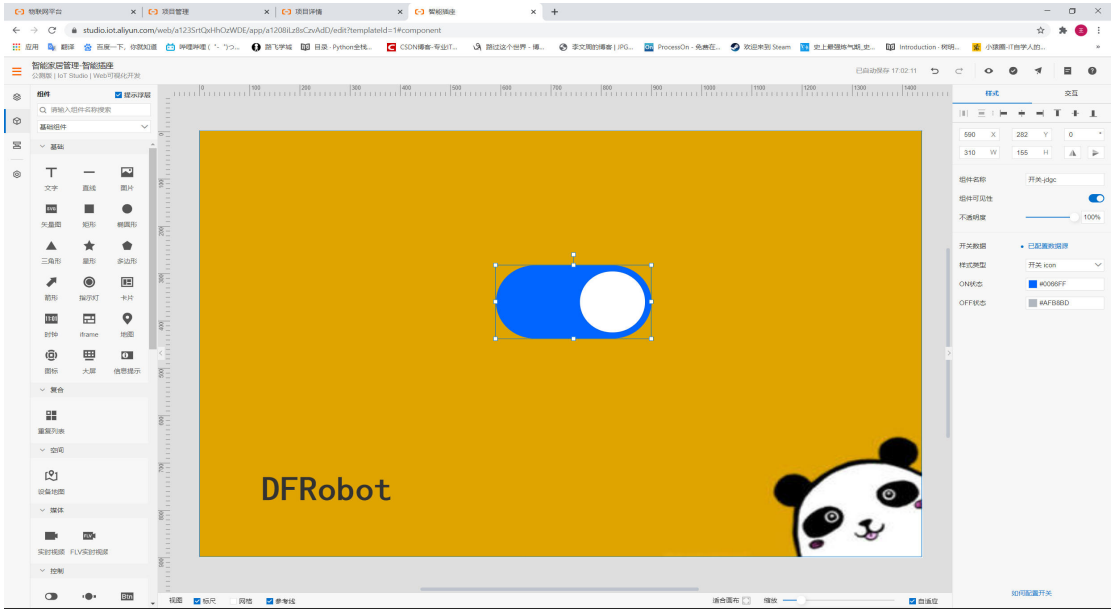
7.创建 Web 可视化开发

如下图所示在**项目管理界面**中点击 **Web 应用**然后点击**新建**。输入名称后点击**确认**即可。



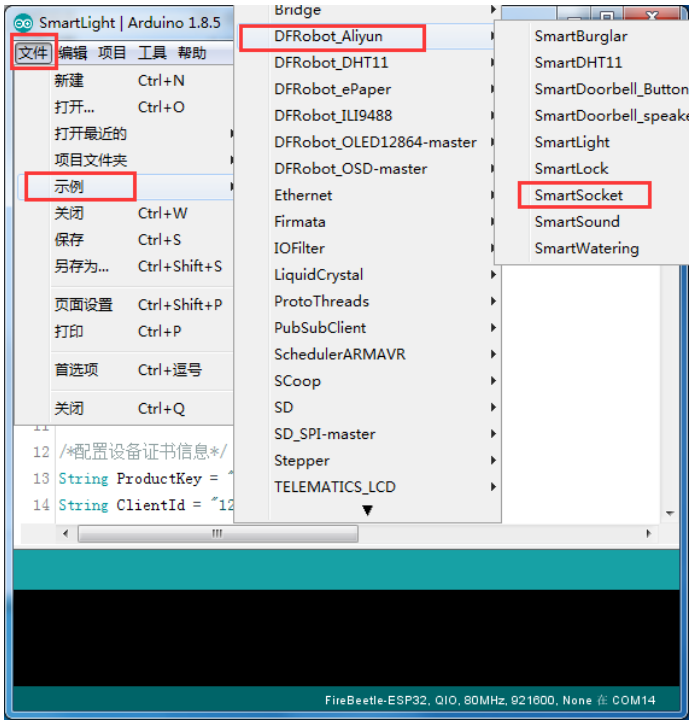
跟应用一 一样的方法，添加图片背景和文字，这里不做多介绍

添加开关。将**开关**拖到画布中，然后**配置参数**，产品配置为智能插座,设备配置为之前创建的设备,属性为灯工作状态，点击**验证数据格式**，再点击**确定**。



点击保存，发布。Web 可视化就完成了。

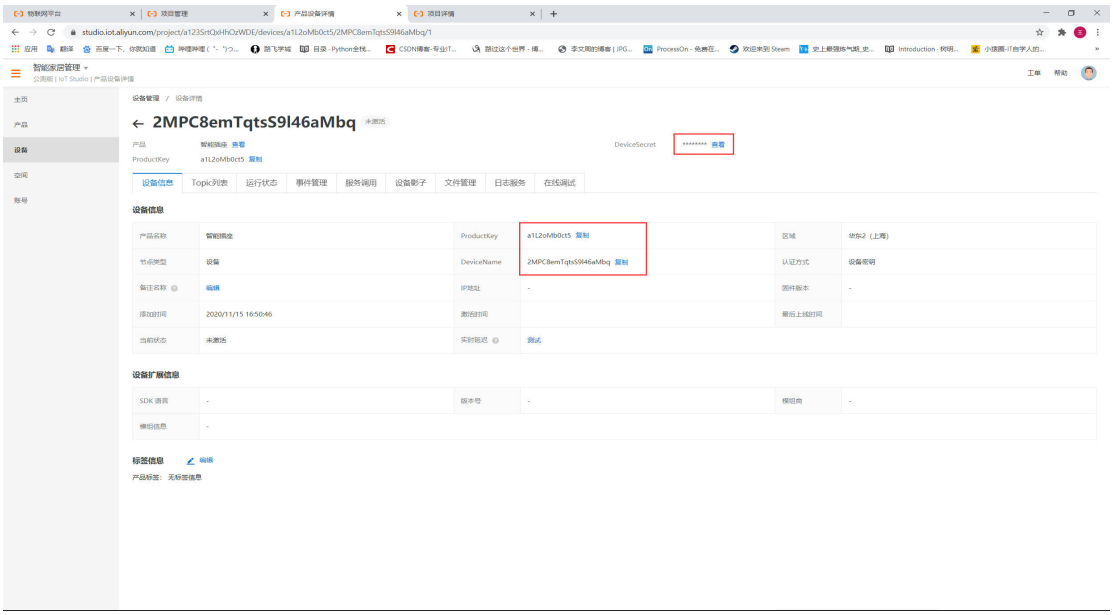
7.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartSocket** 项目，如下图所示:



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";
```

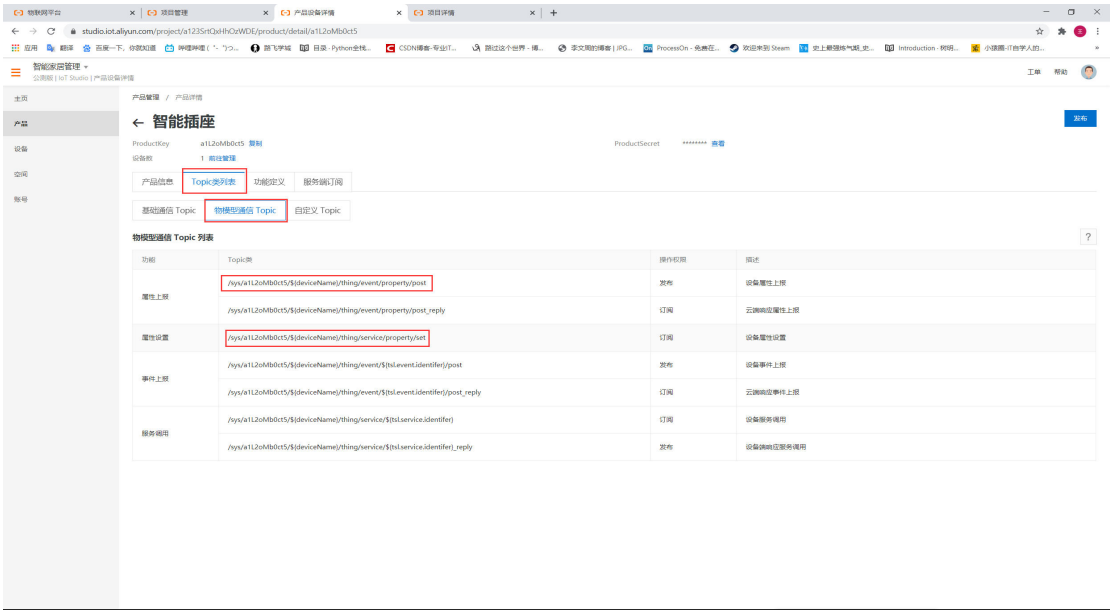
② 填写设备证书信息。获取智能插座设备的设备信息，如下图所示：



并将获取到的**智能插座的**设备信息，复制粘贴替换到如下图所示的程序模块中：

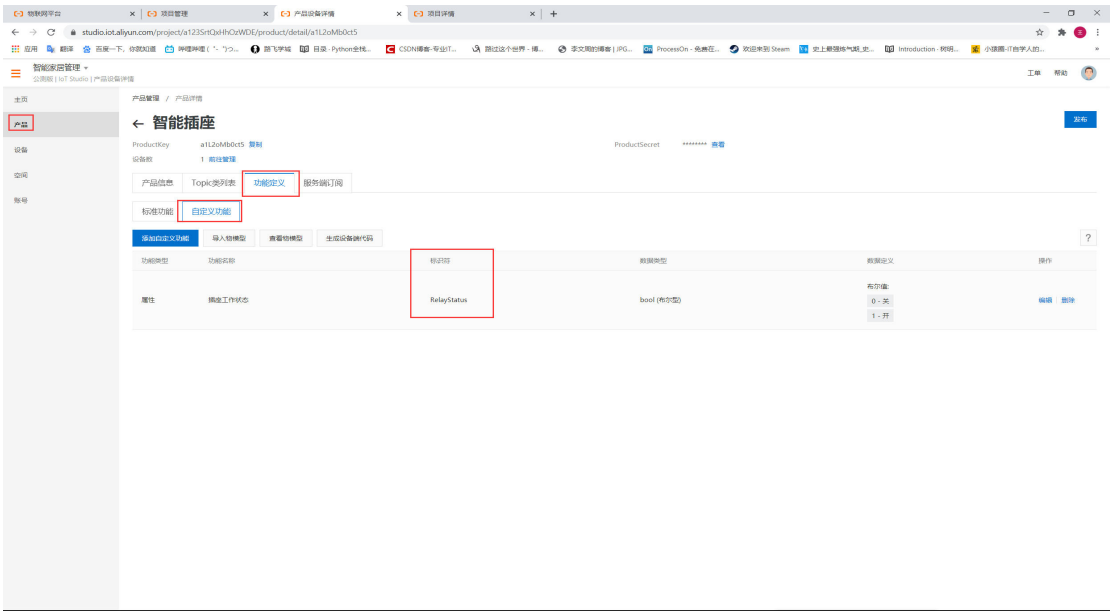
```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

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插座的产品，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。

此时设备即成功与 Aliyun 平台上的 Relay 设备关联，打开下载的智能家居控制台，选择智能插座，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）。

3.3 应用三：温湿度监测

本应用采取 DHT11 温湿度传感器获取环境温度和湿度，并将数据传输给远端 Web 显示，从而实现远程实时监控环境温湿度的功能。

应用目标：通过电脑 Web 实时监测环境温湿度情况。

所需元件

DHT11 温湿度传感器 x1

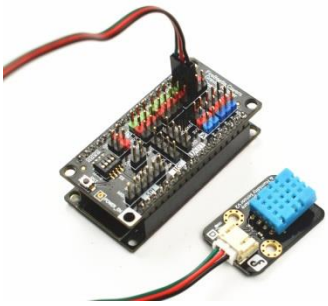
FireBettle Board-ESP32 x1

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 DHT11 温湿度模块接在扩展板的 D2 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。

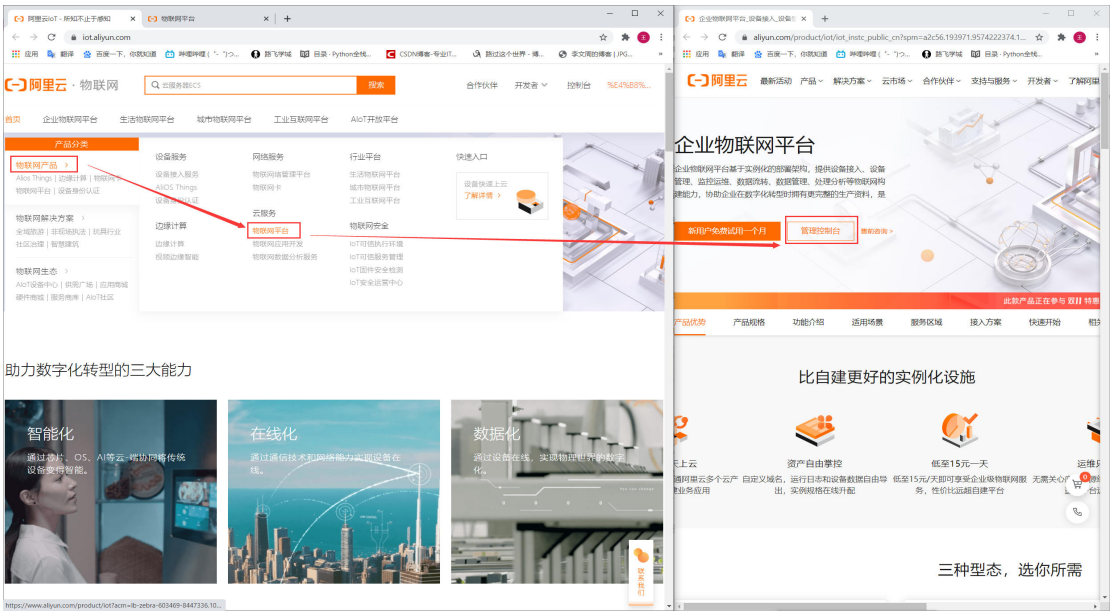


应用分析：通过对上述 2 个应用的学习，我们大概可以了解到，对于温湿度监测而言，我们需要做以下几件事：1.在阿里云 IoT 平台上创建一个名为“温湿度监测”的产品，并为其定义名为“室内温度”和“室内湿度”的功能；2.在“温湿度监测”产品下挂载名为“DHT11”的设备；3.在“智能家居控制台”移动应用项目中添加一个名为“温湿度监测”的 App 页。4.获取设备“DHT11”的设备证书，发布订阅 Topic 列表，以及获取产品的标识符，并将其在 Arduino 程序中修改。

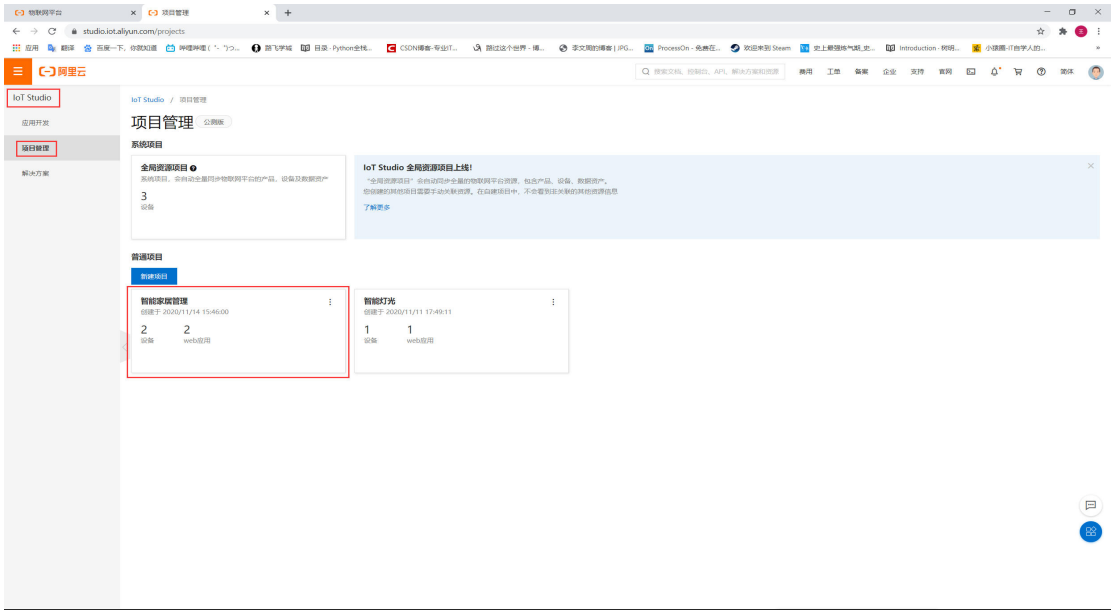
制作步骤

1.打开浏览器，输入网址 https://iot.aliyun.com，登录并进入**物联网平台控制台**；

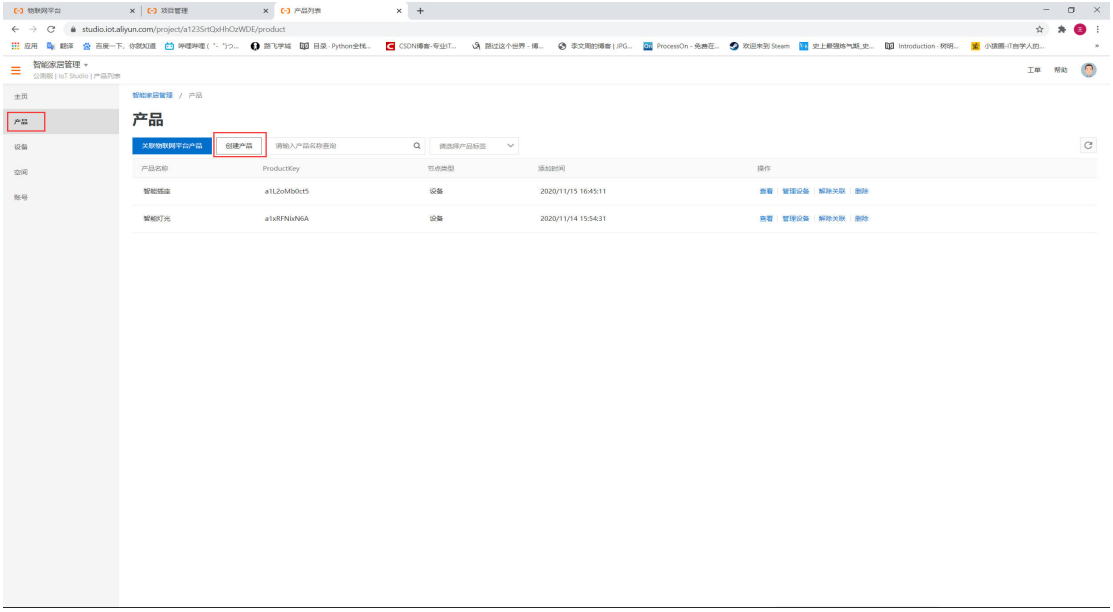
2.在左侧边栏的 **相关产品** 下点击 **IoT Studio** 。

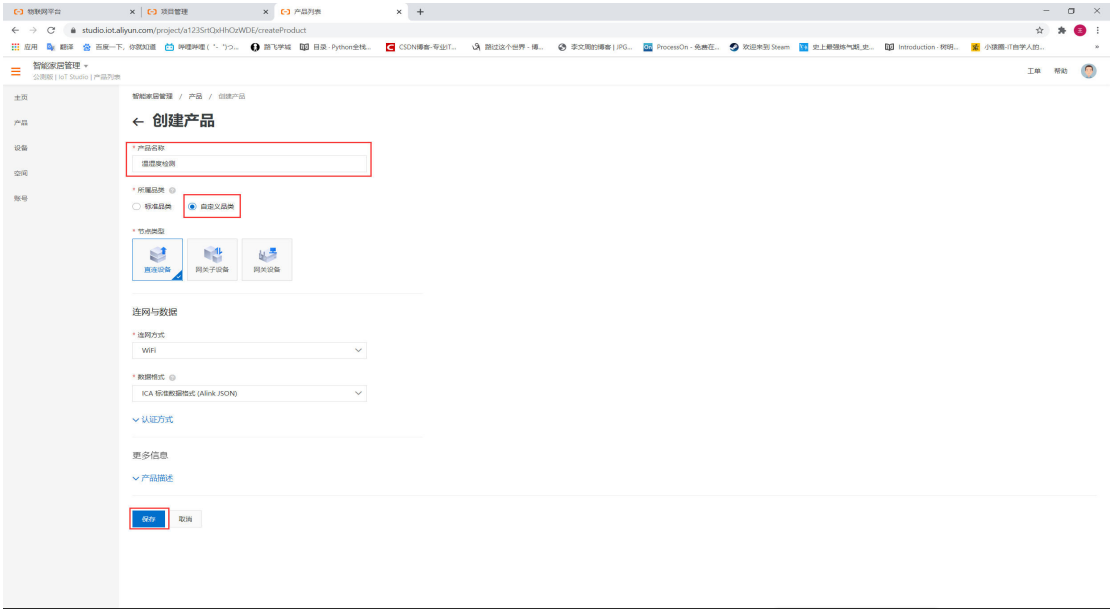


选择 **IoT Studio** 下的**项目管理**进入到之前创建的智能家居管理。

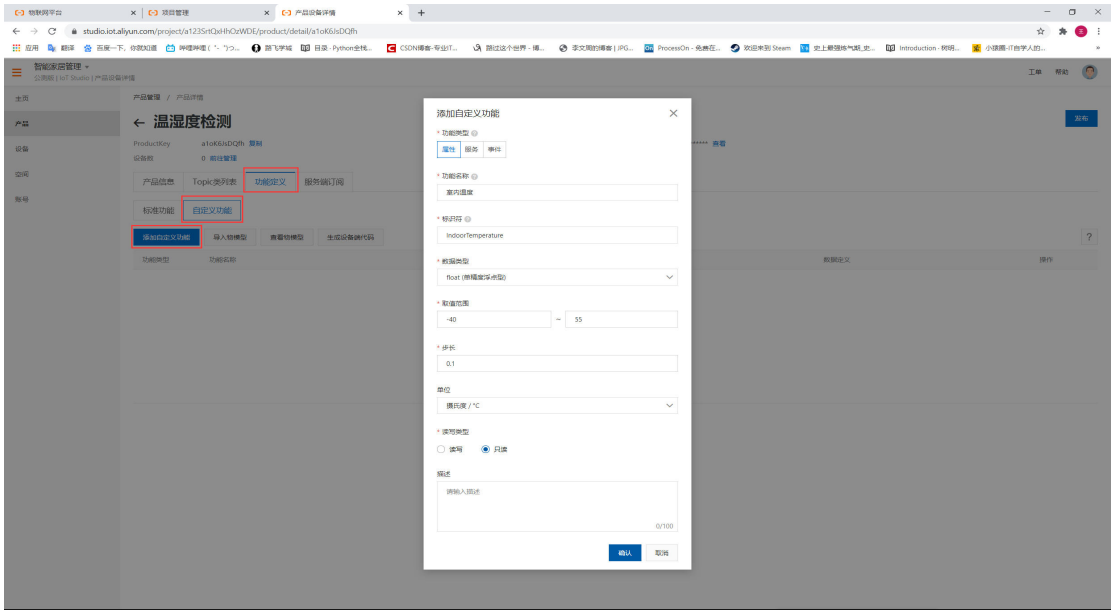


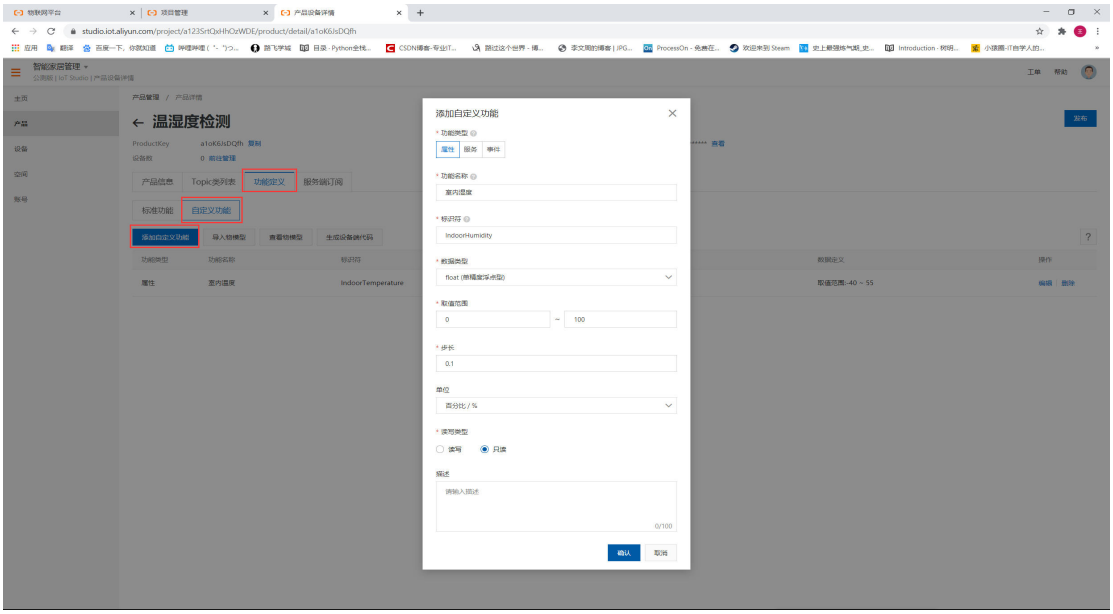
3, 创建一个**温湿度检测**的产品，同时添加功能定义。





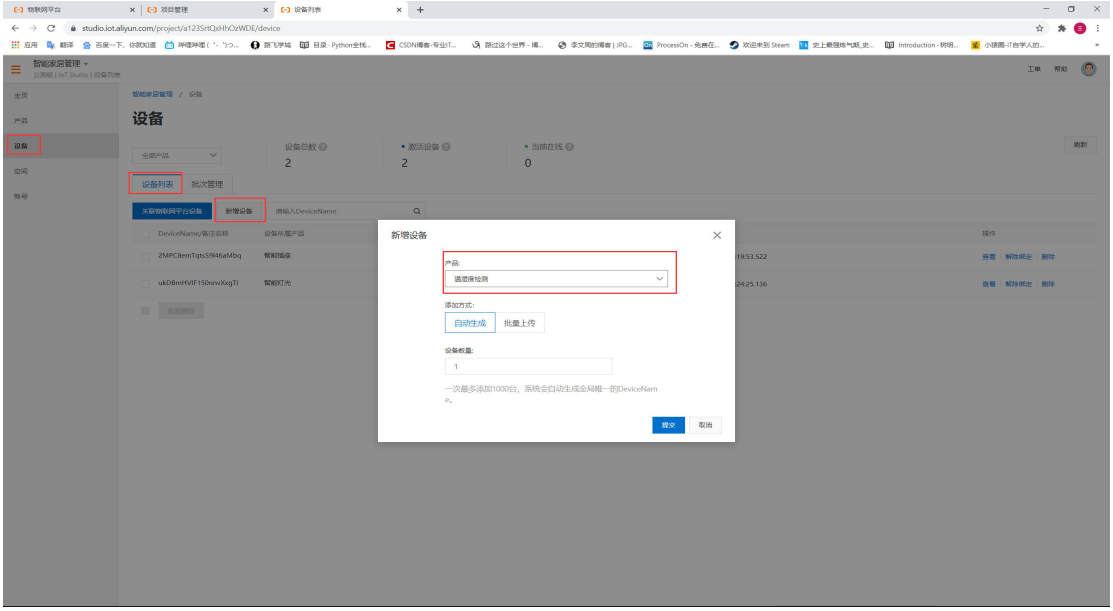
为温湿度监测产品定义功能名分别为“**室内温度**”和“**室内湿度**”的属性，自定义产品标识符，室内温度取值范围为-40~55,室内湿度为 0~100，如下图所示。功能添加完毕后，会自动出现在自定义功能栏下。





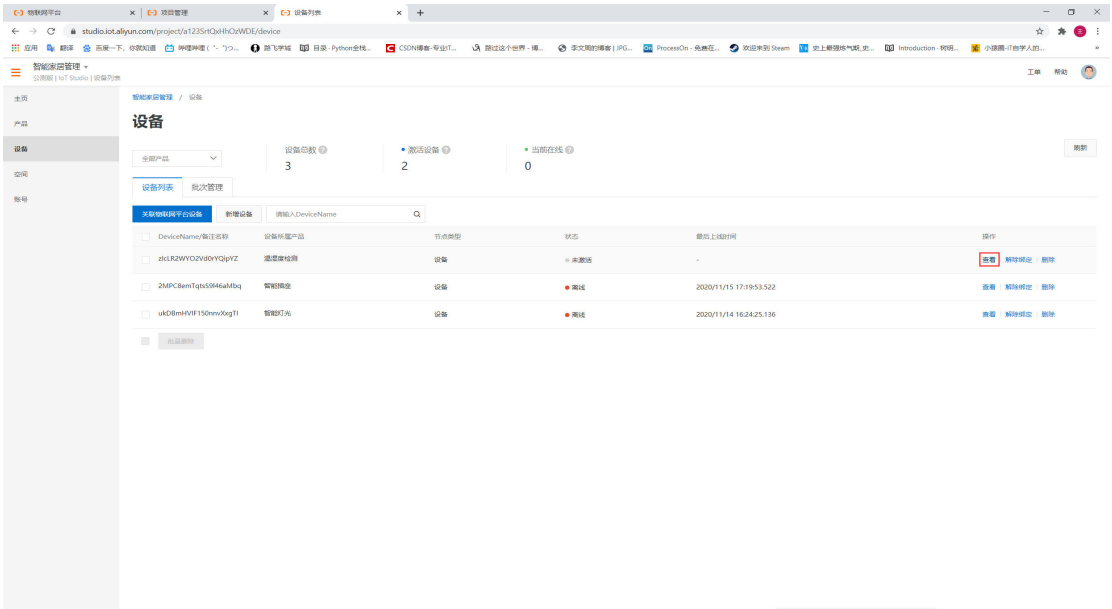
功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	室内温度	IndoorTemperature	float (单精度浮点型)	取值范围:-40 ~ 55	编辑 | 删除
属性	室内湿度	IndoorHumidity	float (单精度浮点型)	取值范围:0 ~ 100	编辑 | 删除

4.为**温湿度监测**产品创建一个的设备，如下图所示，设备添加完毕后，会自动出现在设备列表中。

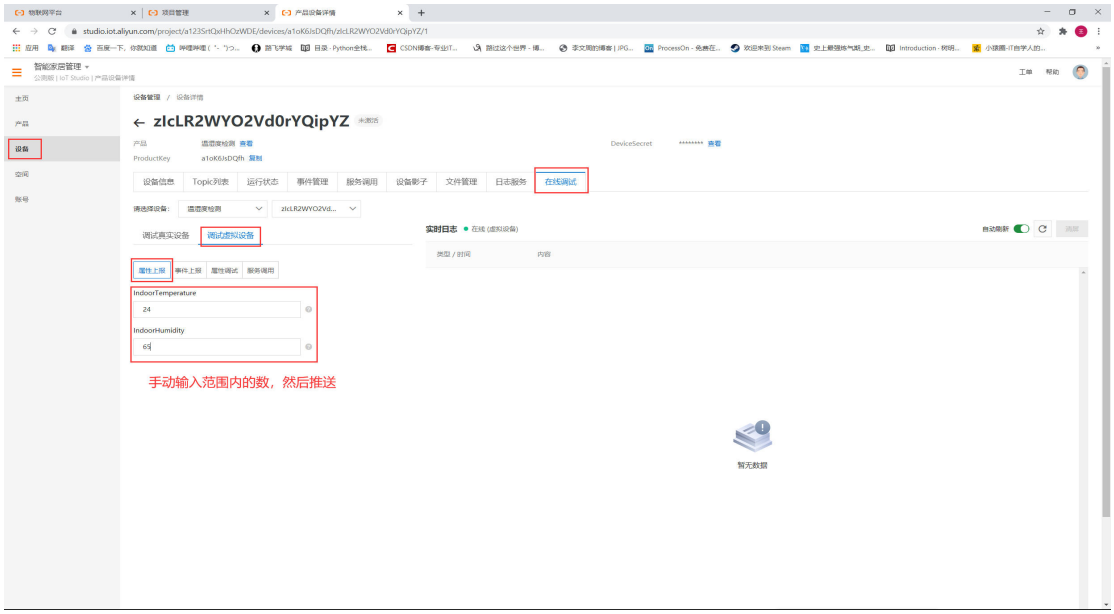


点击**提交**，会弹出设备激活凭证下载，点击下载（激活凭证是我们连接阿里云的关键参数），关闭后，该设备会自动出现在设备列表中。由于某个产品中的设备会继承该产品的所有属性，故设备具有灭、亮的工作状态属性了。

5. 在线调试参数，当设备生成成功后我们可以选择在线调试参数。这一步的目的主要是为了生成移动应用时可以配置数据。点击查看，进入到设备详情页。再点击在线调试，进入到调试页面。

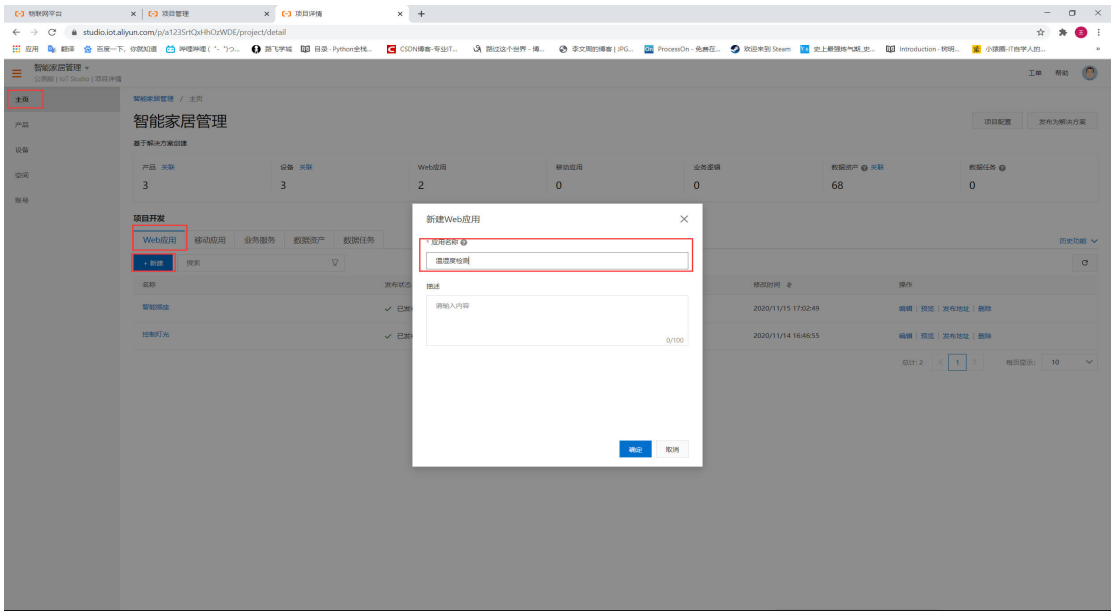


按照下图所示的方式进行调试推送即可。



6.创建 Web 可视化开发

在智能家居管理中新增 Web 应用，默认模板，应用名称改为智能温湿度检测，新增页改为温湿度检测

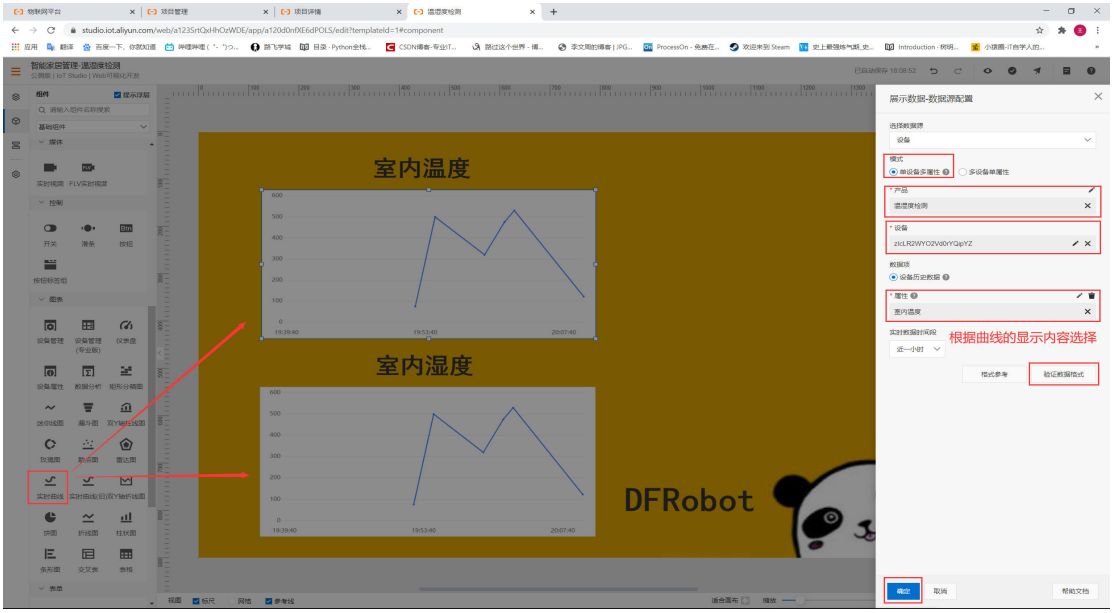


将自定义新增页改为温湿度检测。

跟应用一 一样的方法，添加图片背景和文字，这里不做多介绍。

添加室内湿度曲线

添加曲线。将曲线拖到画布中，调整大小，进行**配置**，将产品关联到智能温湿度检测，设备关联到 DHR11，属性为室内湿度。点击确定。

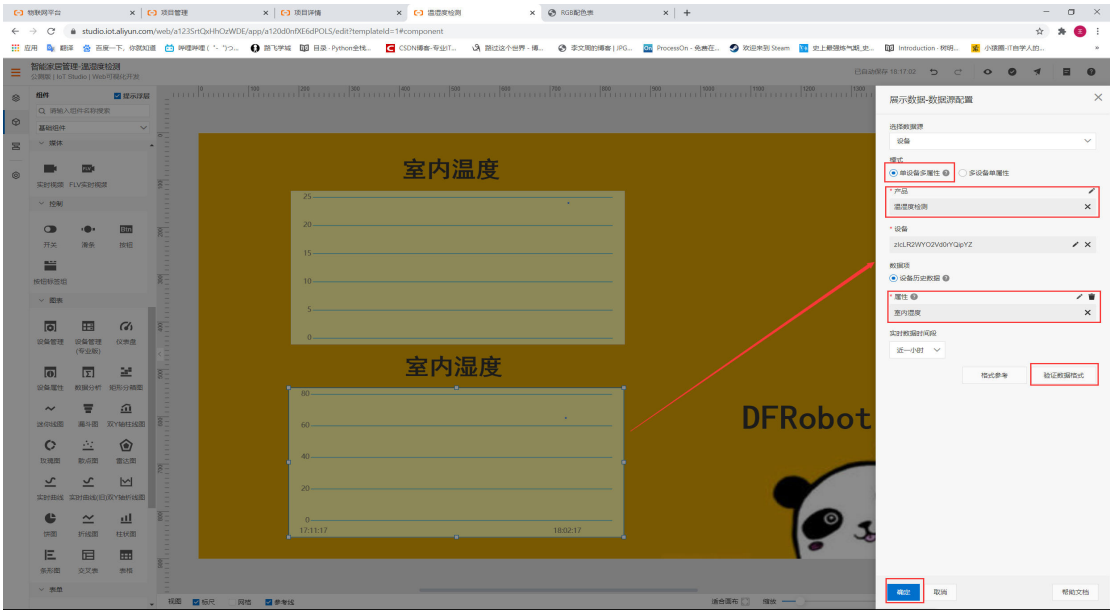


添加文字。

添加室内温度曲线

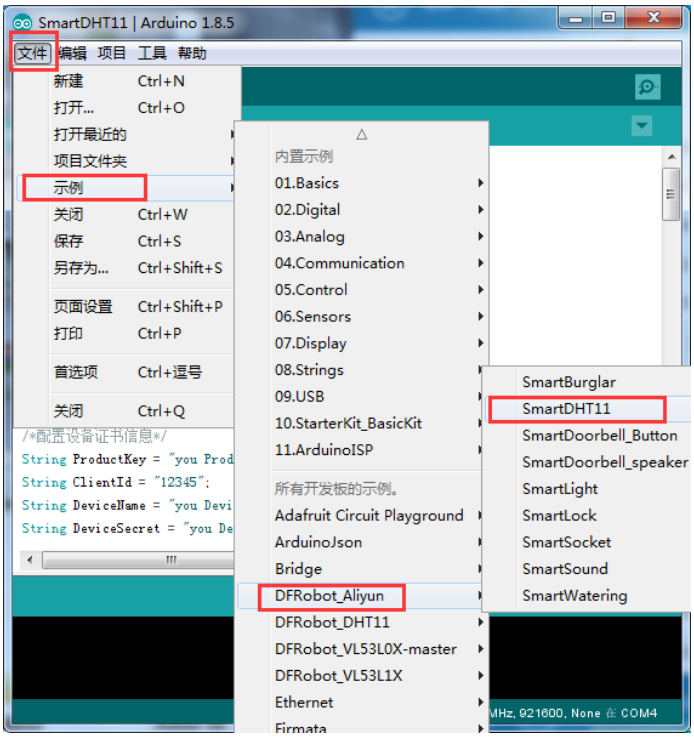
添加曲线。跟添加室内湿度的方法是类似的，只是将属性换为室内温度。

添加文字。



点击保存，发布。直接点击链接打开。Web 可视化应用就完成了。

8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartDHT11** 项目，如下图所示：



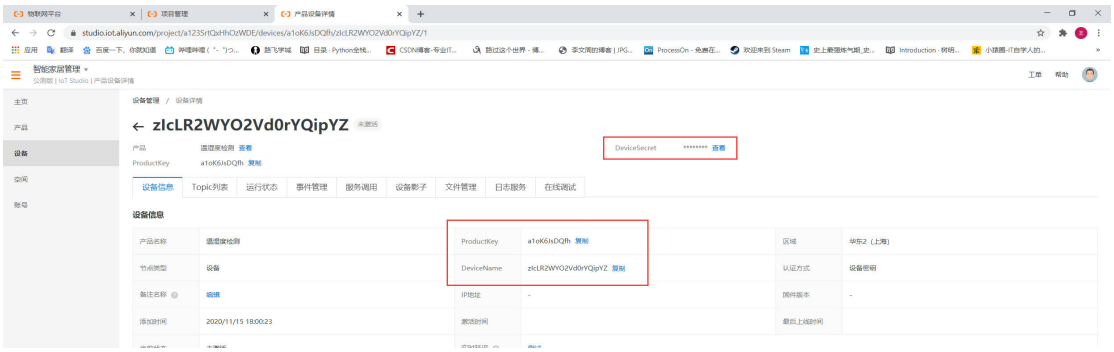
① 配置 WiFi 名和密码。将程序中配置 WiFi 名和密码的部分(如下图所示）更换为可用的 WiFi。

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

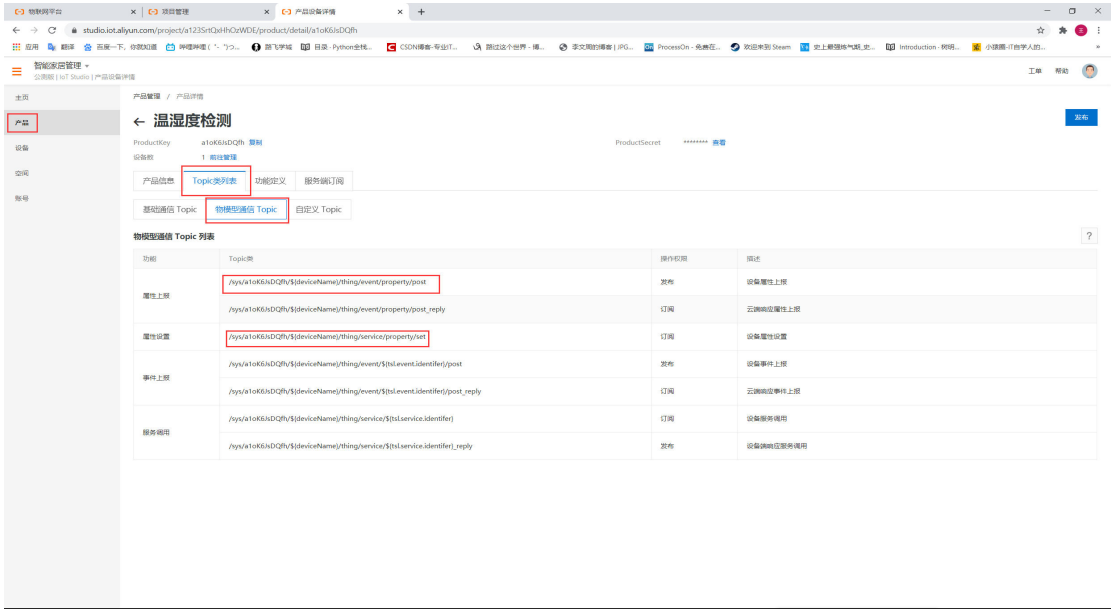
② 填写设备证书信息。获取设备 **DHT11** 的设备信息，如下图所示：

将获取到的 **DHT11** 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 上获取设备 DHT11 的 Topic 列表，如下图所示，



将获取到的 Topic 列表信息，复制粘贴到如下图所示的程序中：

```
/*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
```

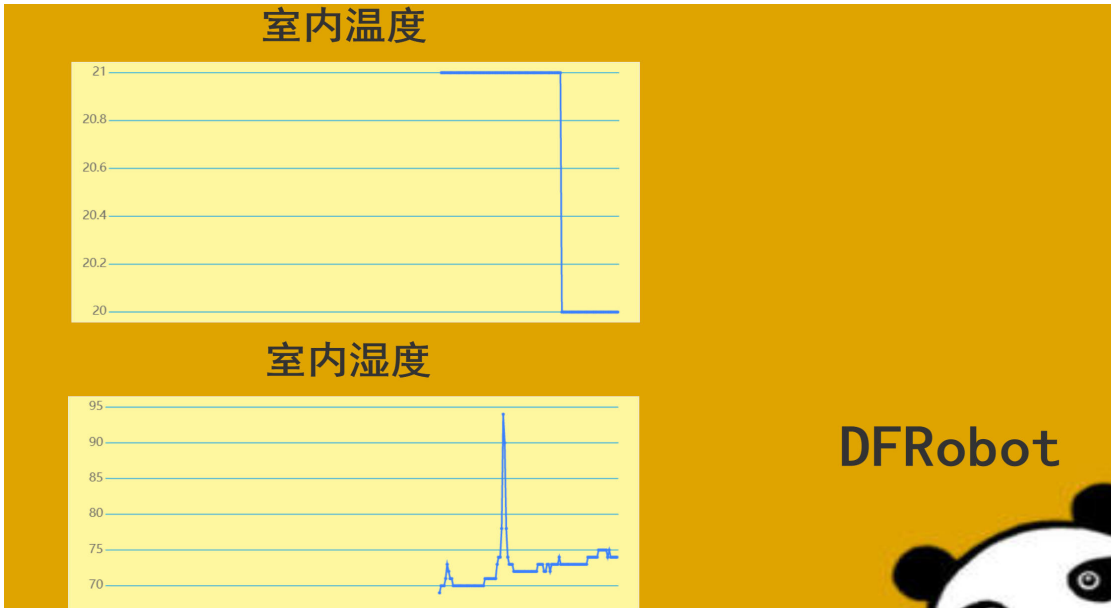
④替换产品标识符。在设备管理的产品项找到名为温湿度监测的产品，点击查看，点击功能定义，即可在自定义功能栏看到标识符，如下图所示：

将上图红色框内的标识符，分别复制粘贴到如下图所示的程序中。

```
/*需要操作的产品标识符(温度和湿度两个标识符)*/
String TempIdentifier = "you Temp Identifier";
String HumiIdentifier = "you Humi 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 DHT11 设备关联，打开下载的智能家居控制台，选择温湿度监测，即可进入控制页面，一段时间后，即可在手机 Web 页面上看到温湿度变化的实时曲线，如下图所示：



3.4 应用四：智能浇花

通过前一小节对温湿度监测的学习，我们大概了解了 App 如何获取 DHT11 温湿度传感

器感知的信息。接下来，我们会学习另一个获取传感器信息的项目——智能浇花。

应用目标：通过手机 App 监测土壤湿度状况，已根据湿度情况，确定是否给花浇水。

所需元件

土壤湿度传感器 x1

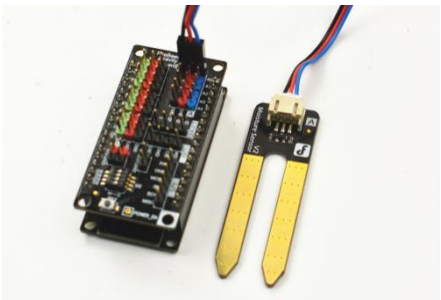
FireBettle Board-ESP32 x1

FireBettle Gravity 扩展板 x1

智能手机 x1

硬件连接

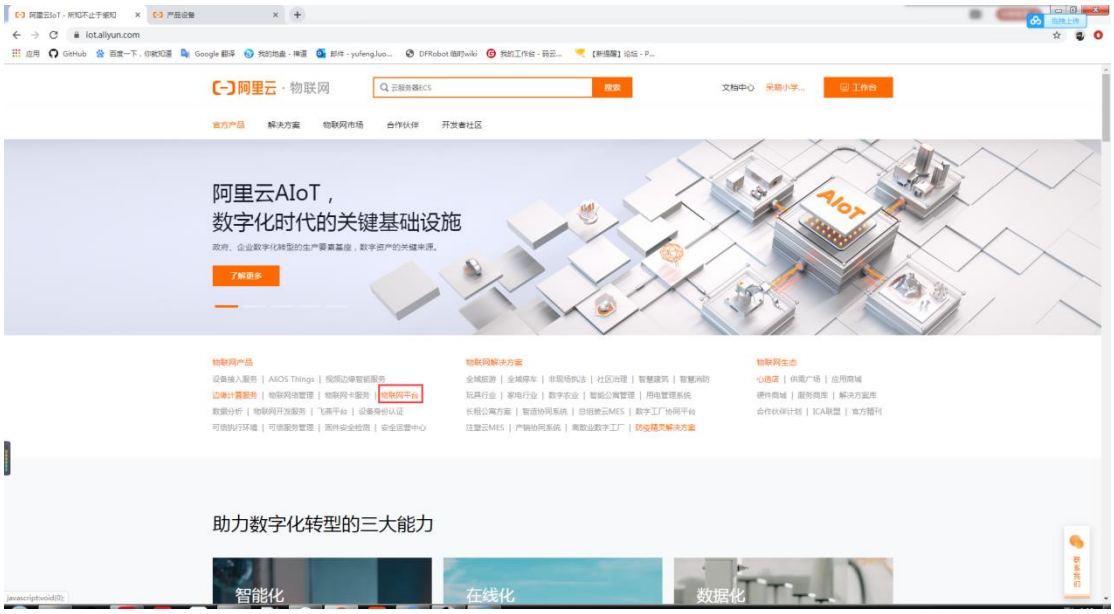
把 FireBettle Gravity 扩展板直插在 FireBettle Board-ESP32 主板上，再把土壤传感器接在扩展板的 A0 数字引脚上，确保电源引脚、接地引脚和信号引脚都连接正确。



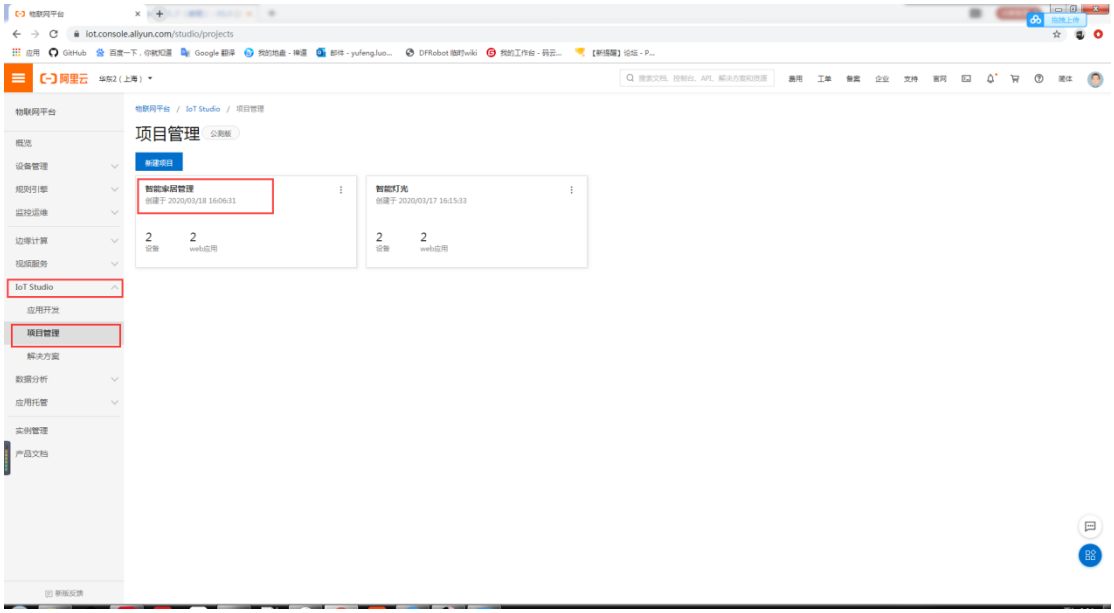
应用分析：智能浇花应用就是将土壤传感器获取的数据上传至手机 App。为此，我们需要做以下几件事：1.在阿里云 IoT 平台上创建一个名为“**智能浇花**”的产品，并为其定义功能为“**土壤湿度**”的功能；2.在“**智能浇花**”产品下挂载名为相应的设备；3.在“智能家居控制台”移动应用项目中添加一个名为“智能浇花”的 App 页。4.获取对应的设备证书，发布订阅 Topic 列表，以及获取产品的标识符，并将其在 Arduino 程序中修改。

制作步骤

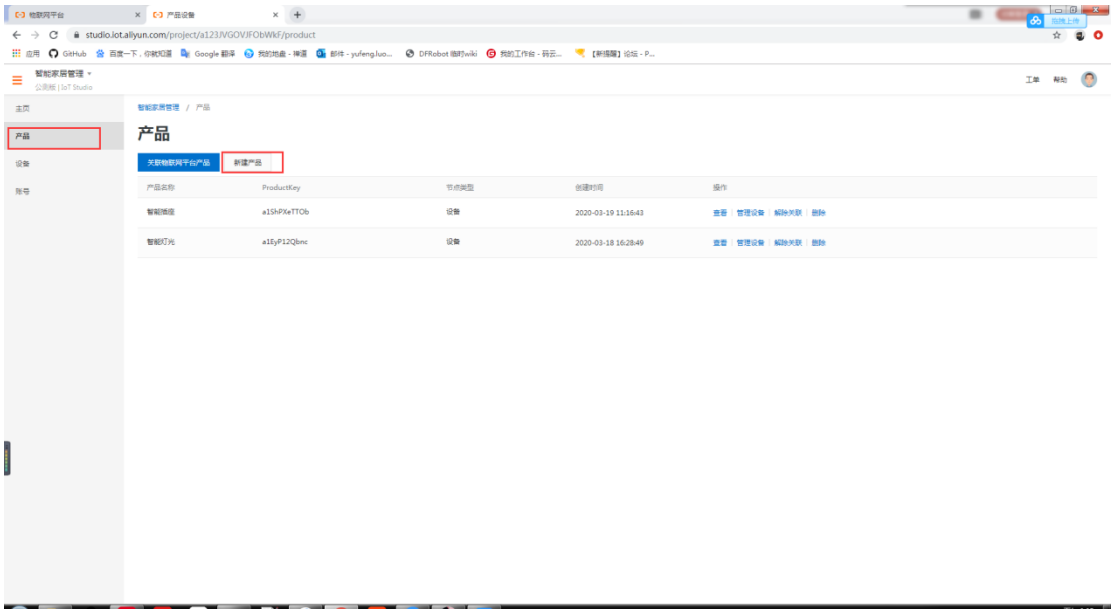
1，打开浏览器，输入网址 https://iot.aliyun.com,登录并进入**物联网平台**；

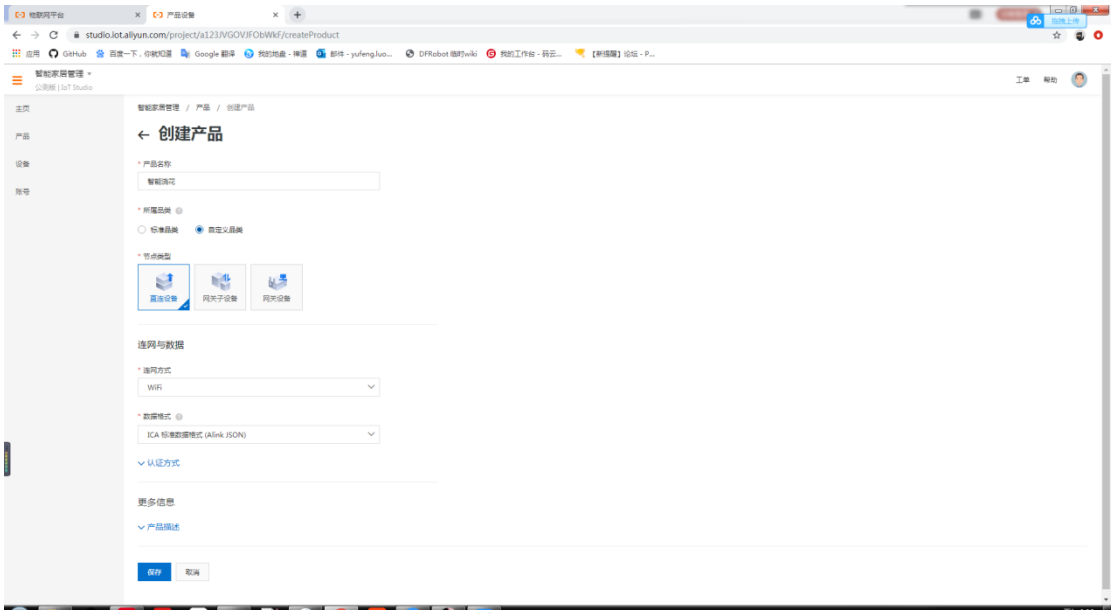


选择 IoT Studio 下的项目管理进入到之前创建的智能家居项目。

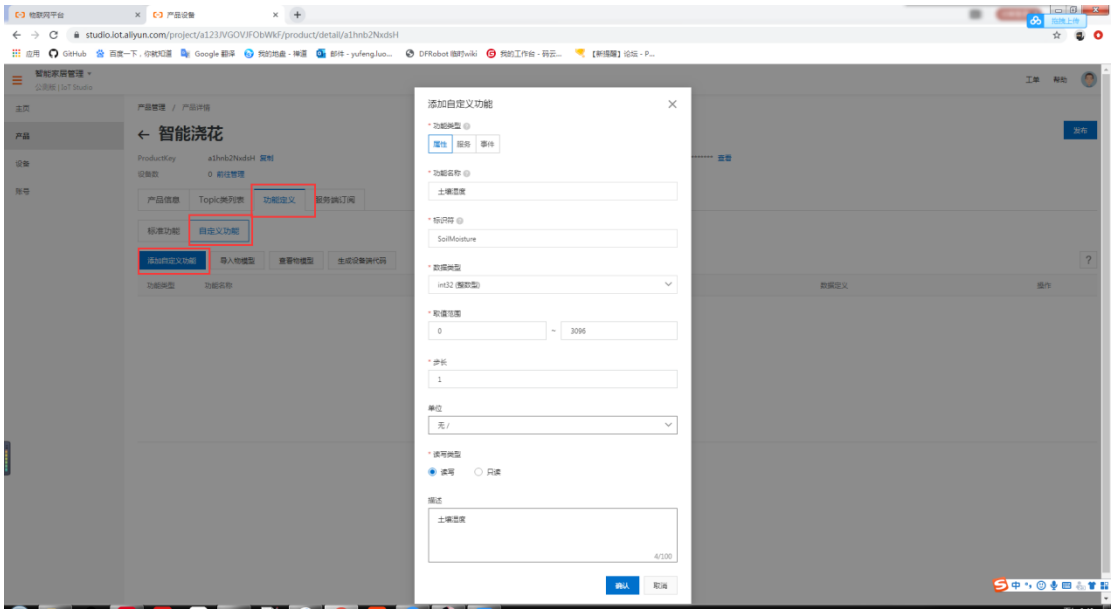


2.创建一个名为**智能浇花**的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



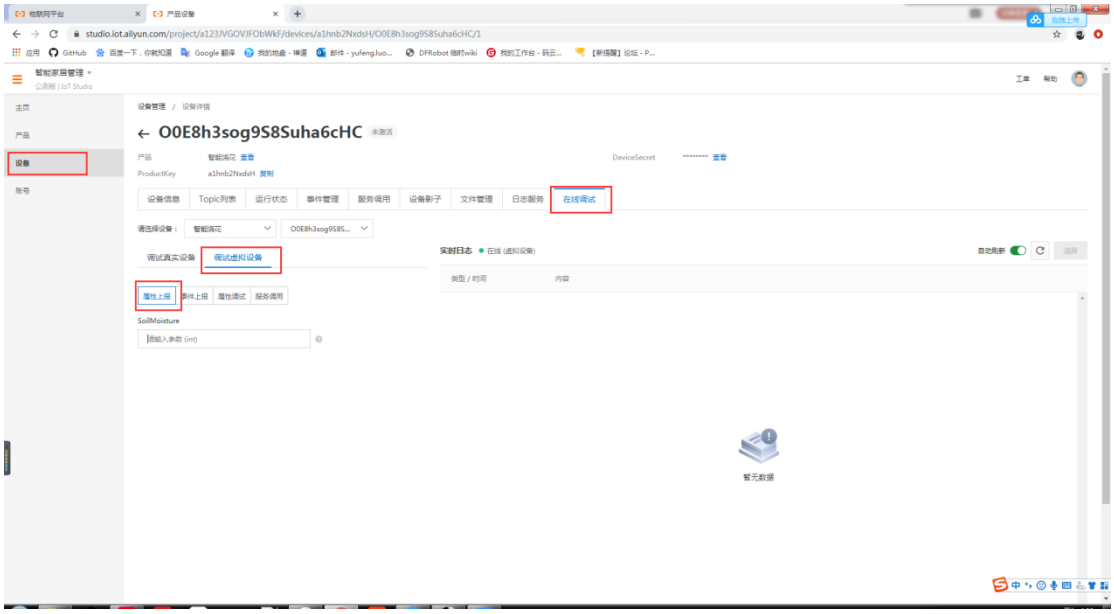


3. 为智能浇花产品定义一个功能名为“**土壤湿度**”的属性，自定义产品标识符，湿度数值用 int32 整数型表示，取值范围 1~3096，步长为 1，如下图所示：

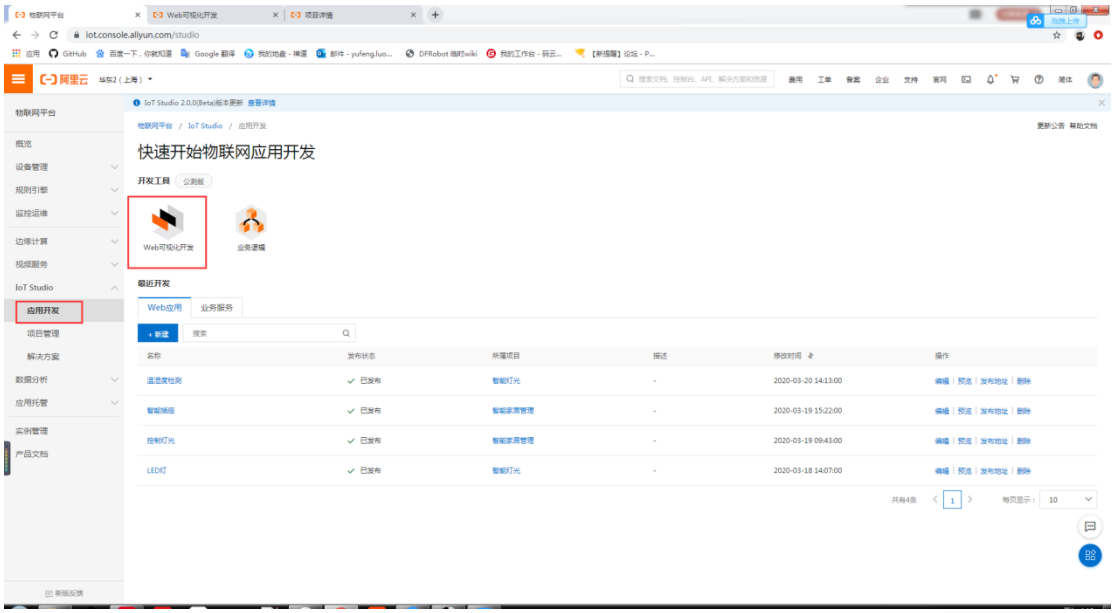


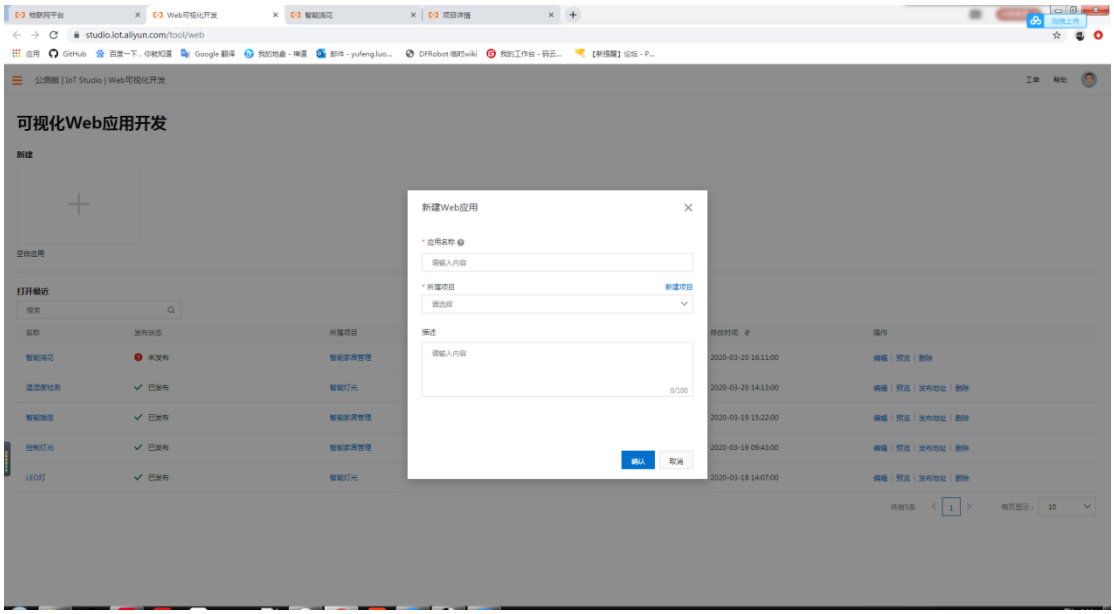
4. 为**智能浇花**添加一个设备，如下图所示，点击确定。下载好**激活凭证**后（激活凭证是我们连接到阿里云的关键参数），点击**关闭**。

5. 设备创建成功后，点击产品进行在线调试。点击**产品**，再点击**在线调试**。按照下图所示操作



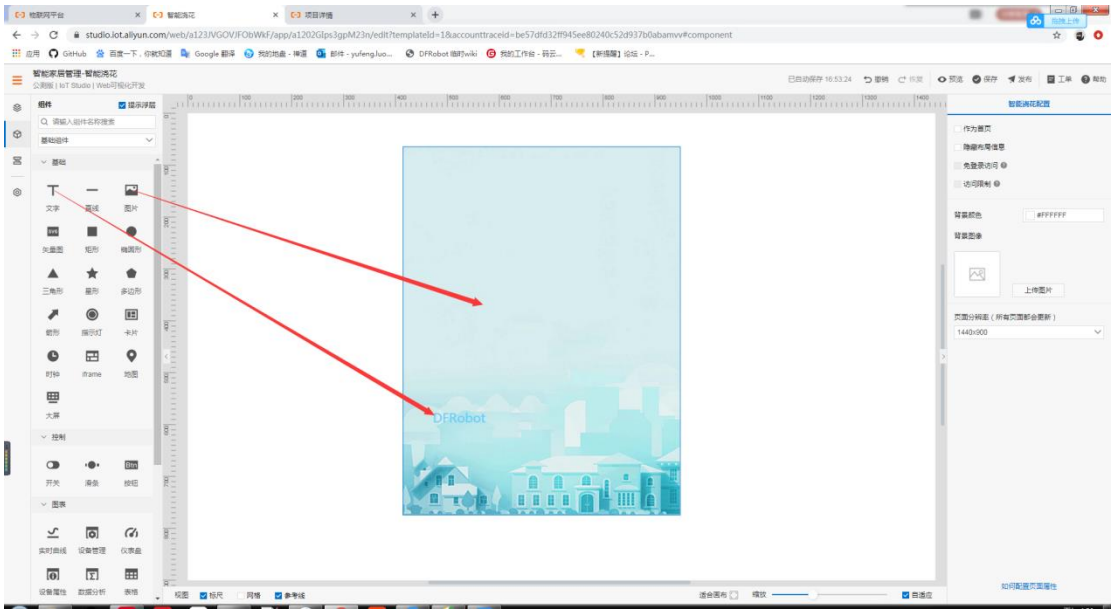
6. 如下图所示在**项目管理界面**中点击 **Web 可视化开发**然后点击**新建 web 应用**。输入名称后点击**确认**即可。



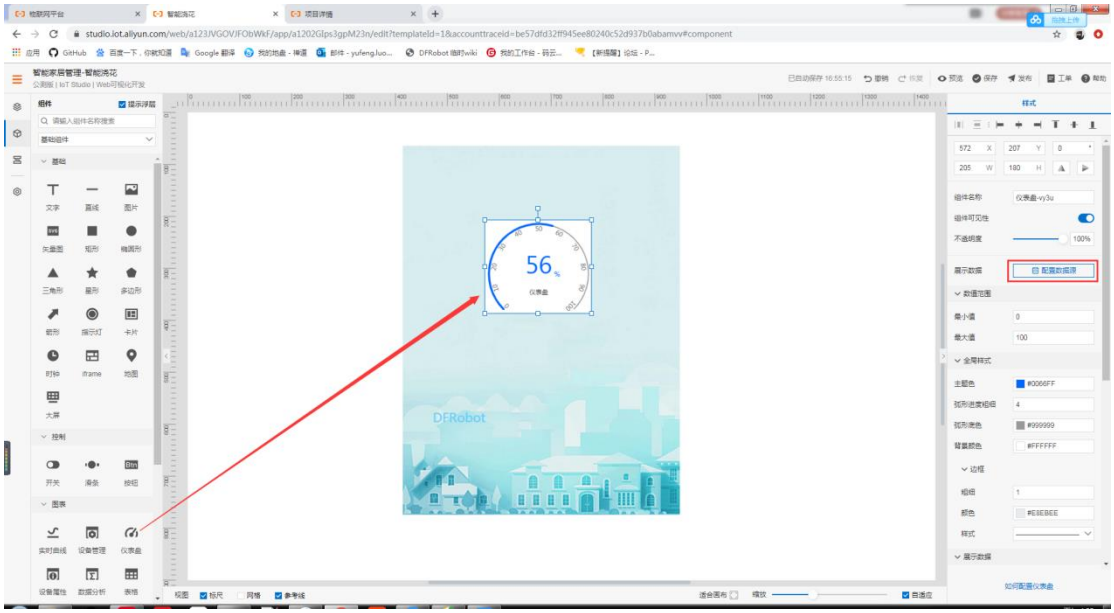


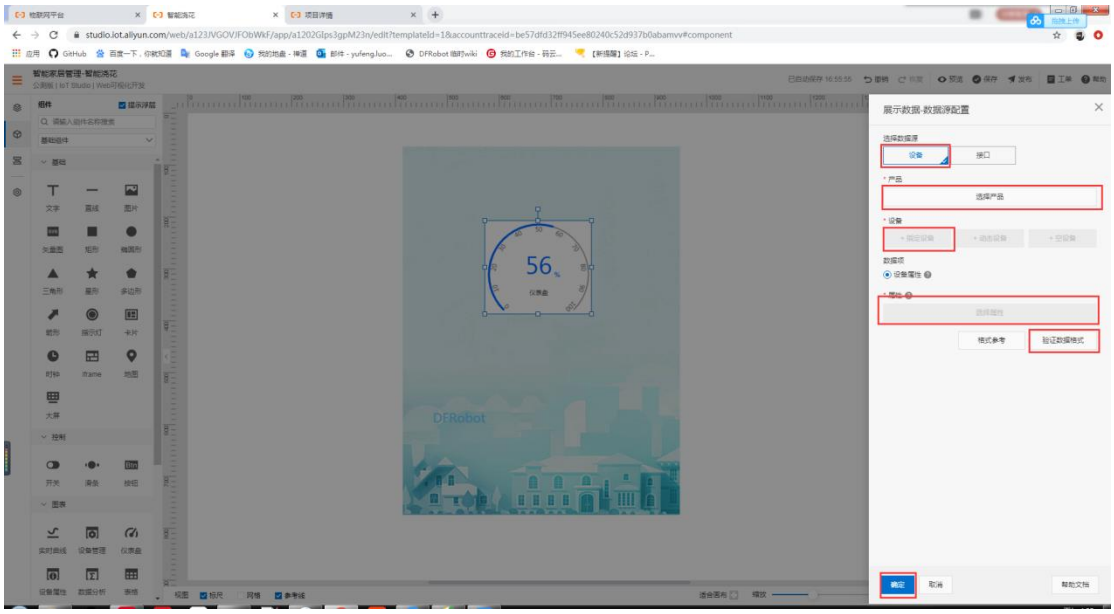
在智能家居管理中新增 Web 应用，新增页改为智能浇花。

跟应用一 一样的方法，添加图片背景和文字，这里不做多介绍



添加仪表盘。将仪表盘拖到画布中，进行**配置**，将产品关联到智能浇花，设备关联到之前创建的设备，属性为土壤湿度。点击确定。

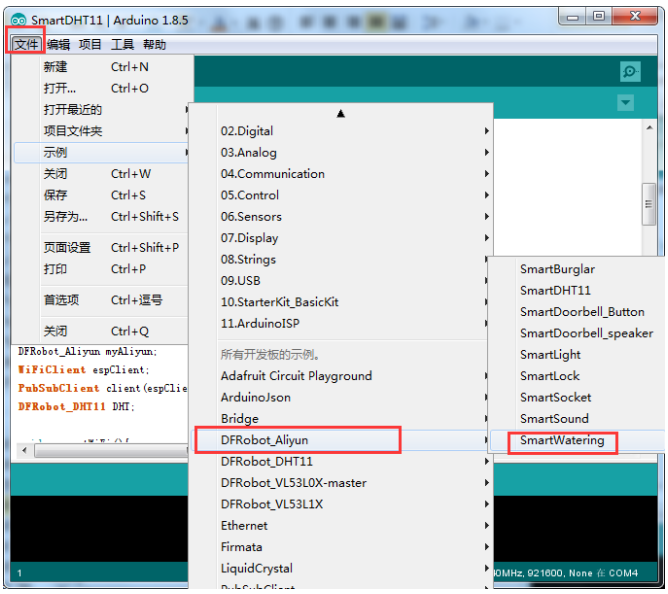




点击保存，发布。直接点链接打开。Web 可视化开发就完成了。

8. Arduino 程序修改

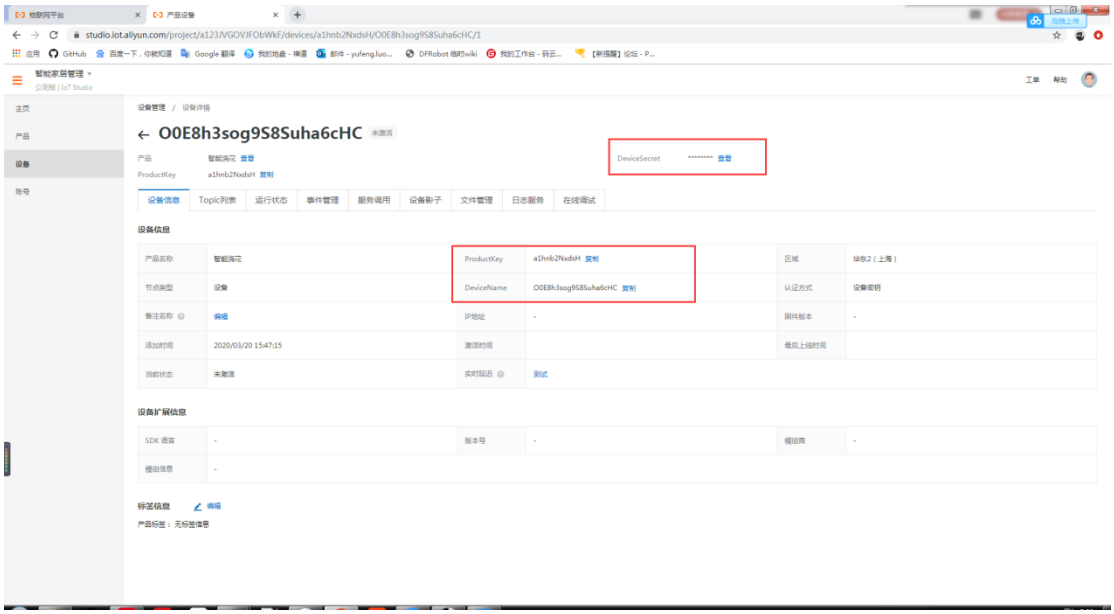
点击**文件->示例->DFRobot_Aliyun->SmartWartering** 项目，如下图所示：



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";
```

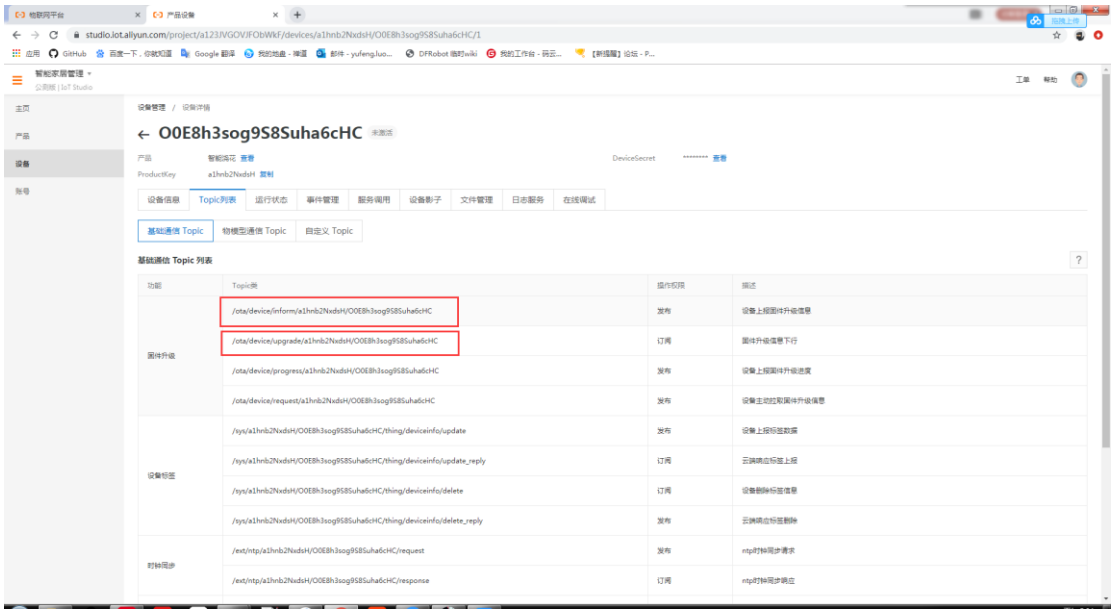
② 填写设备证书信息。获取设备的信息，如下图所示：



将获取到的设备证书信息，复制粘贴到如下图所示的程序模块中。

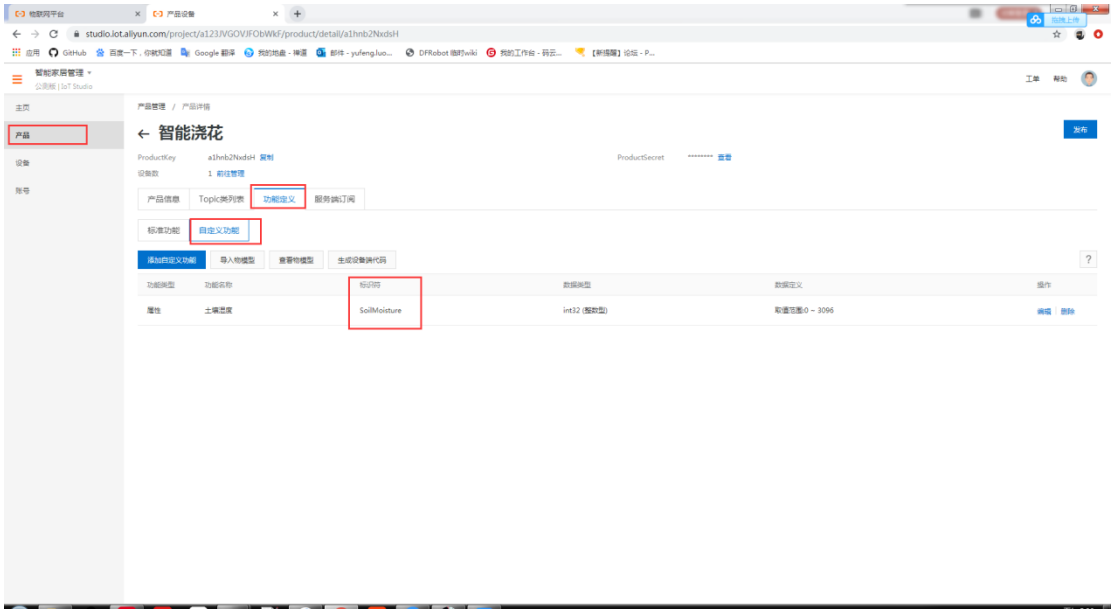
```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

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
```

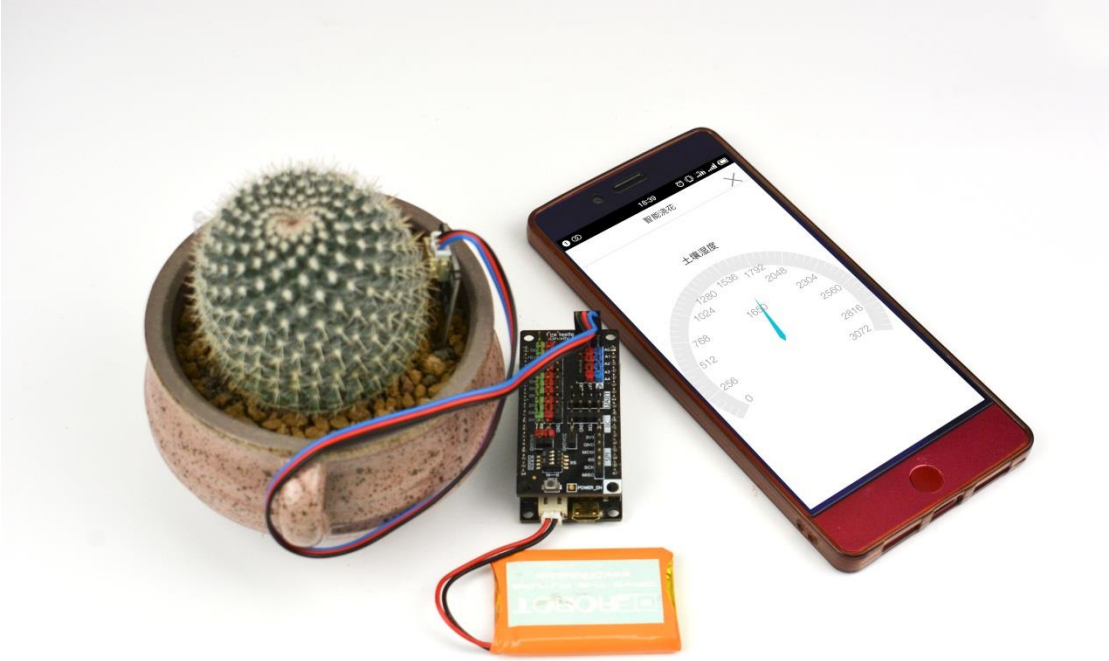
④替换产品标识符。在设备管理的产品项找到名为智能浇花的产品，点击查看，点击功能定义，即可在自定义功能栏看到标识符，如下图所示，将其替换到程序中的产品标识符即可。




```
/*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 平台上的智能浇花设备关联，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

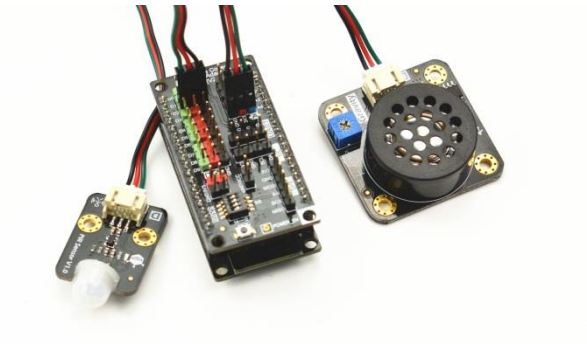
FireBettle Board-ESP32 x1

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。

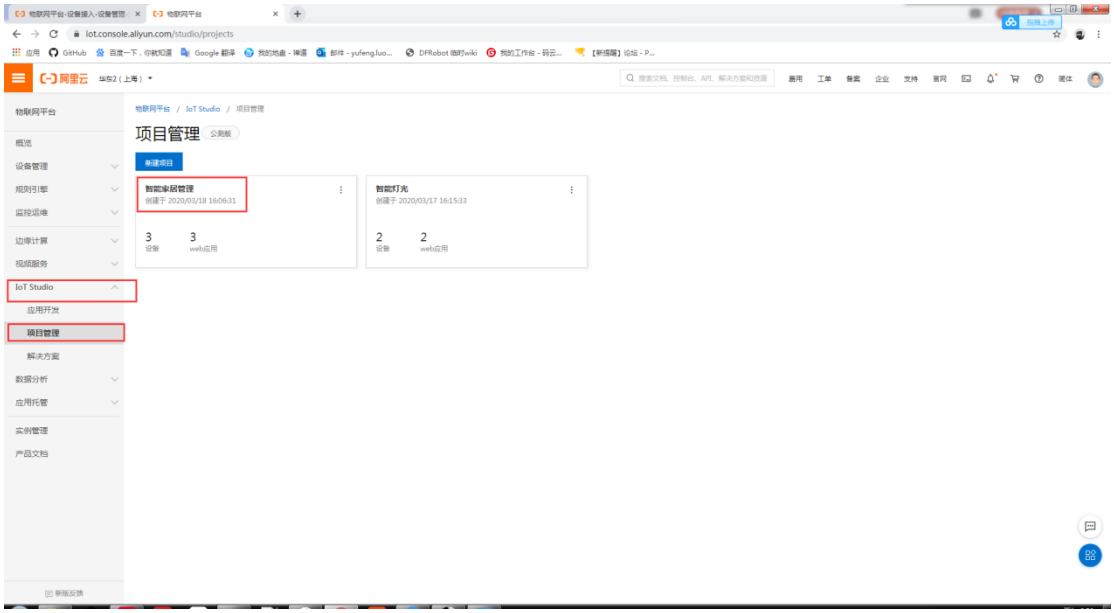


应用分析：智能防盗应用中，PIR 检测到有人，则启动喇叭报警提示，同时将“家里有人闯入”的消息推送至钉钉，提醒用户。这个项目需要我们做以下事情：1.在阿里云 IoT 上创建一个名为“**智能防盗**”的产品，并为其定义一个功能为“**防盗警告**”的属性；2.为“**智能防盗**”产品添加一个设备；3、创建并开发一个名为“**智能防盗**”的服务开发；4 获取设备证书，发布订阅 Topic 列表，以及获取产品的标识符，并在 Arduino 程序中填写修改。

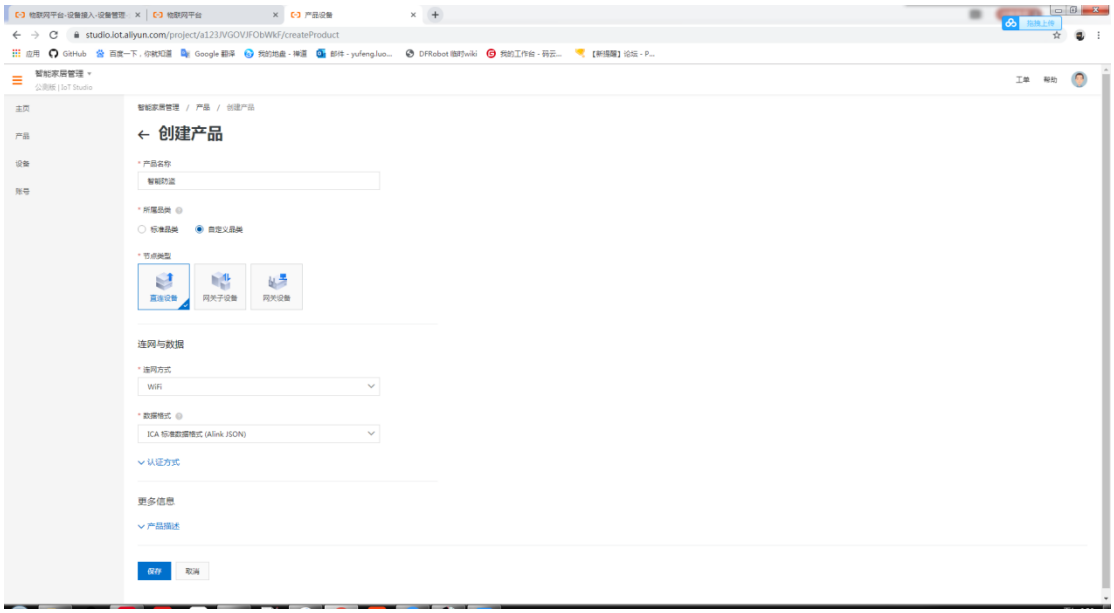
制作步骤

1，打开浏览器，输入网址 https://iot.aliyun.com,登录并进入**物联网平台**；**登陆到首页，**如下图所示点击**工作台**进入到**控制台。**

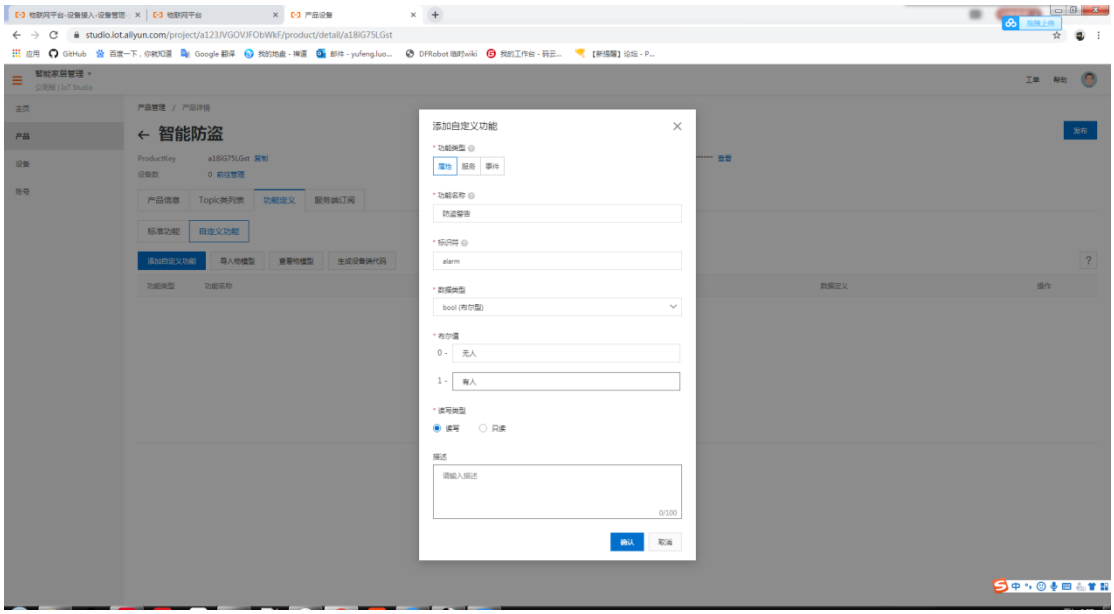
选择**开发服务**下的 **IoT Studio** 进入到之前创建的智能家居项目。



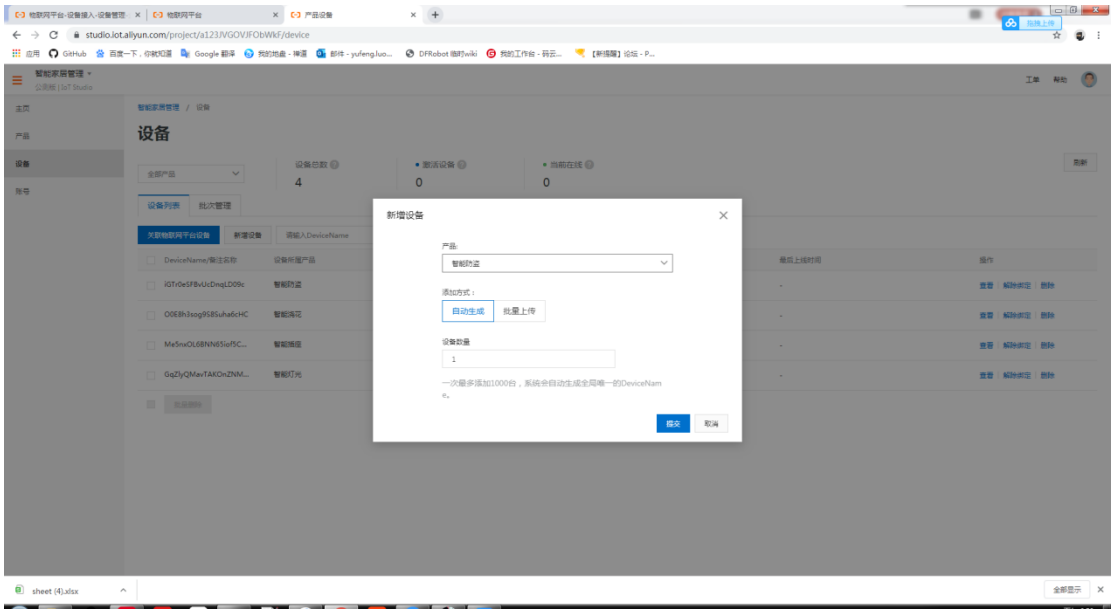
2.创建一个名为**智能防盗**的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



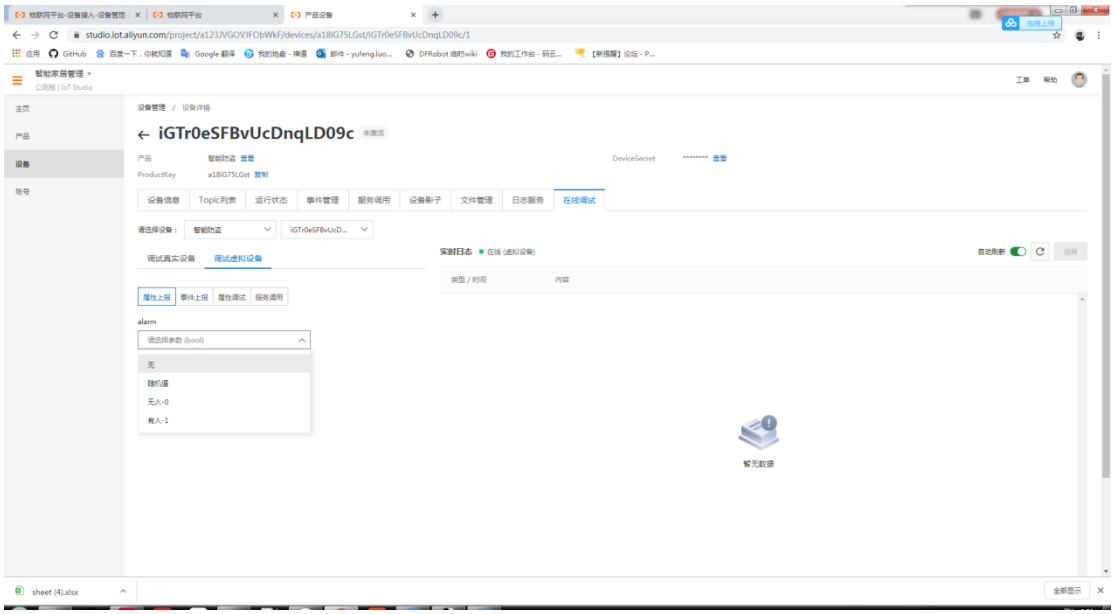
3. 为智能防盗产品定义一个功能名为“**防盗警告**”的属性，自定义一个产品标识符，防盗警告有 2 种状态：有人闯入（1），无人闯入（0）。如下图所示：



4. 为**智能插座**添加一个设备，如下图所示，点击确定。下载好**激活凭证**后（激活凭证是我们连接到阿里云的关键参数），点击**关闭**。

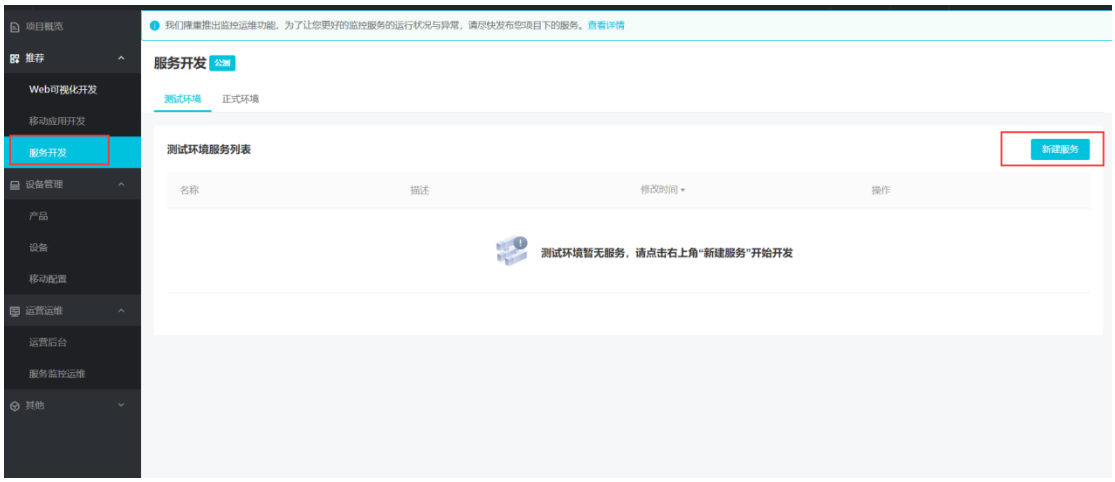


5. 设备创建成功后，点击产品进行在线调试。点击**设备**，再点击**在线调试**。按照下图所示操作



将下拉框的所有选项均推送一次：

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:

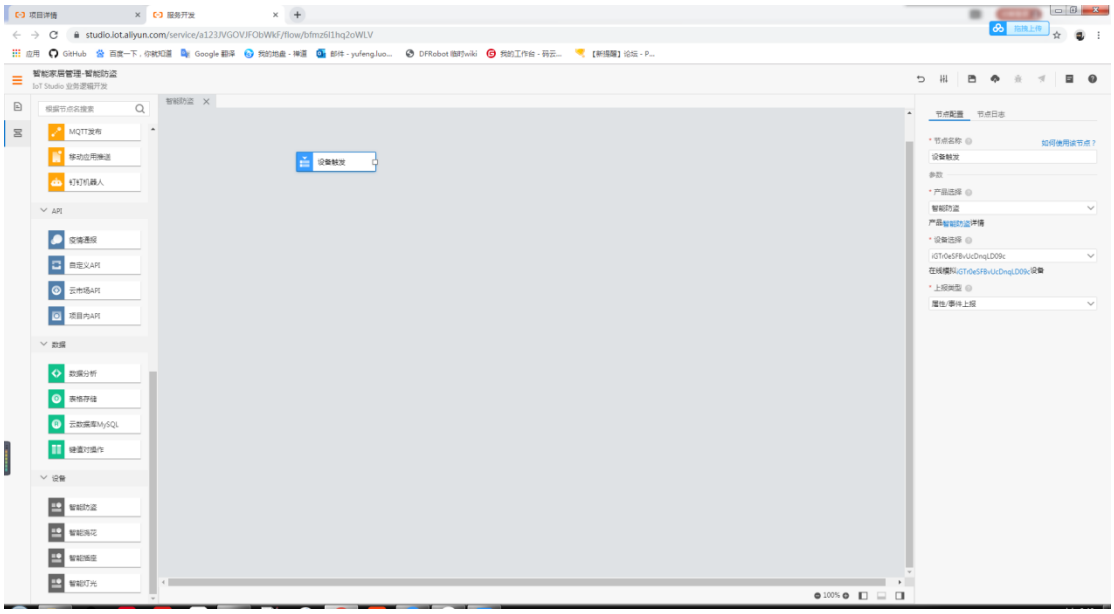


根据页面提示，创建一个名为“**智能防盗**”的服务，该服务内容大致为：当 PIR 传感器检测到有人闯入则启动喇叭报警，并将信息推送至钉钉，如下图所示:

7.点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址:

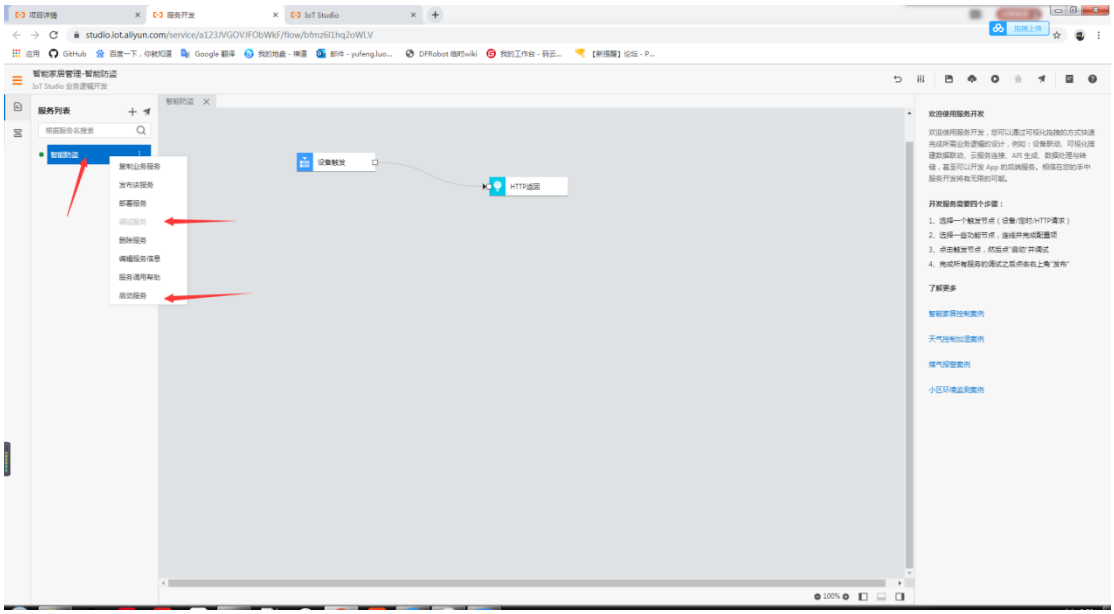
https://linkdevelop.aliyun.com/studioservice-doc#index.html

❶ 点击左侧导航栏的**节点**，在节点列表中选择**设备->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**之前创建的设备**；右侧栏配置项分别选择“智能防盗”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:

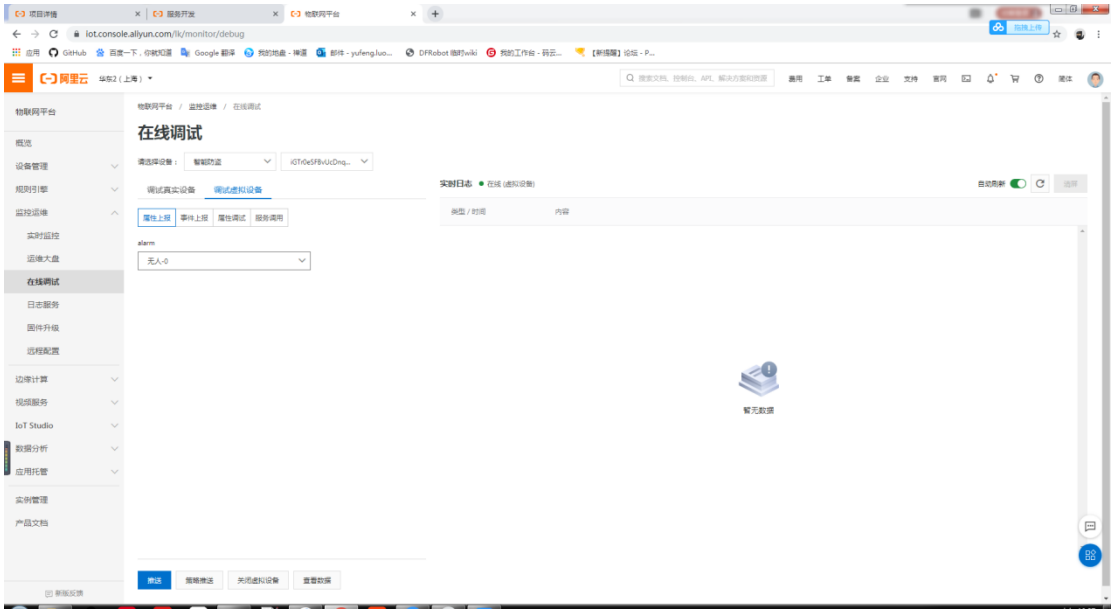


❷获取设备节点的节点输出信息。设备节点配置完成后，选中左侧节点列表的**功能**

->HTTP 返回（将设备触发和 http 返回的模块连接起来），点击服务列表，再右键点击**智能防盗**，选择**启动服务**，然后点击**调试服务**，如下图所示：

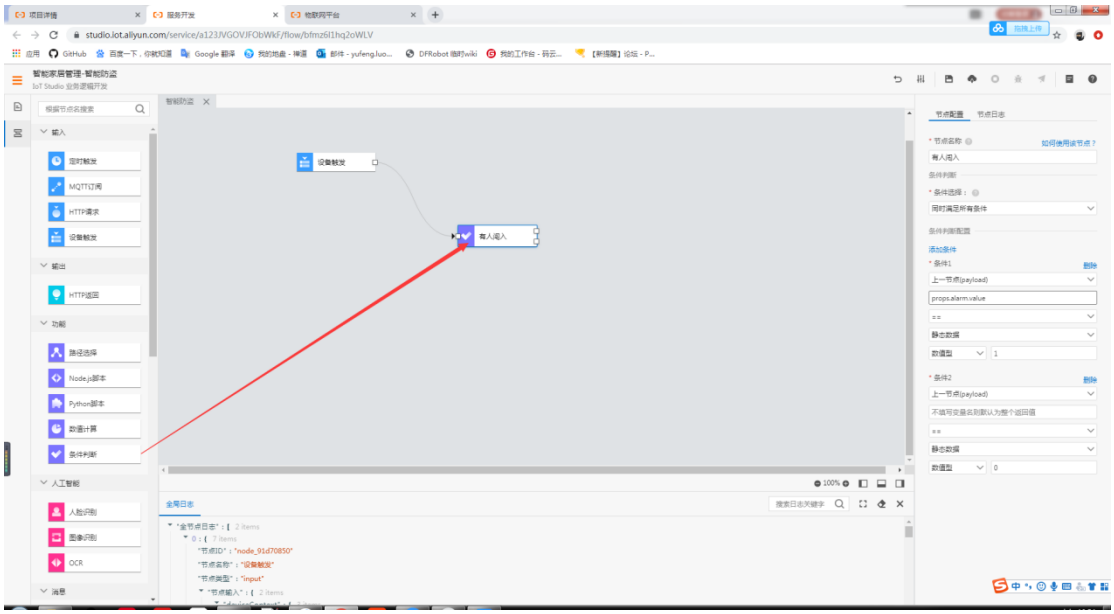


在弹出的“**在线调试**”页面中选择某个参数，点击推送，回到服务开发页面，点击**节点日志**，即可看到该设备节点的输入输出节点信息，如图：

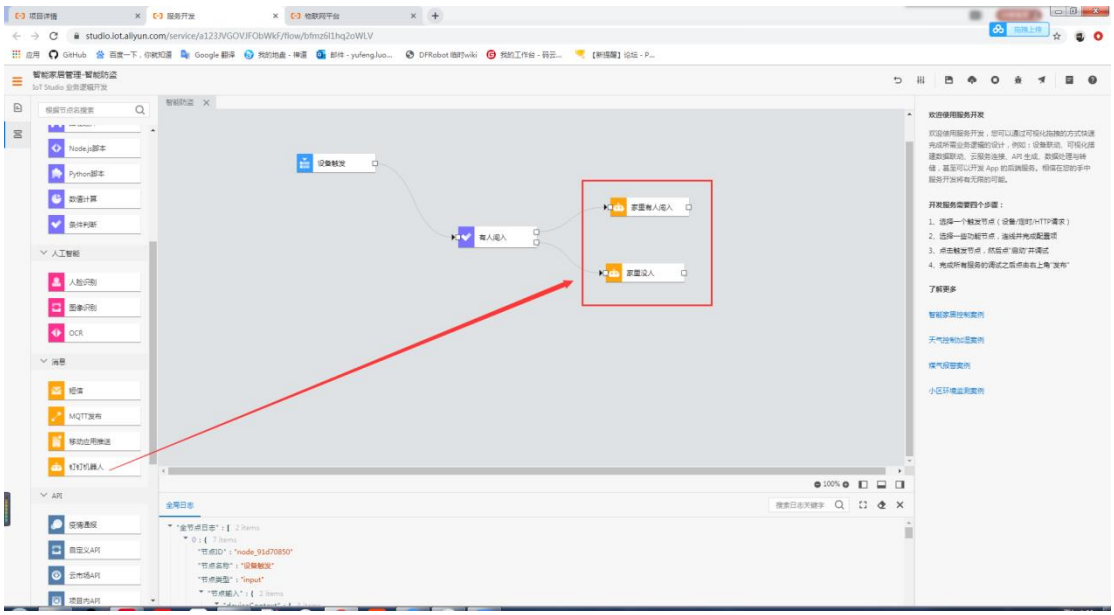


❸添加一个“**条件判断**”节点，条件判断节点相当于一个 if-else 判断。选中左侧节点列表的功能->条件节点，将其改为“有人闯入”，条件选择选为**任意一个满足条件**，条件 1 配置

为“**上一个节点(payload)**”+“**props.alarm.value**”；选择“**<**”，**静态数据**，**数值型**，最后一个框输入 **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？**

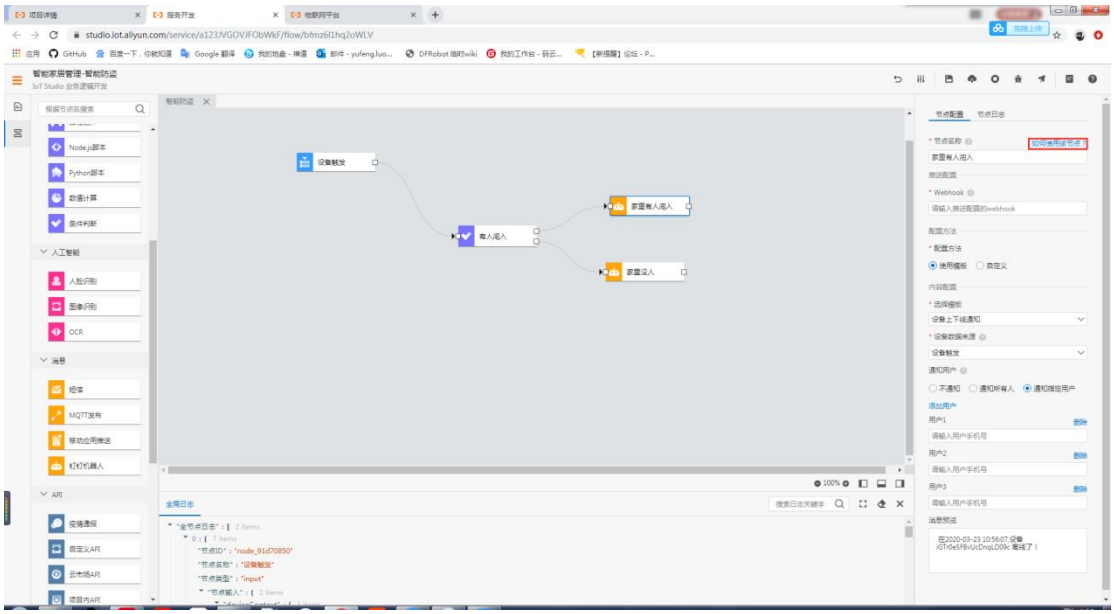


❹删除“HTTP 返回”节点，选中消息->钉钉机器人，添加钉钉机器人节点。并将 PIR 传感器、有人闯入、钉钉机器人节点连接起来。添加两个钉钉机器人。一个要选择是满足条件的接口连接。另一个选择不满足条件的接口。

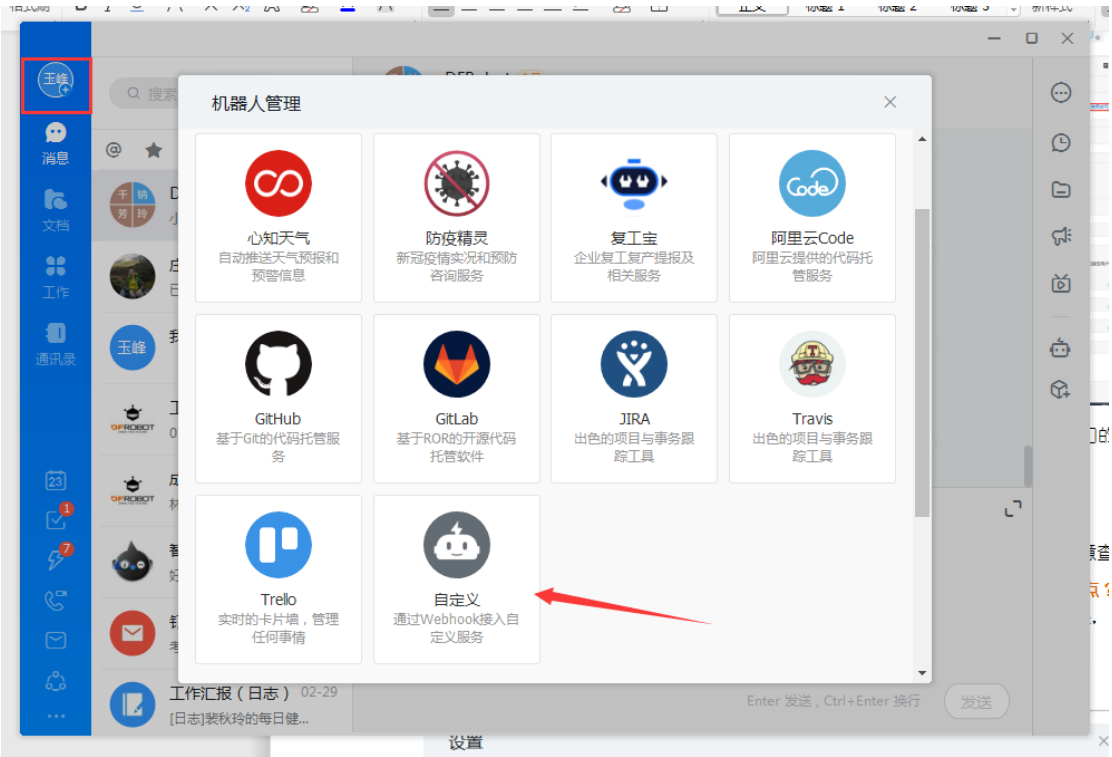


❺配置钉钉机器人节点。若不知道如何配置钉钉机器人，可参考**如何使用该节点？->如**

何配置钉钉机器人节点？ 如下图所示：



❺将不满足条件接口的钉钉机器人，节点名称改为**家里有人闯入**，将满足条件接口的钉钉机器人改为**家里安全**。



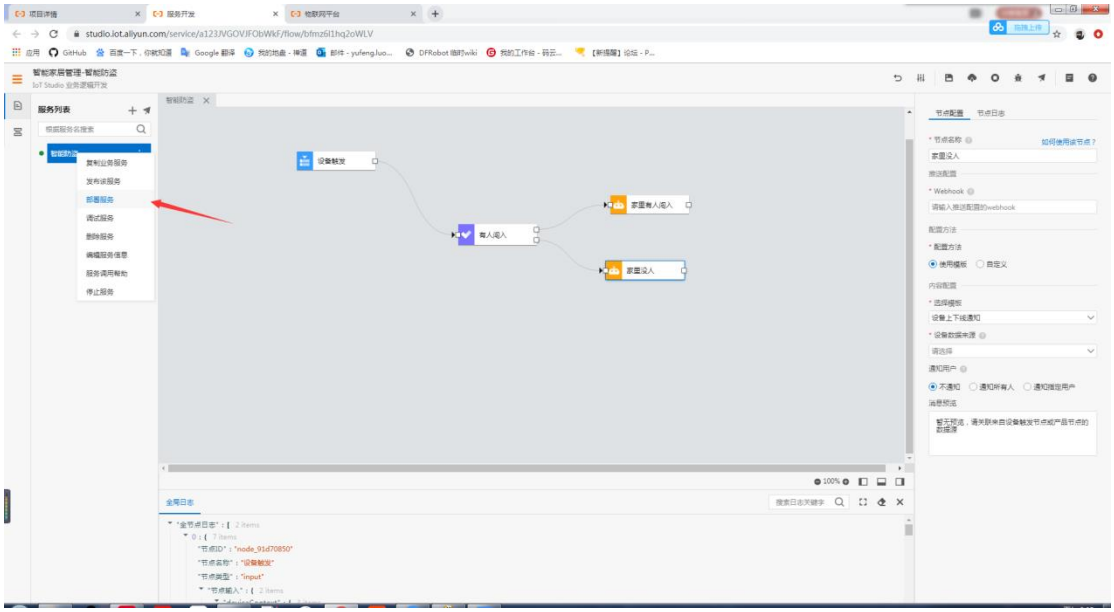
a. 获取 Webhook。在获取之前，你需要在钉钉群中，添加一个钉钉机器人(注意查看

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：

我们必须要设置发送内容的关键词：只有包含了这个关键词的内容，钉钉机器人才能推送到群里

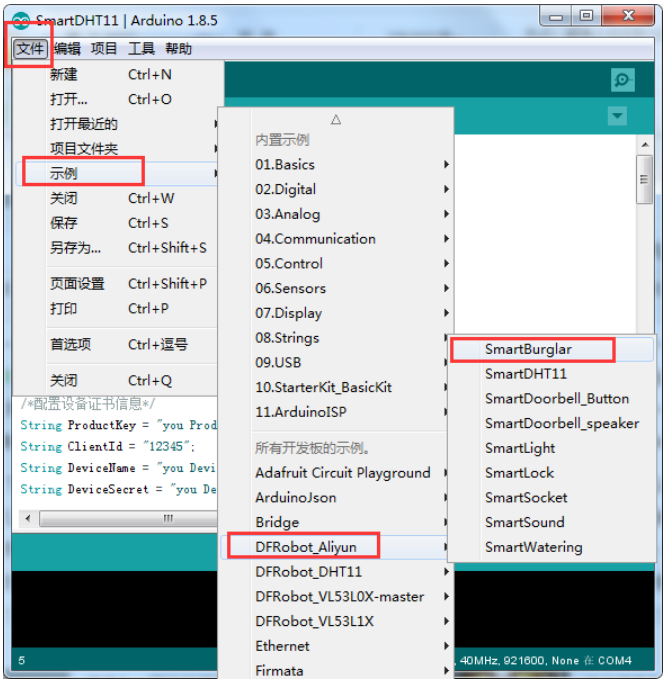
b. 配置方法选择**自定义**，消息类型选择“**text**”作为消息类型，并在 content 的双引号内填入信息“**家里有人闯入**”和“**家里安全**”。

(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。

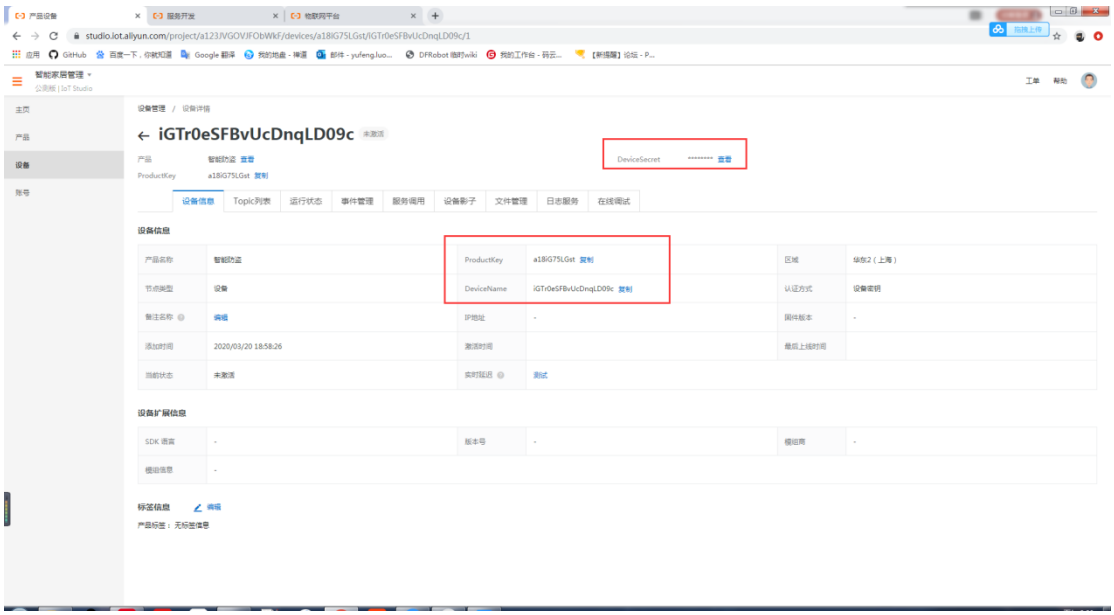
8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartBurglar** 项目，如下图所示：



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";
```

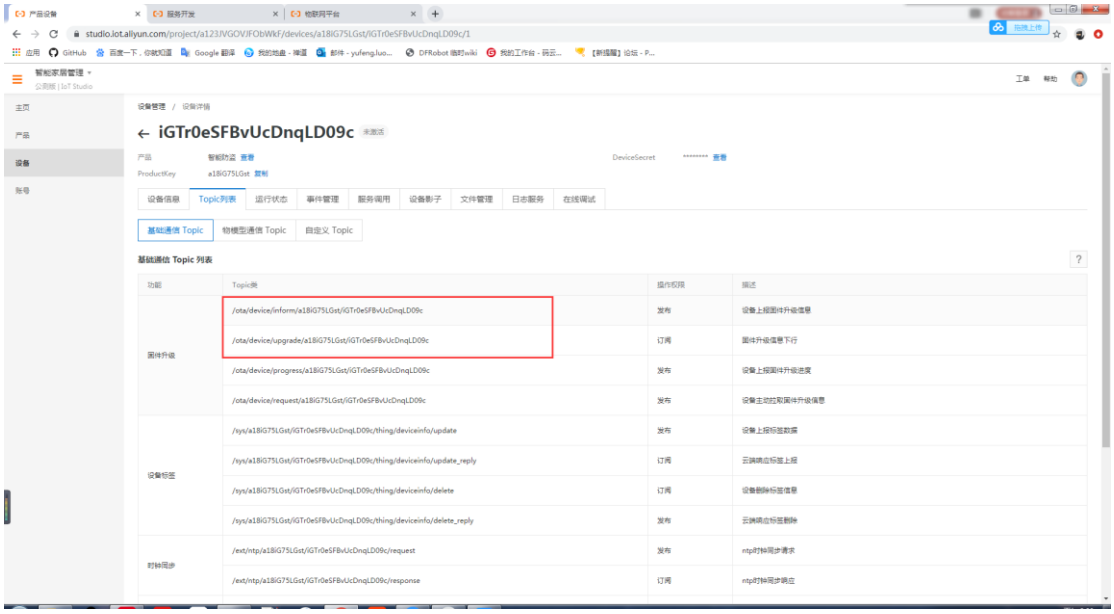
② 填写设备证书信息。获取设备信息，如下图所示：



将获取到的设备证书信息，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。

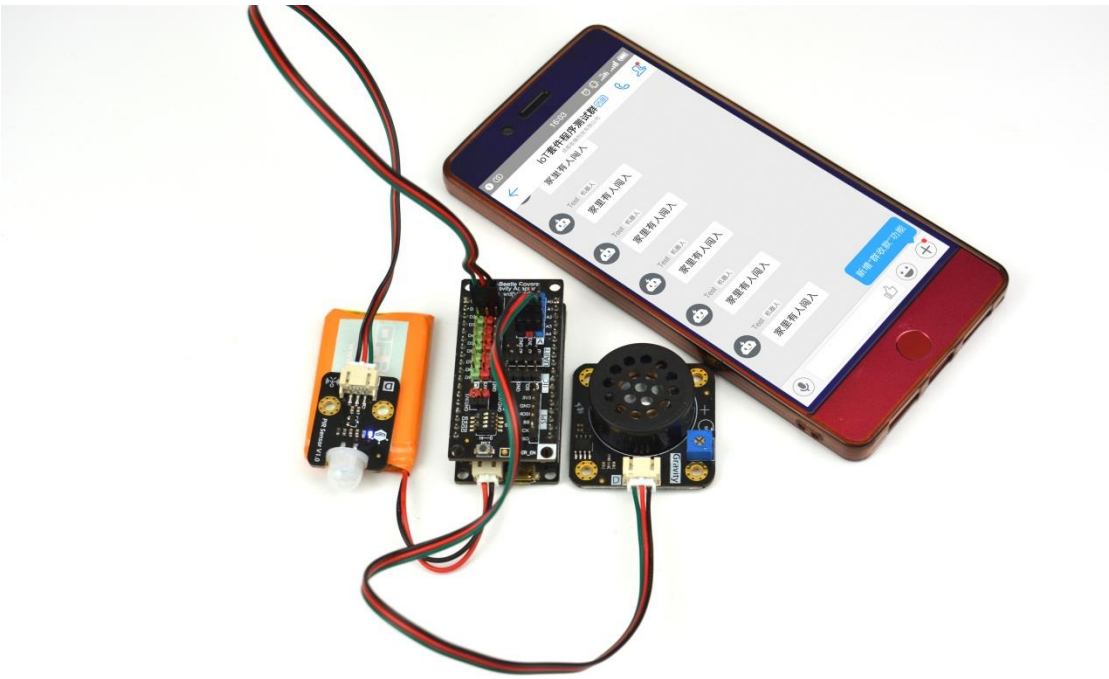
```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

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	防盗警报	alarm	bool (布尔型)	布尔值：无人 - 0；有人 - 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

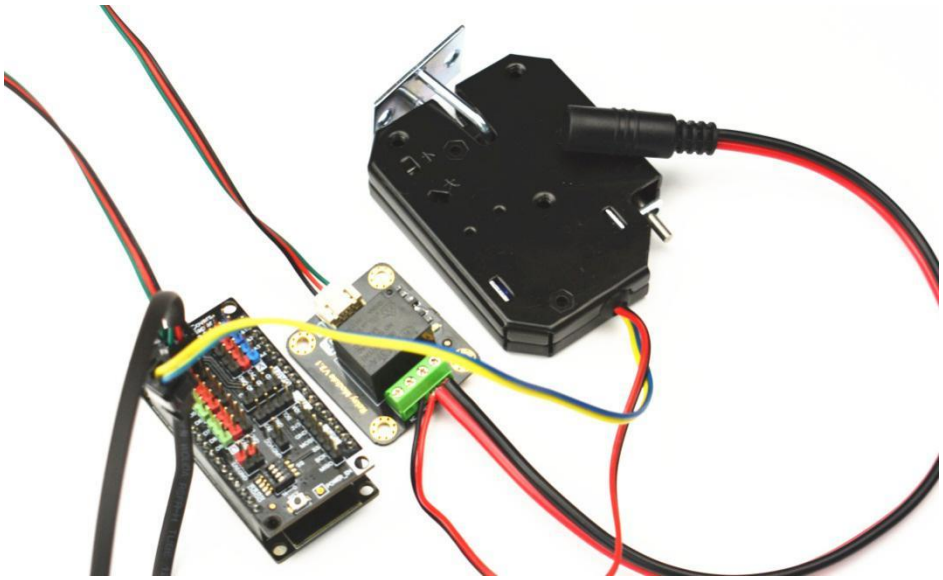
FireBettle Gravity 扩展板 x1

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。



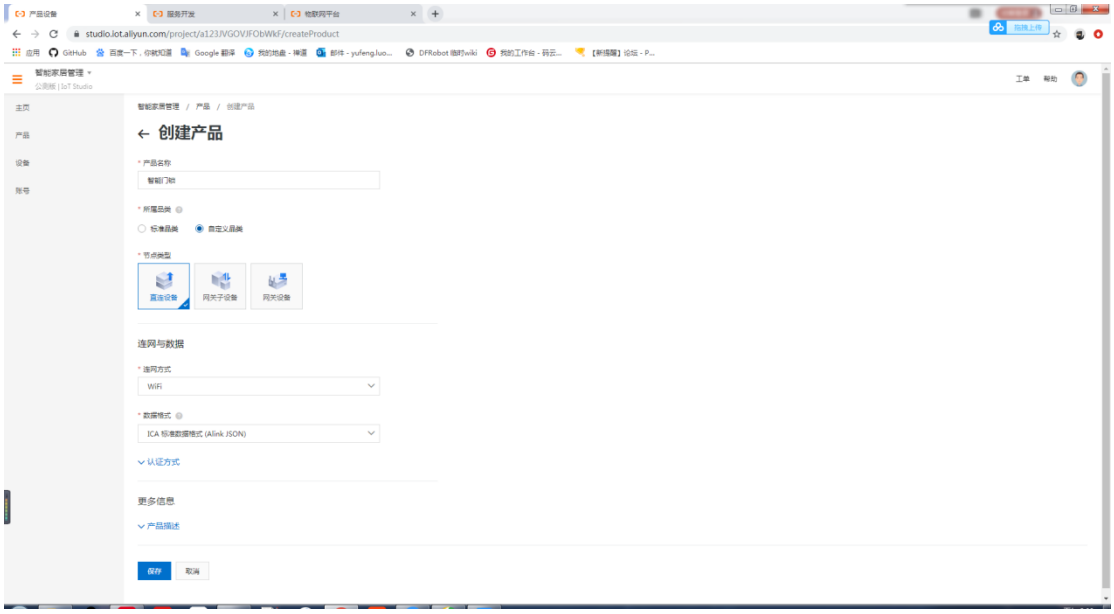
应用分析：智能门锁项目需要创建 2 个不通的开发服务，即移动应用开发来控制继电器通断，钉钉服务开发“门没关好！！”的消息推送至钉钉。其次将智能防盗产品的设备的

证书信息、Topic 列表，产品标识符替换到 Aruduino 程序的相应位置中。

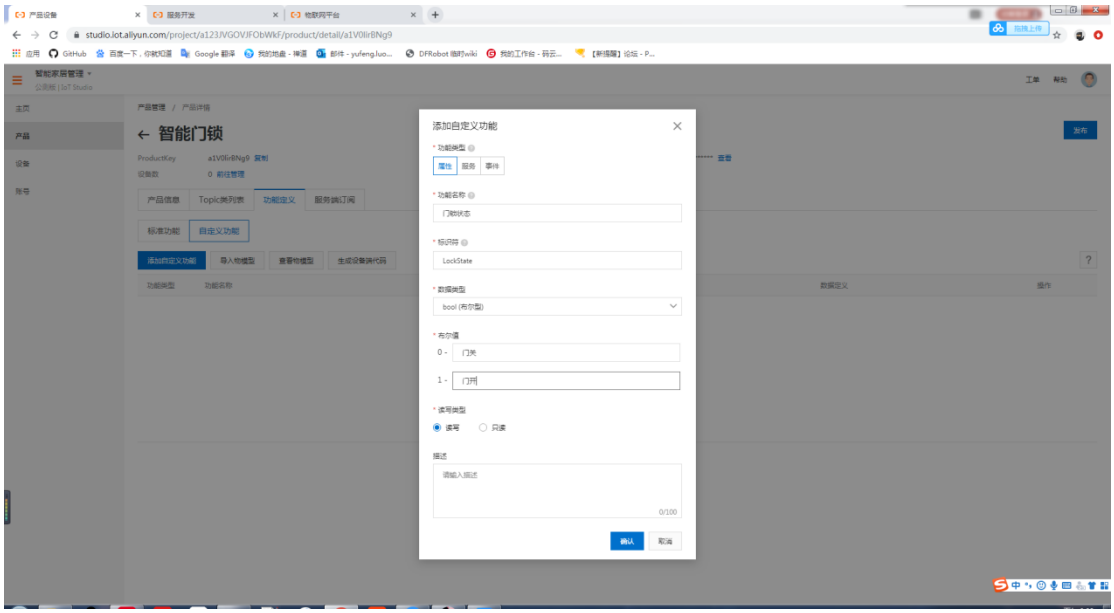
制作步骤

1.打开浏览器，输入网址 https://iot.aliyun.com,登录并进入**物联网平台**；

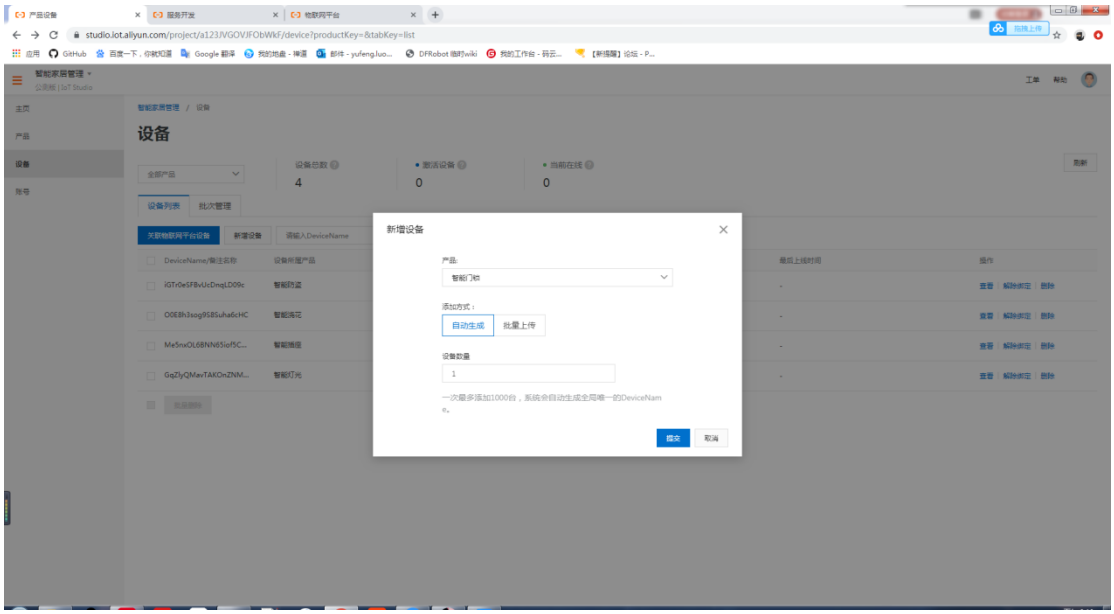
2.创建一个名为**智能门锁**的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



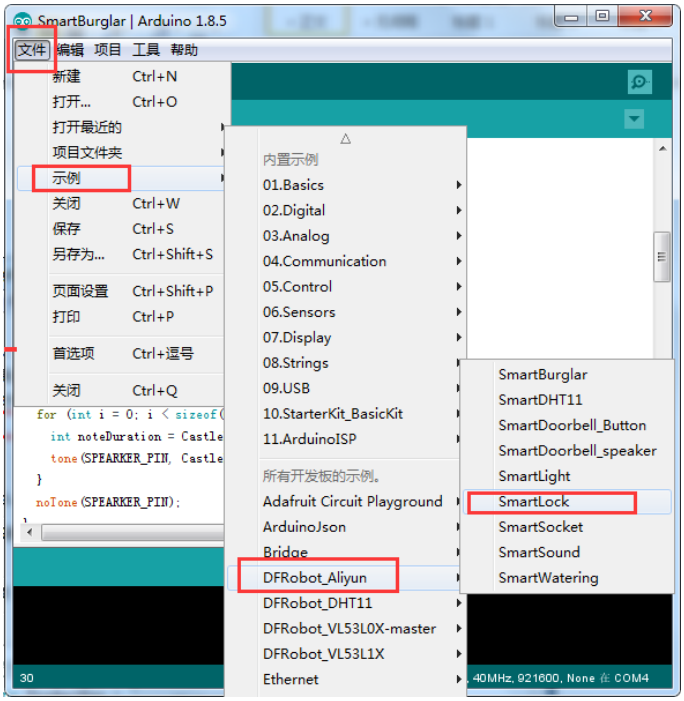
3. 为智能门锁产品定义一个功能名为“**门锁状态**”的属性，自定义一个产品标识符，定义门开为 1，门关为 0。如下图所示:



4.为**智能门锁**添加一个名为 **Lock** 的设备，如下图所示：



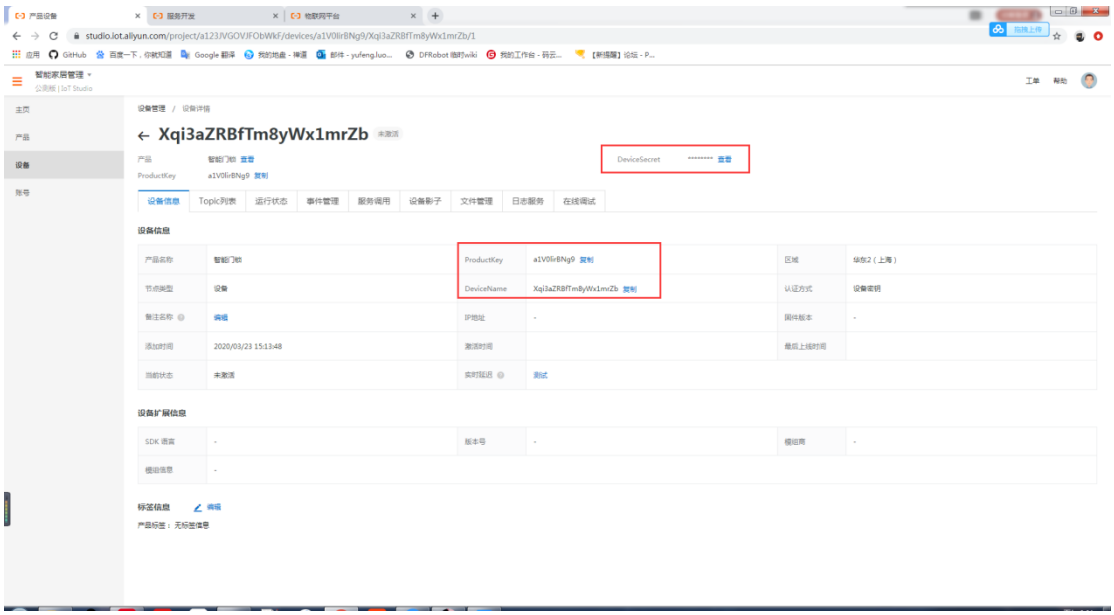
5.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartLock** 项目，如下图所示：



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";
```

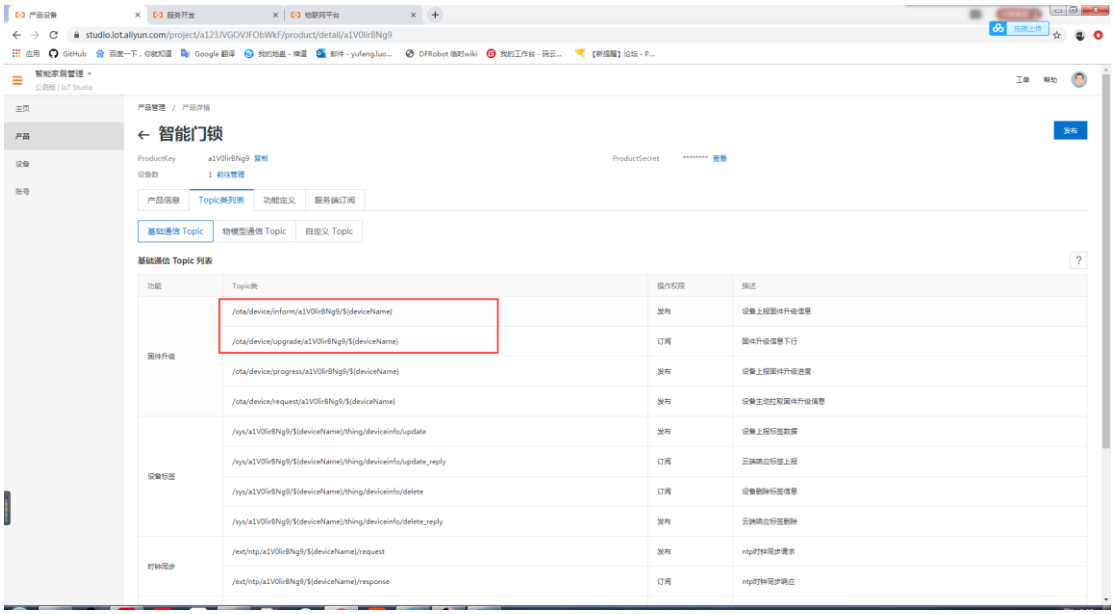
② 填写设备证书信息。获取设备 **Lock** 的设备信息，如下图所示：



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。

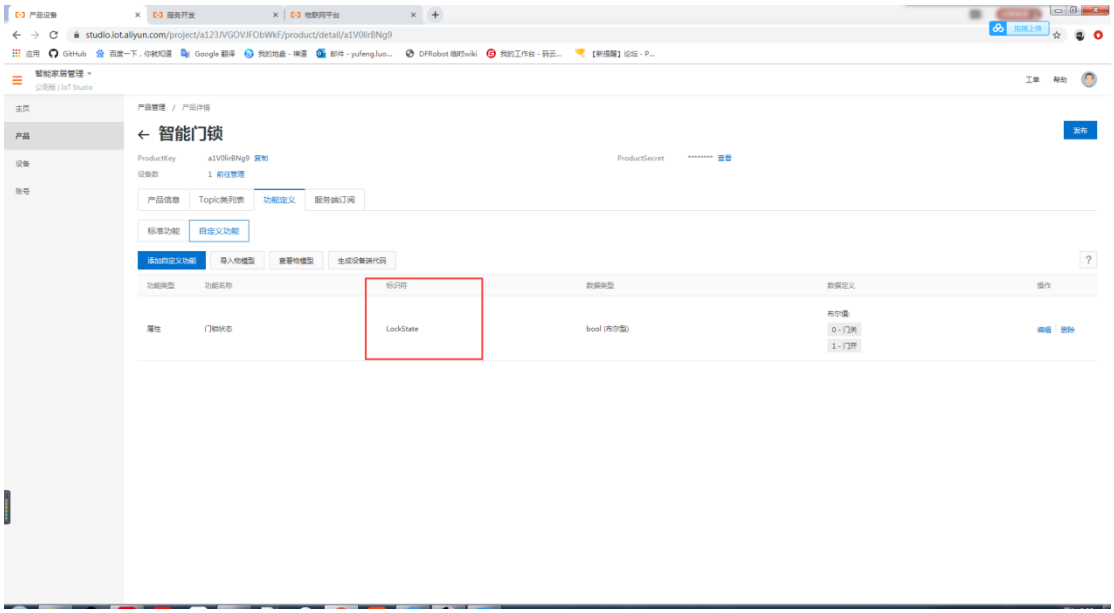
```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

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。



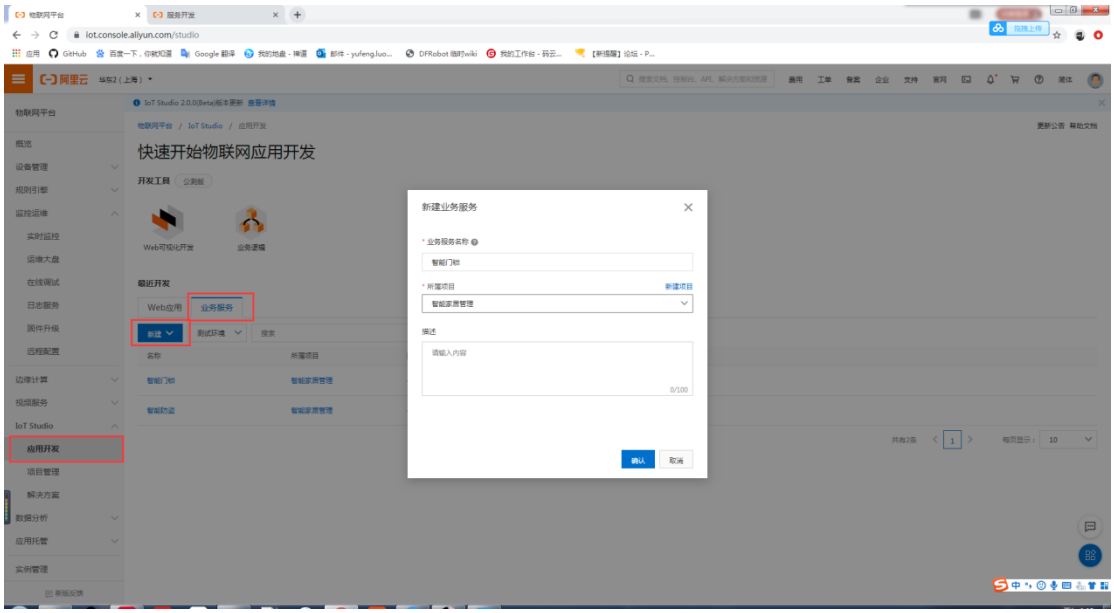

```
/*需要操作的产品标识符*/
String Identifier = "you Identifier";
```

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。

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

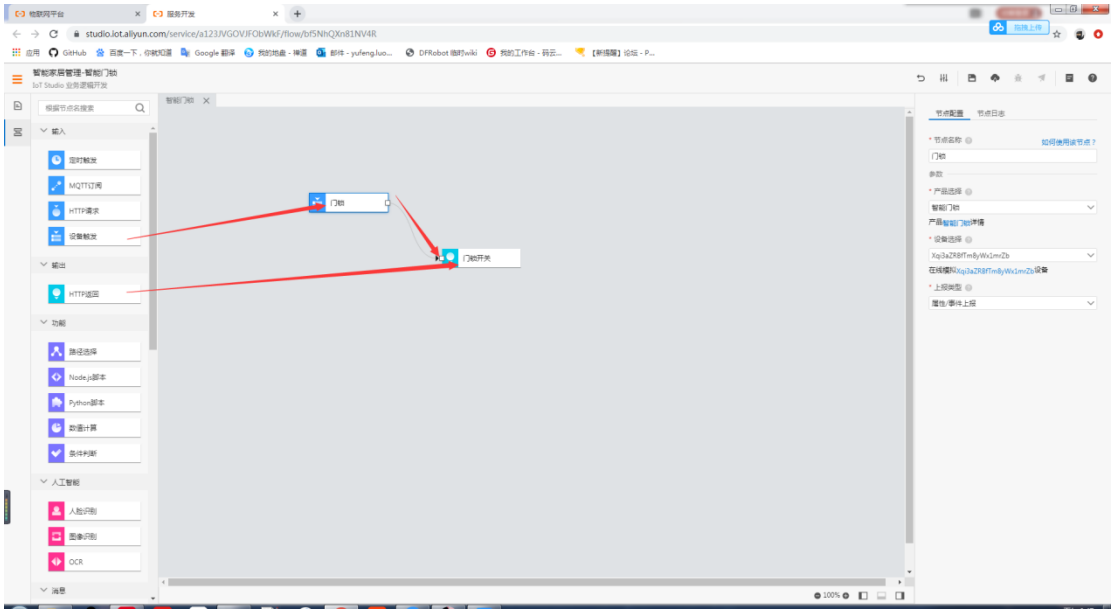
根据页面提示，创建一个名为“**智能门锁**”的服务，该服务内容为当检测到门锁开了后，将信息推送至钉钉，如下图所示:



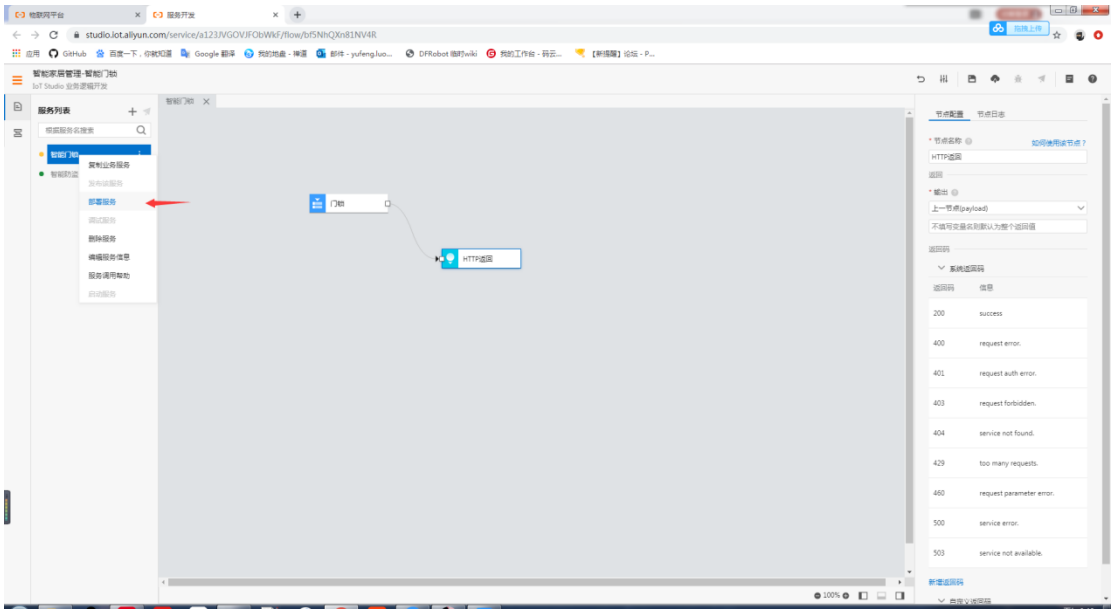
7.点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址:

https://linkdevelop.aliyun.com/studioservice-doc#index.html

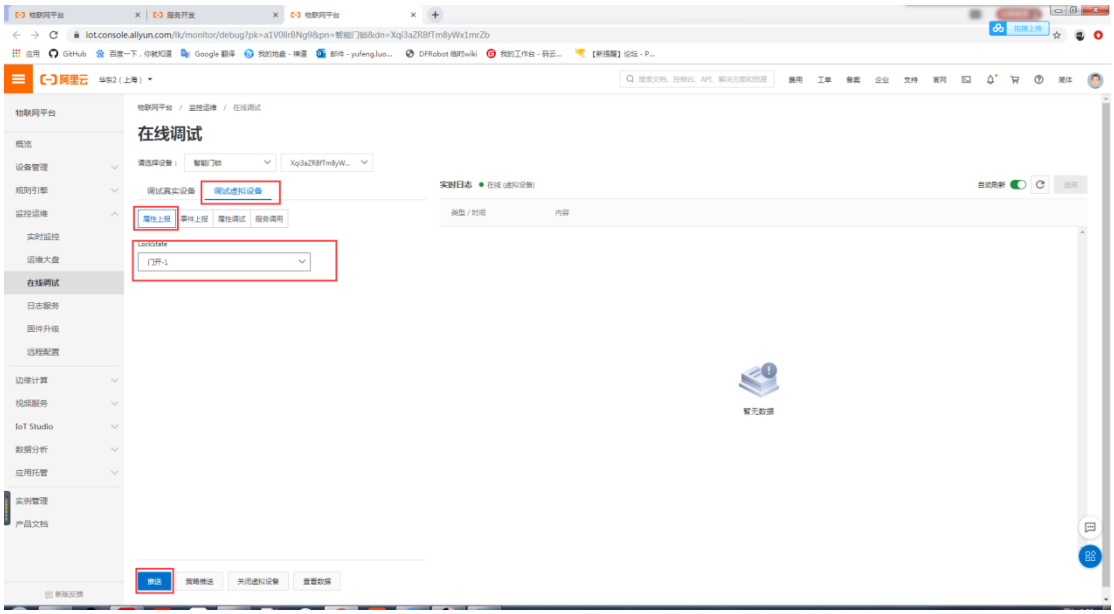
❶ 点击左侧导航栏的**节点**，在节点列表中选择**设备->设备触发**，并将其拖动到服务画布内，把这个设备触发节点改为**门锁**；右侧栏配置项分别选择“智能门锁”产品，创建的设备名以及选择监听“属性或事件上报”，如下图所示:



❷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**功能->HTTP 返回-**，右键点击**智能门锁->部署->启动->调试**，如下图所示:

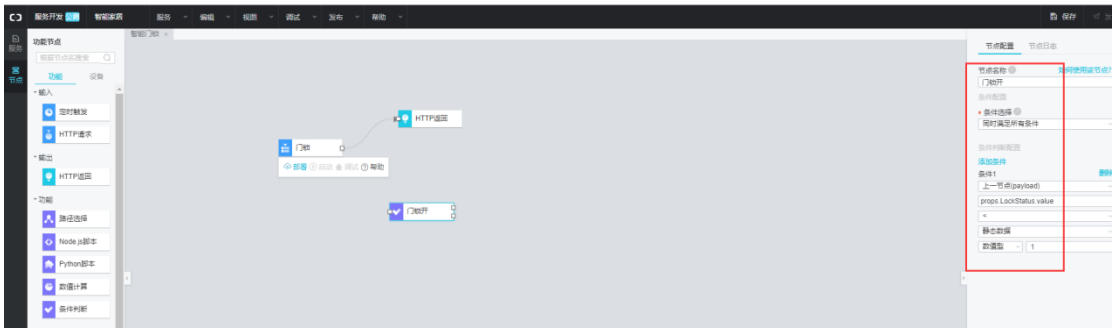


在弹出的“在线调试”页面中选择某个参数，点击推送，回到服务开发页面，点击节点日志，即可看到“门锁”节点的输入输出节点信息，如图：

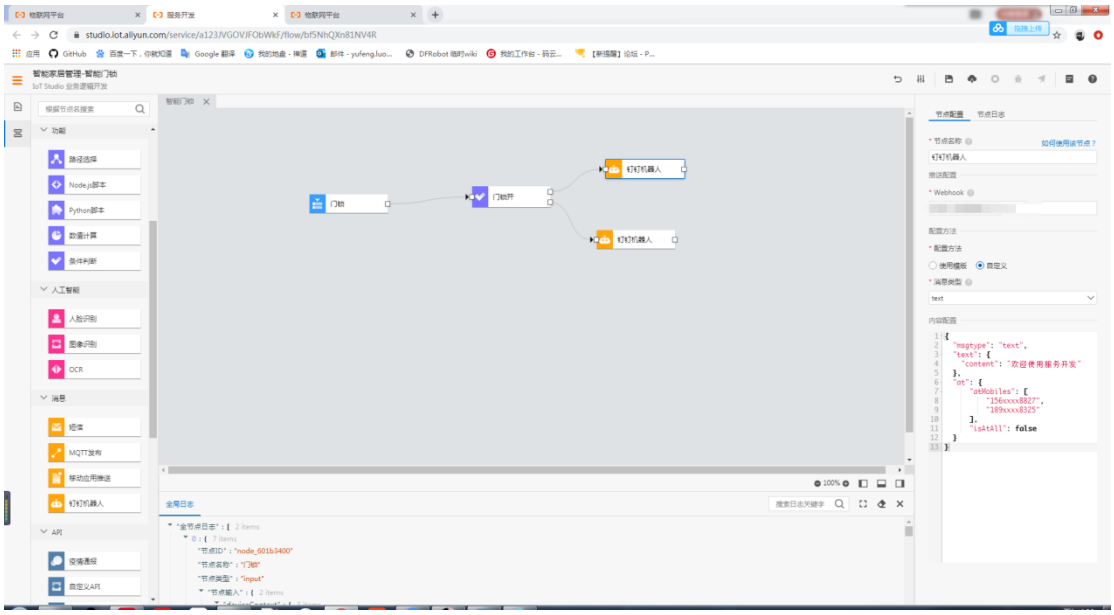


❸添加一个“**条件判断**”节点，条件节点相当于一个 if-else 判断。选中左侧节点列表的功能->条件节点，将其改为“**门锁开**”，并根据上面得到的“门锁”输出节点信息配置“门锁开”节点的输入设置，将条件 1 配置为“**上一个节点**”+“**props.LockStatus.value**”；选

择<,**静态数据**，**数值型**，最后一个框填入 **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?**



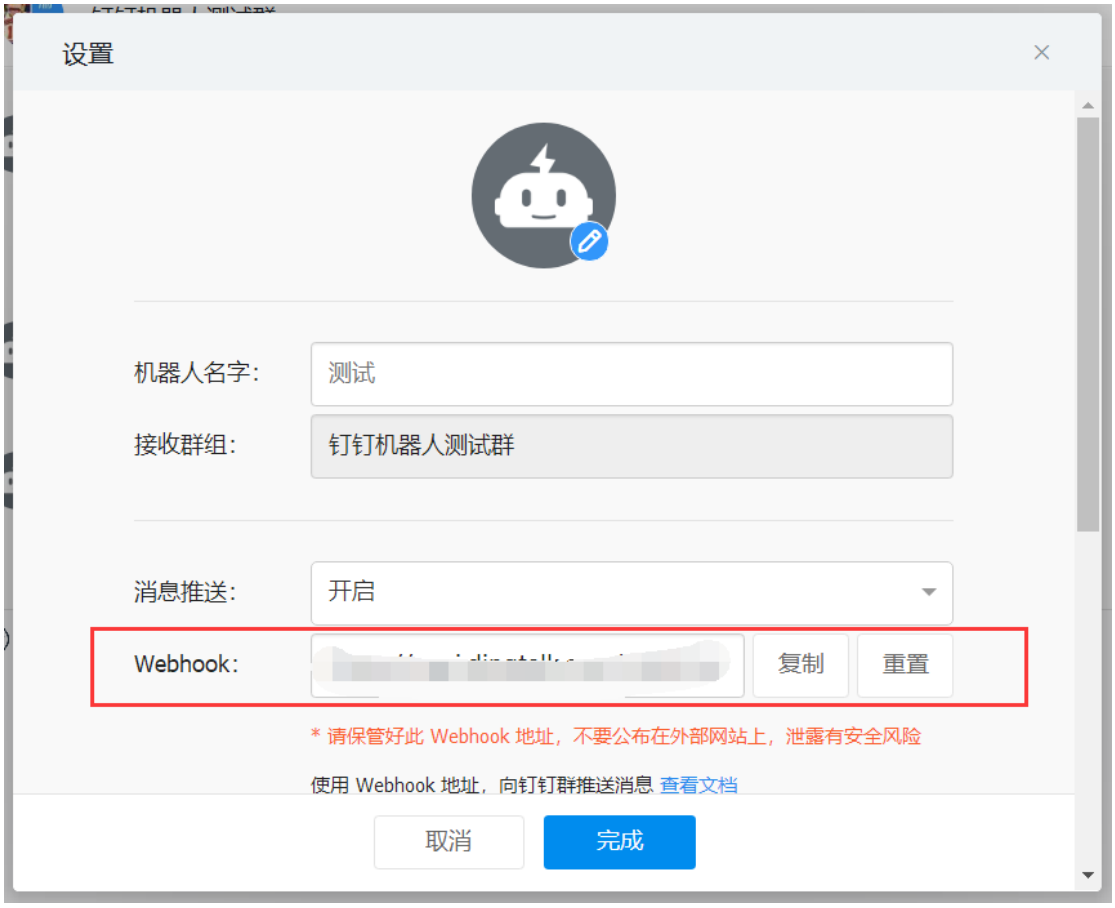
❹删除“HTTP 返回”节点，选中消息->钉钉机器人，添加两个钉钉机器人节点。并将门锁、门锁开、钉钉机器人节点连接起来。



❺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?**

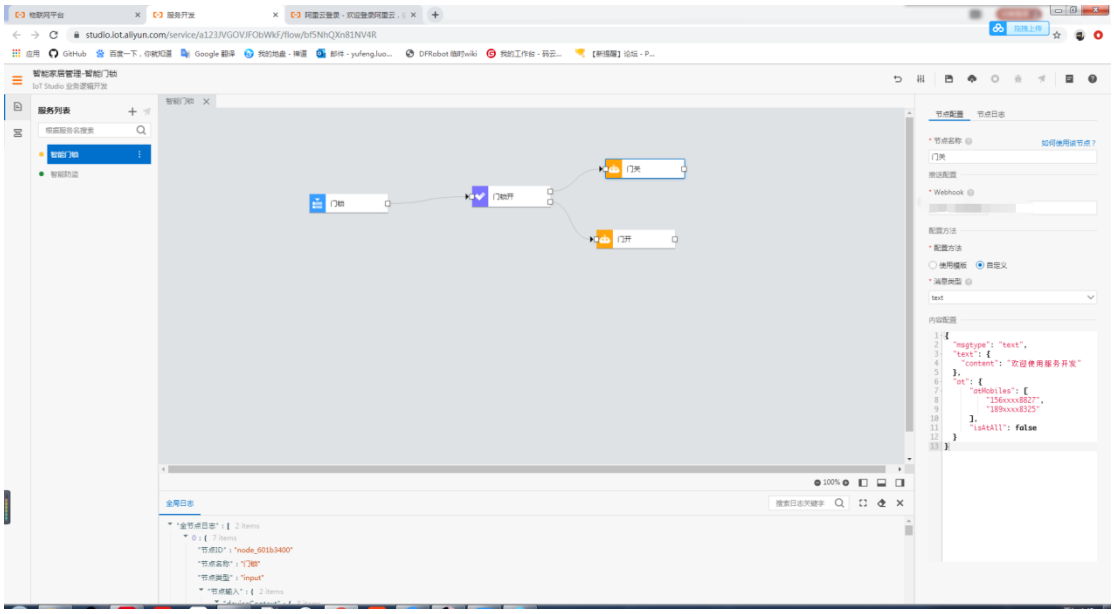
a.获取 Webhook。在获取之前，你需要在钉钉群中，添加一个钉钉机器人(注意查看钉钉机器人的 Webhook 时只能用电脑端的钉钉)，具体添加方法请参考**如何使用该节点？->如何配置钉钉机器人节点?** 最终获取的 Webhook，复制粘贴至右侧栏的 Webhook 处，如

下图所示：

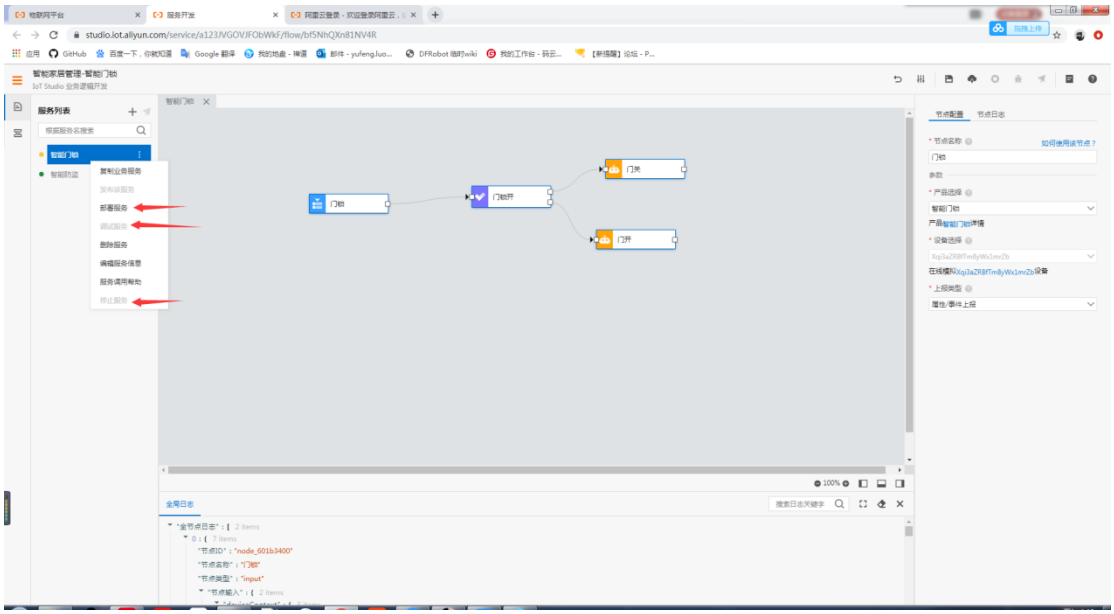


定义安全关键词：

c. 将满足条件的钉钉机器人节点名称改为**门关**，配置方法选择**自定义**，选择“**text**”作为消息类型，并在 content 的双引号内填入信息“**门关好了**”。将不满足条件的钉钉机器人节点名称改为**门开**，配置方法选择**自定义**，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**”

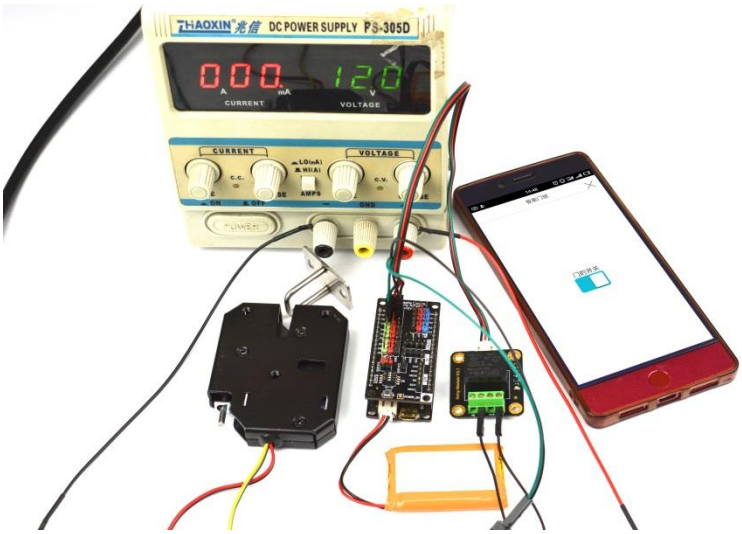


(4) 点击设备触发节点的**部署**按钮，点击**启动**，如下图所示：



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

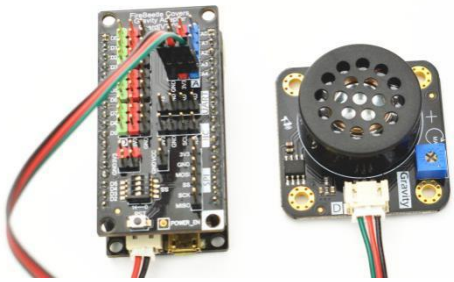
FireBettle Board-ESP32 x1

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。

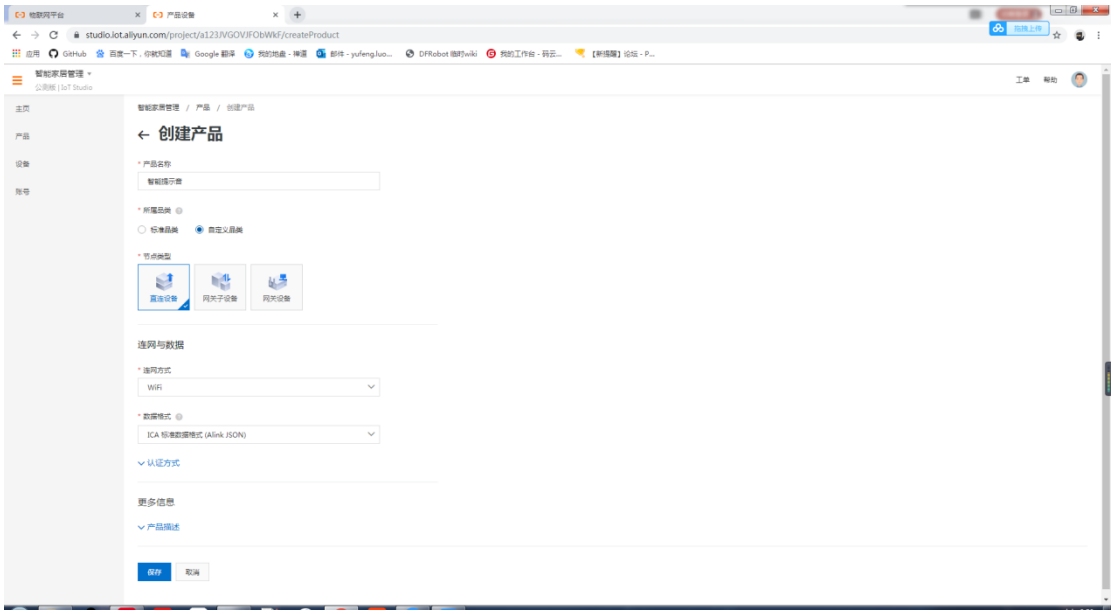


应用分析：此应用在创建产品、设备和服务的同时，还需要购买墨迹天气服务（免费购买）。

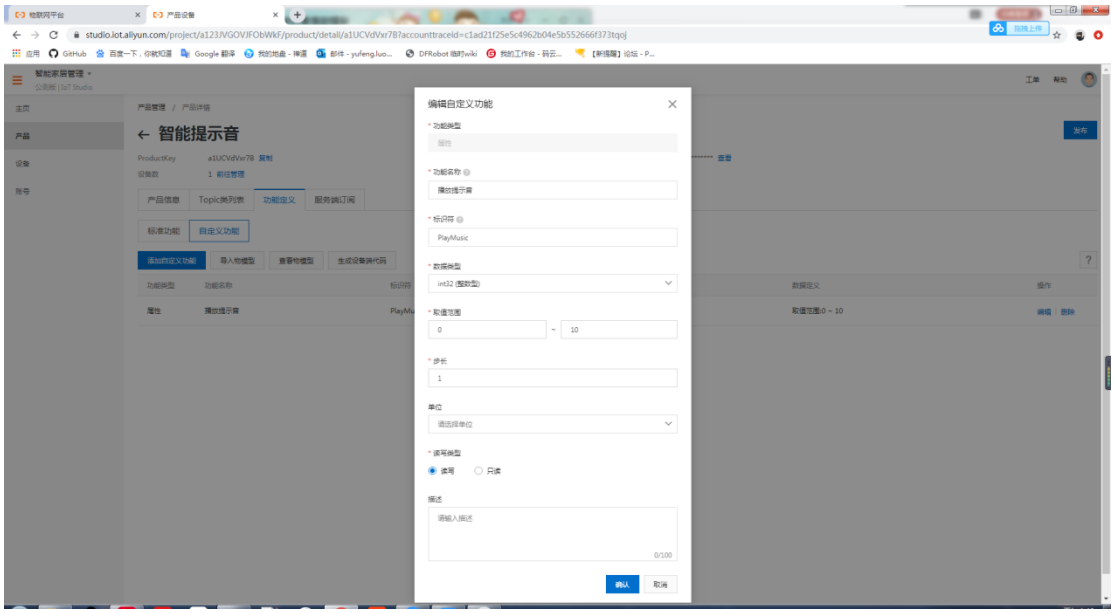
制作步骤

1.打开浏览器，输入网址 https://iot.aliyun.com,登录并进入**物联网平台**；

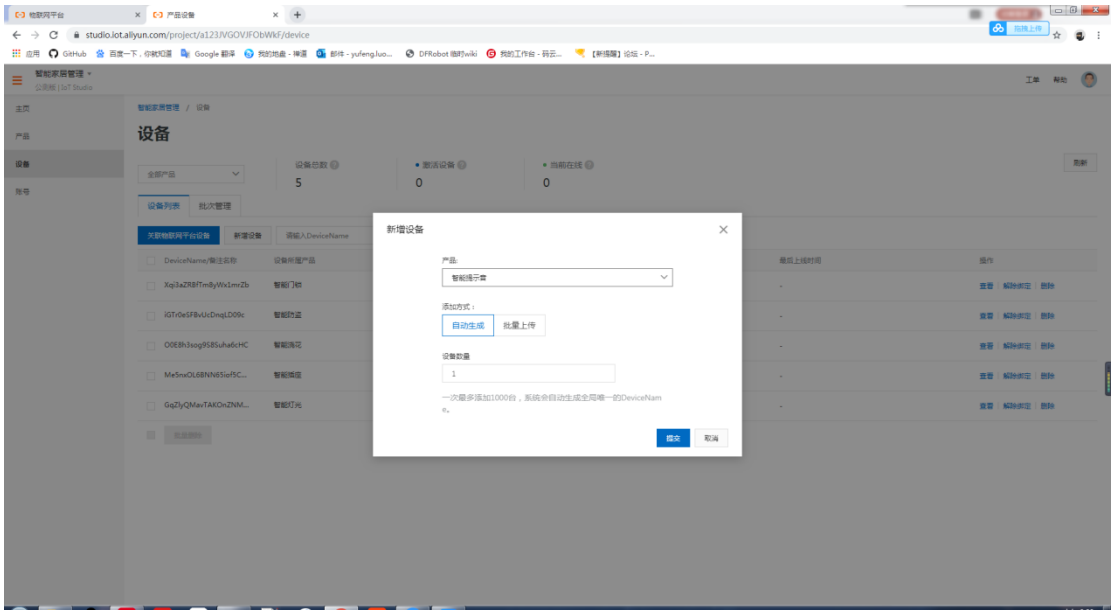
2.创建一个名为**智能提示音**的产品，如下图所示，若产品创建成功，会自动出现在产品列表中；



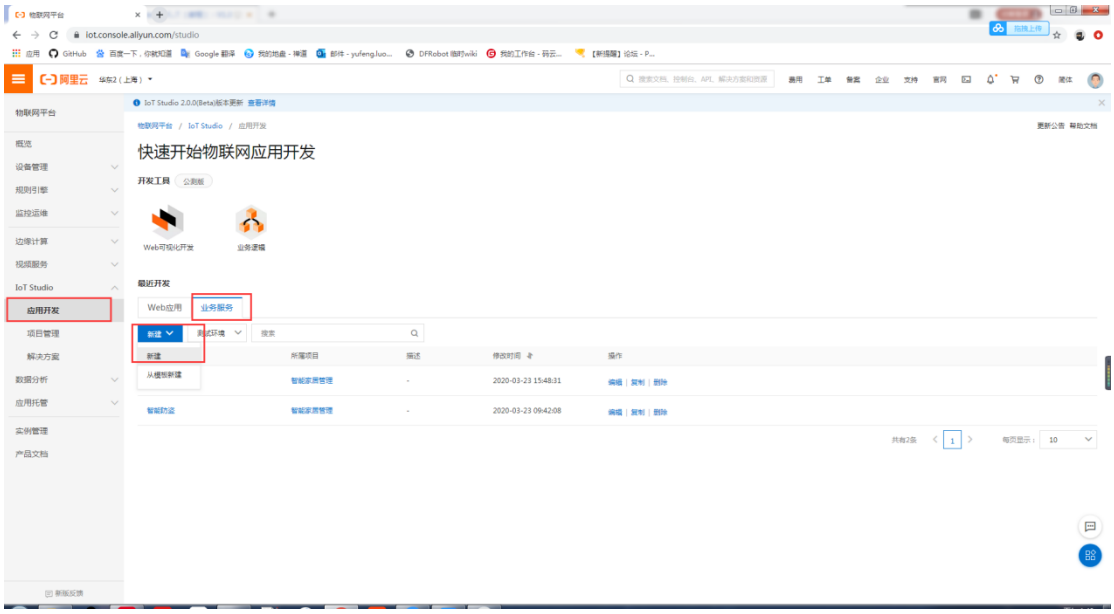
3. 为智能防盗产品定义一个功能名为“**播放提示音**”的属性，自定义产品标识符，定义其数据为整型，取值 0~10，步长为 1，如下图所示：

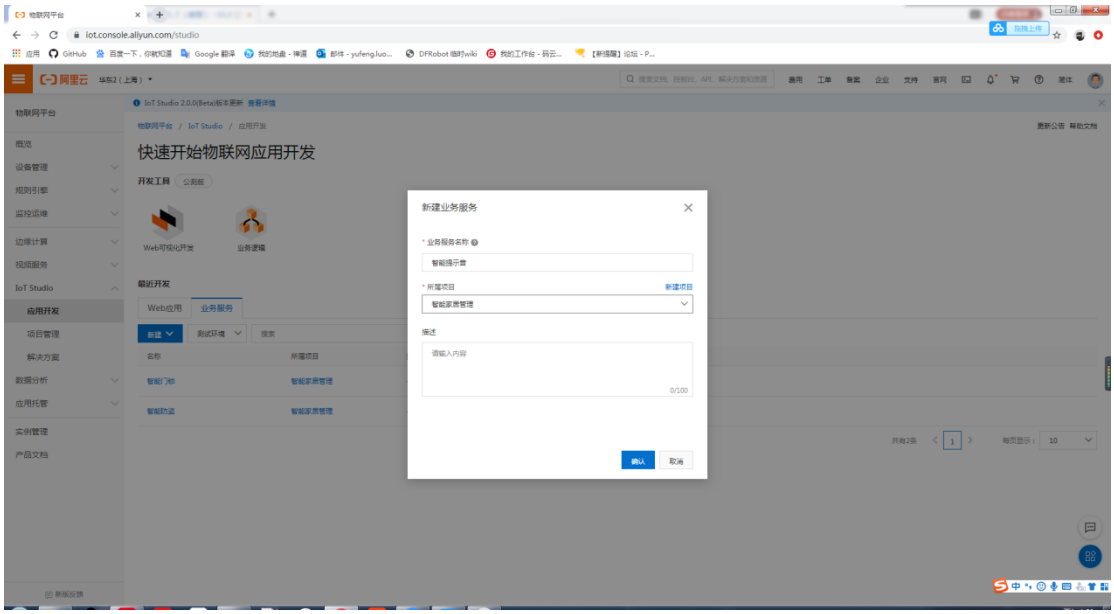


4.为**智能提示音**添加一个设备，如下图所示：



6. 选中左侧导航栏的 **IoT Studio->应用开发**,点击**业务服务**，再点击**新建**，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：

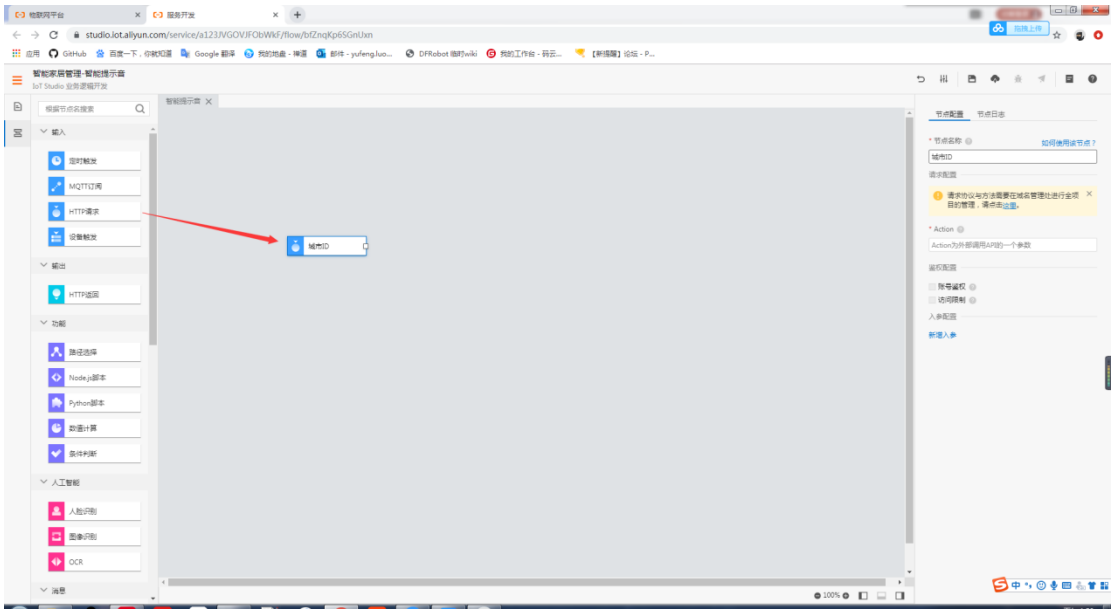




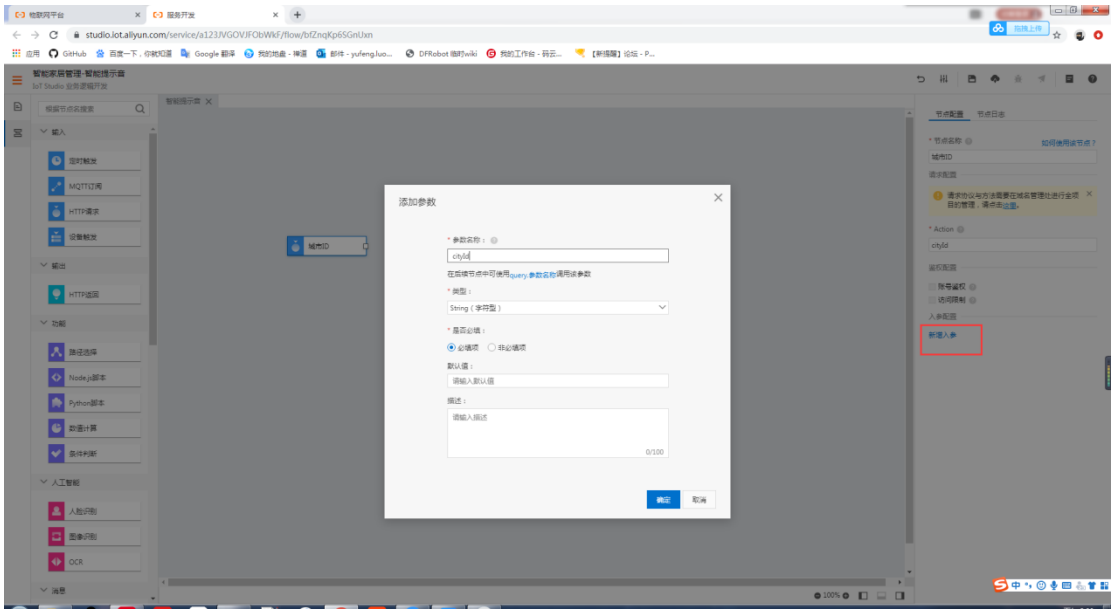
7.点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址:

https://linkdevelop.aliyun.com/studioservice-doc#index.html

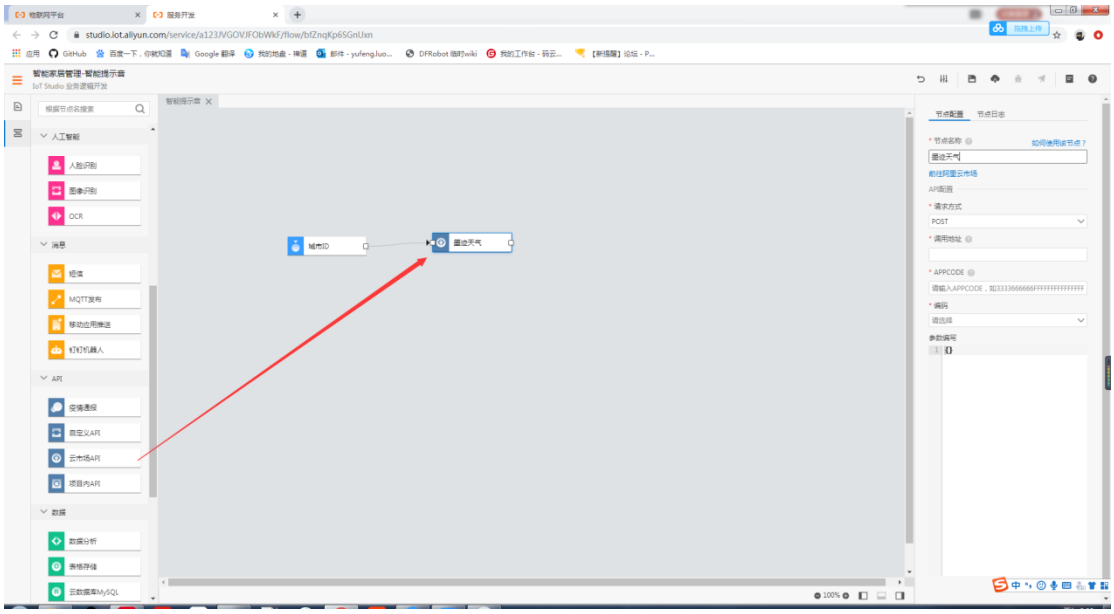
❶ 点击左侧导航栏的**节点**，在节点列表中选择**功能->HTPP 请求**，并将其拖动到服务画布内，把这个设备名称改为**城市 ID**；如下图所示:



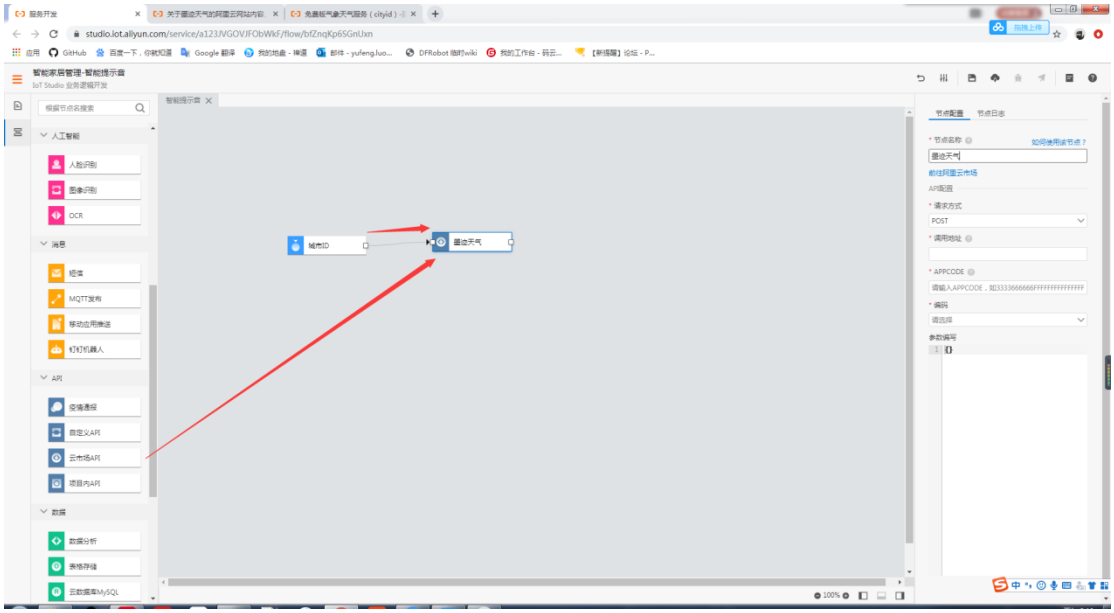
❷添加一个新增参数。由于后面查询天气的时候需要用到所在城市，因此 API 参数设定为字符“cityId”，配置如图所示。



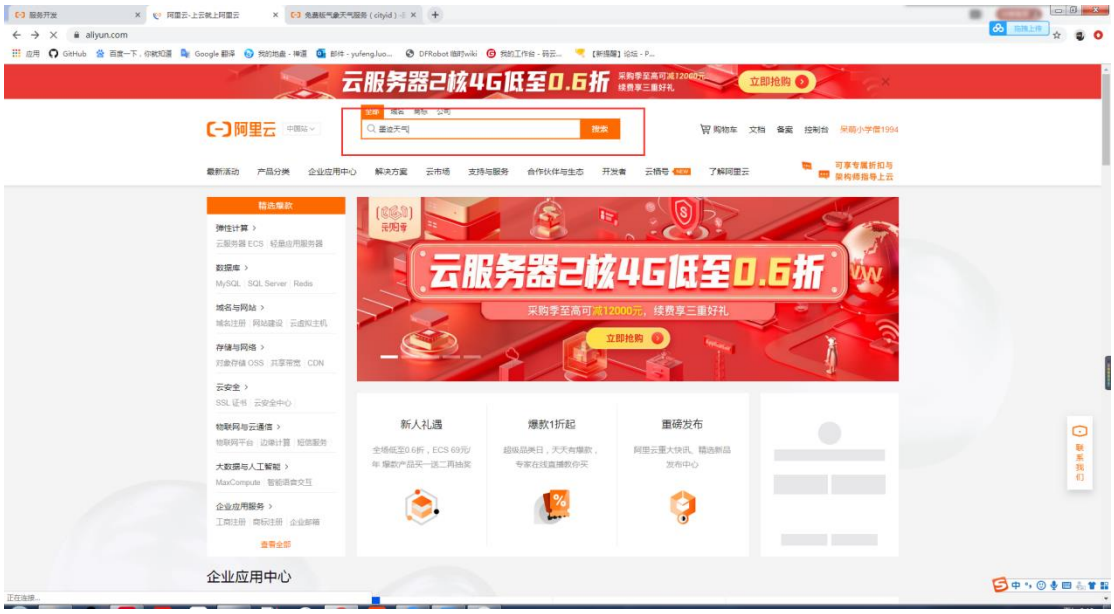
❸添加一个**云市场 API** 节点。选中 **API->云市场 API**。并将其改为**墨迹天气**，如下图所示：

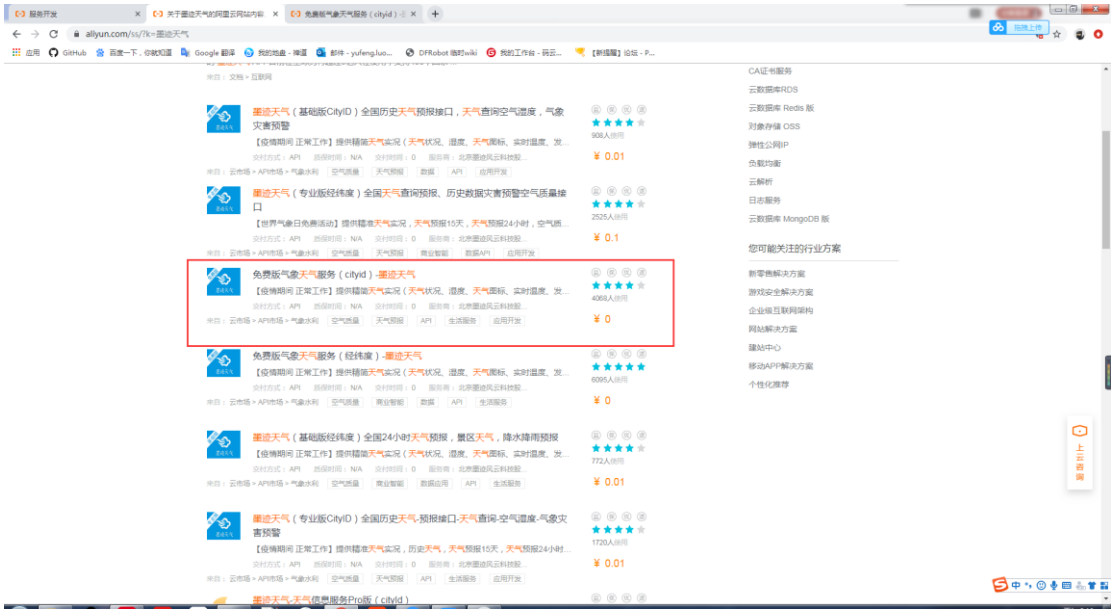


a. **购买墨迹天气服务。**打开新页面前往阿里云市场选择 API，购买墨迹天气服务，该服务是免费的，具体购买方式请参考阿里云 IoT 案例“天气加湿器”，也可以直接搜索墨迹天气购买，或者搜索免费版气象服务（cityid）-墨迹天气。

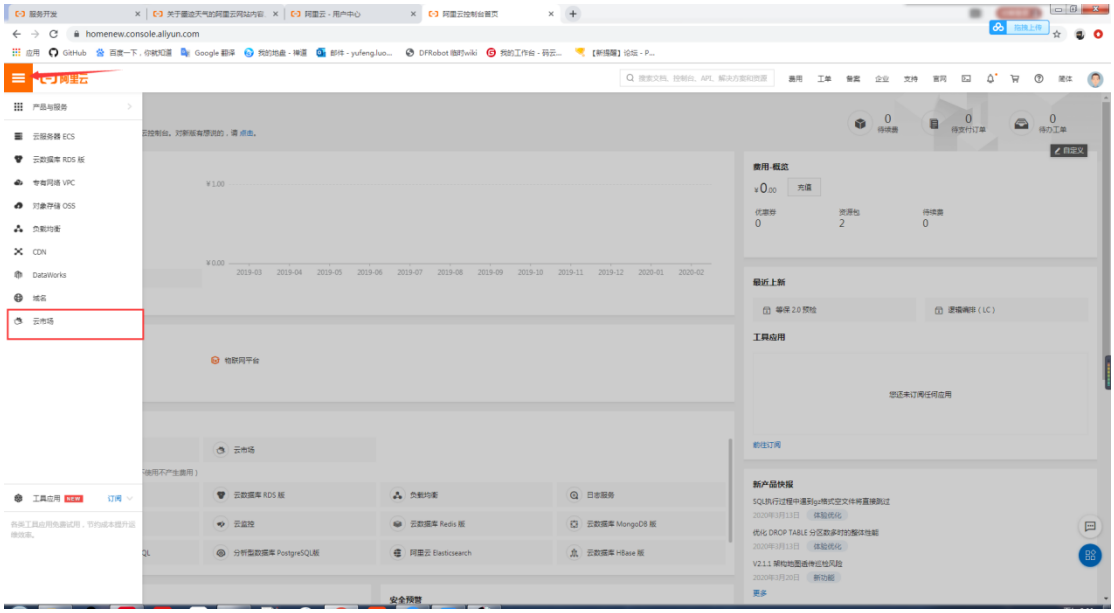


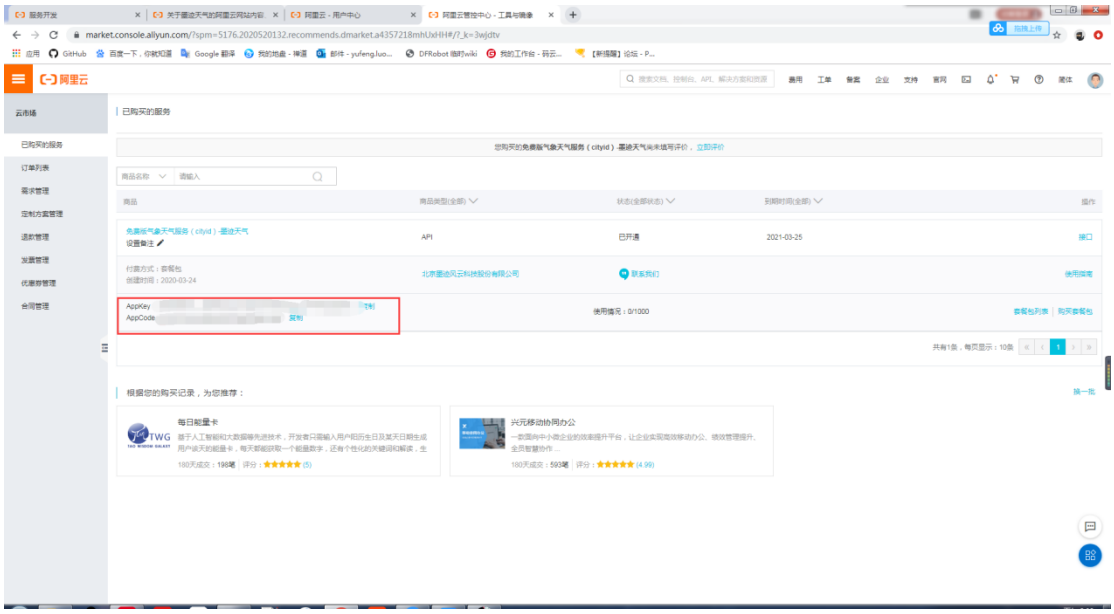
搜索免费版气象服务（cityid）-墨迹天气。


云服务器2核4G低至0.6折
立即抢购
阿里云
墨迹天气
搜索
云服务器2核4G低至0.6折
采购季至高可减12000元，续费享三重好礼
立即抢购
新人礼遇
爆款1折起
重磅发布
企业应用中心


墨迹天气（基础版CityID）全国历史天气预报接口，天气查询空气温度，气象灾害预警
墨迹天气（专业版经纬度）全国天气查询预报、历史数据灾害预警空气质量接口
免费版气象天气服务（cityid）-墨迹天气
免费版气象天气服务（经纬度）-墨迹天气
墨迹天气（基础版经纬度）全国24小时天气预报，景区天气，降水降雨预报
墨迹天气（专业版CityID）全国历史天气-预报接口-天气查询-空气温度-气象灾害预警
墨迹天气-天气信息服务Pro版（cityid）
您可能关注的行业方案

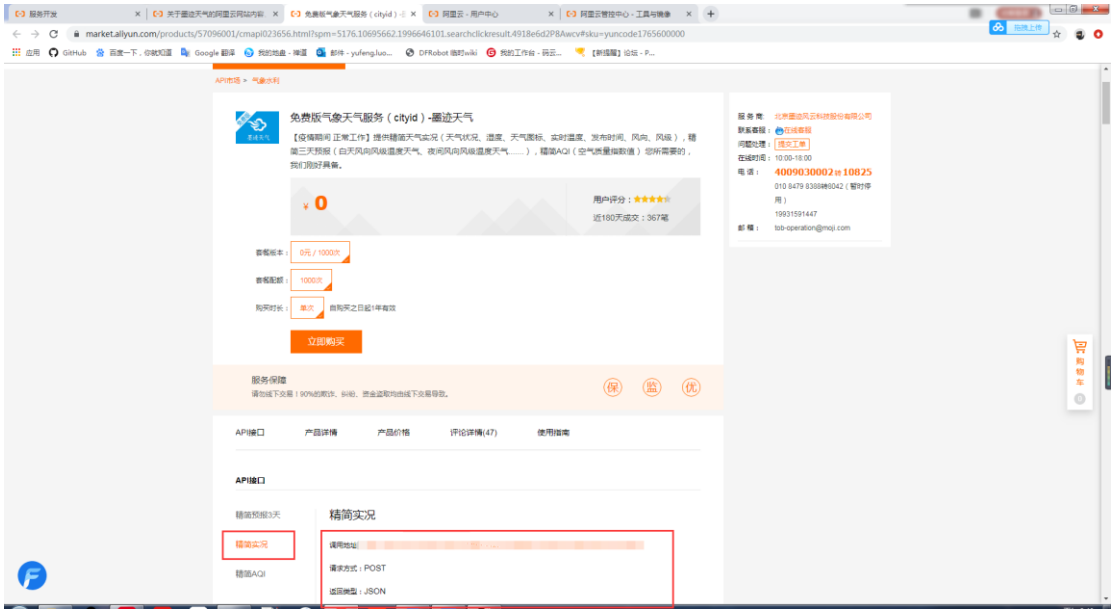
b. **获取 AppCode。墨迹天气**购买成功后，回到**物联网平台**，选中左侧导航栏的**产品与服务**，搜索“**云市场**”即可找到 **AppCode**。如下图所示：



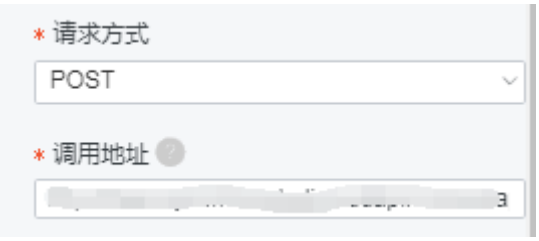


复制该 AppCode 值到墨迹天气的配置栏“AppCode”即完成了 AppCode 的配置，如下图所示：

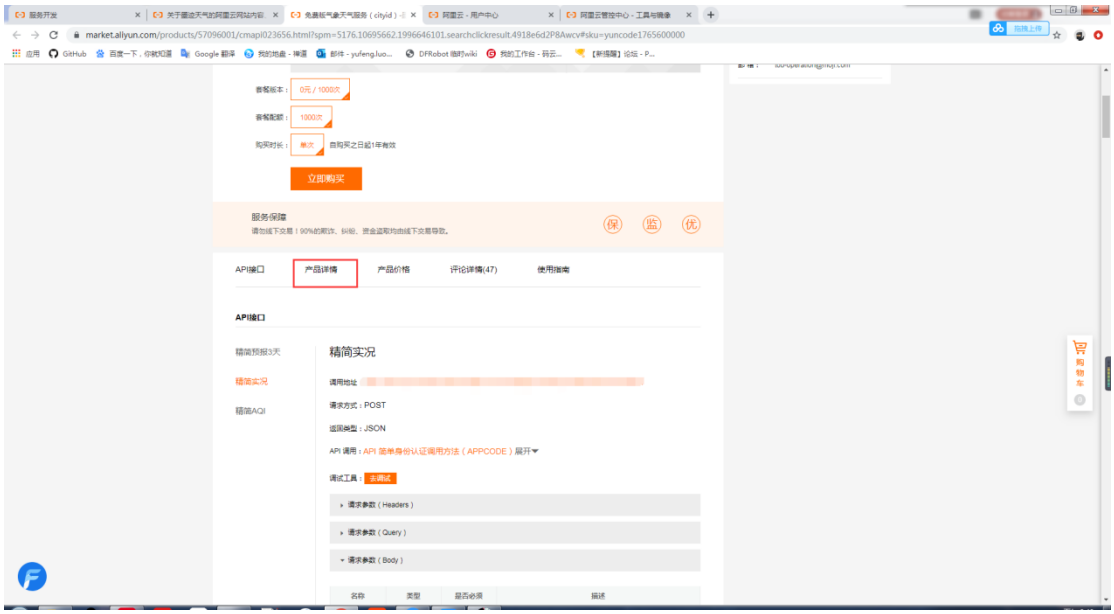
c.获取精简实况的调用地址和请求方式。回到刚才的**云市场**页面，点击“免费版气象服务（cityid）-墨迹天气”，进入购买页面，下拉点击**精简实况**，即可获取**精简实况的调用地址和请求方式**，如下图所示：

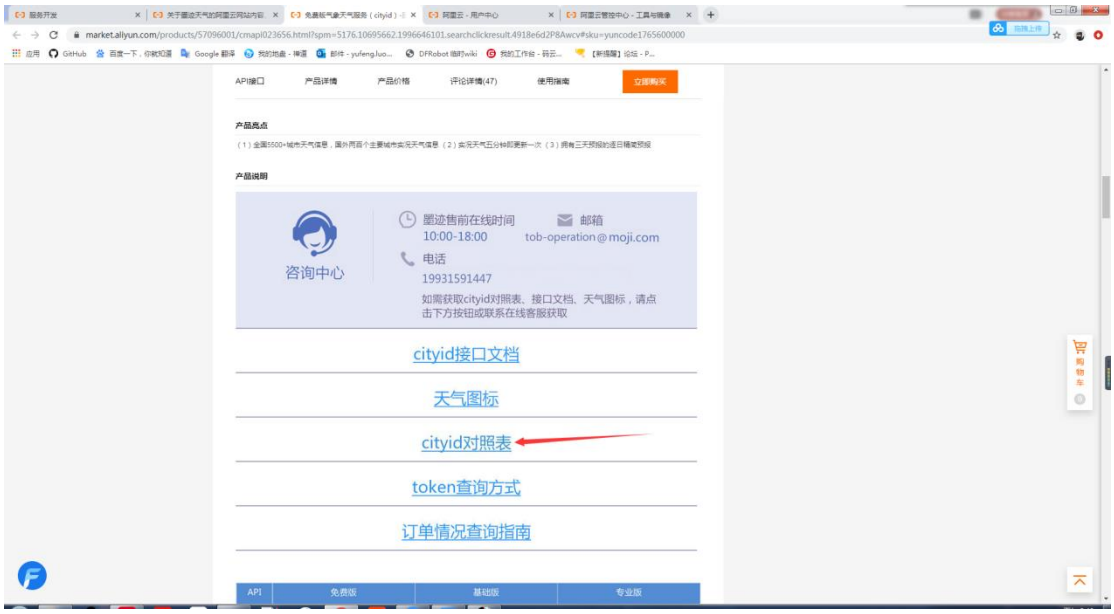


复制该调用地址即可配置“墨迹天气”的调用地址，并将该请求方式改为“POST”，如下图所示：



c. **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**，即可获取各个城市的城市 ID，(请用 excel 打开下载的文件)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：

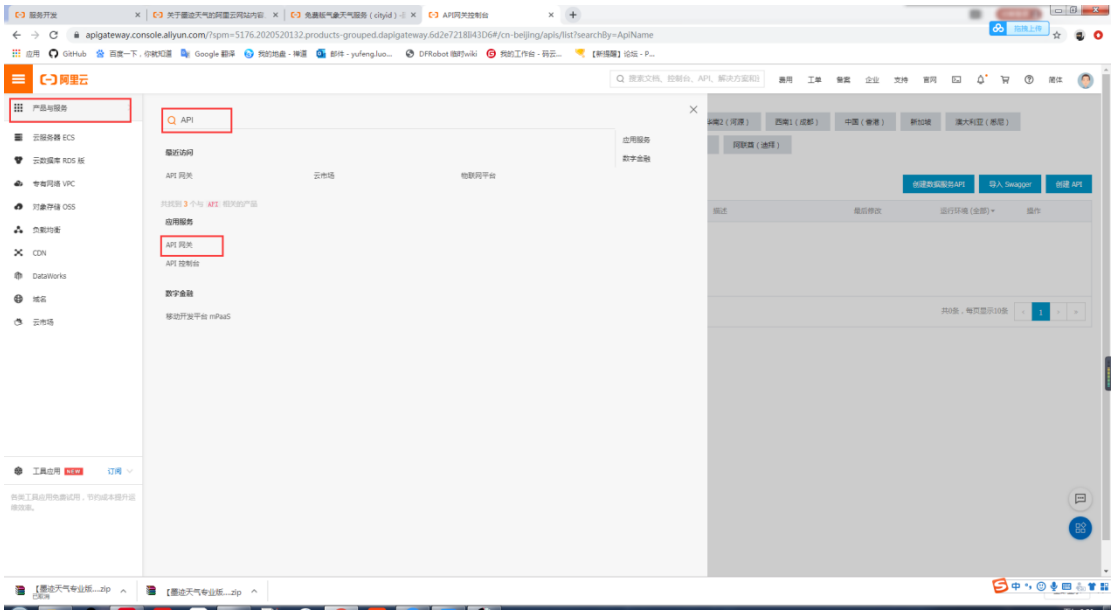


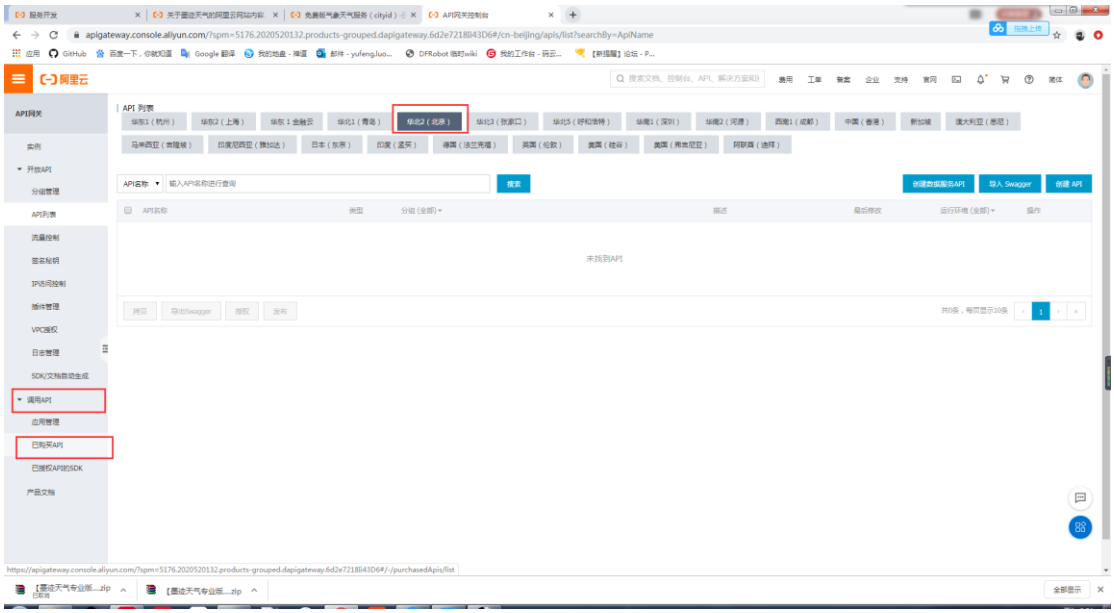


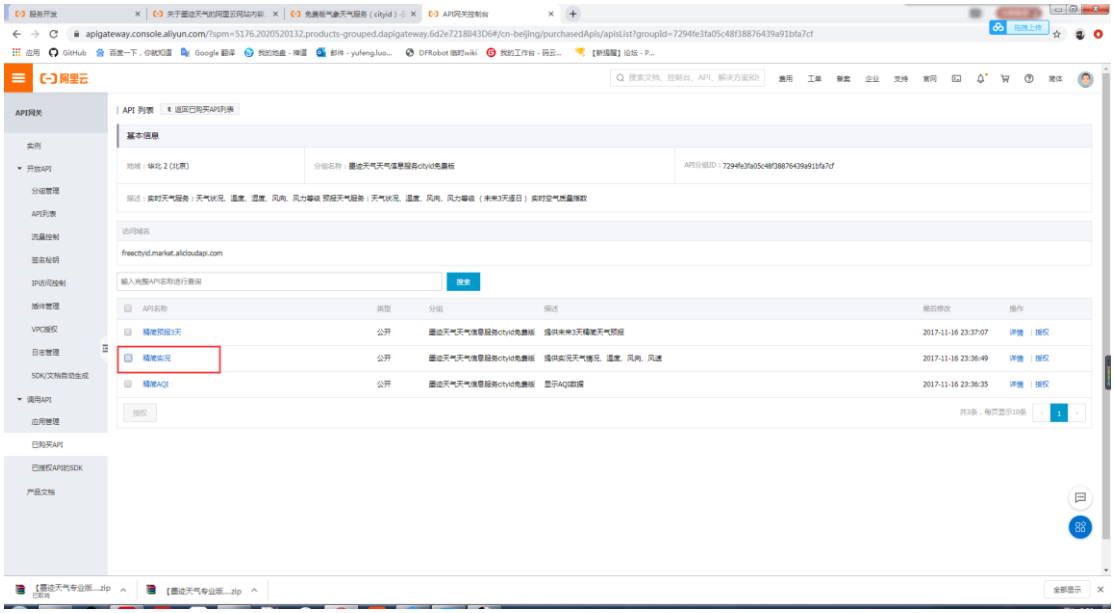
如成都温江的 ID 为 2637。

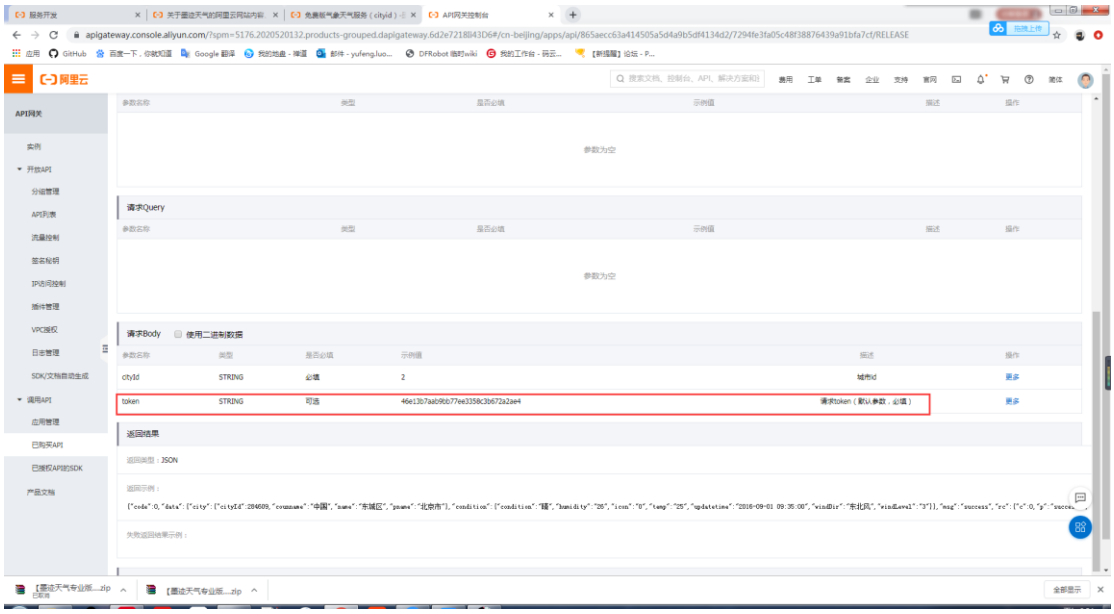
2151	2632	新疆维吾尔	阿勒泰地区	阿勒泰地区	Altay Prefec	Aletaidiqu
2152	2635	四川省	成都市	成都市	Chengdu	chengdushi
2153	2636	四川省	成都市	新都区	Xindu Distric	xinduqu
2154	2637	四川省	成都市	温江区	Wenjiang D	wenjiangqu
2155	2638	四川省	成都市	都江堰市	Dujiangyan	dujiangyanshi
2156	2639	四川省	成都市	彭州市	Pengzhou	pengzhoushi
2157	2640	四川省	成都市	邛崃市	Qionglai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

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：

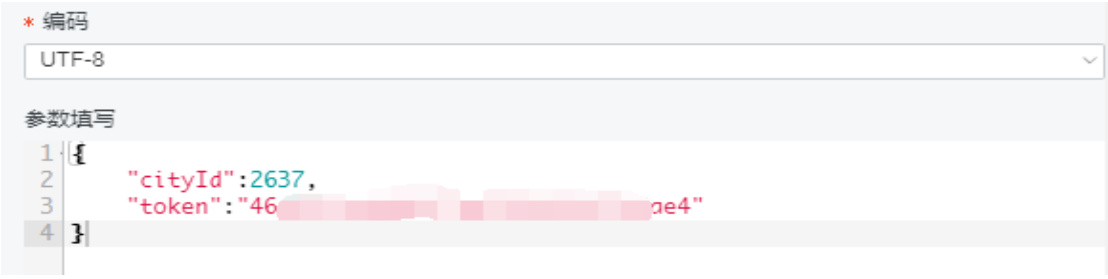






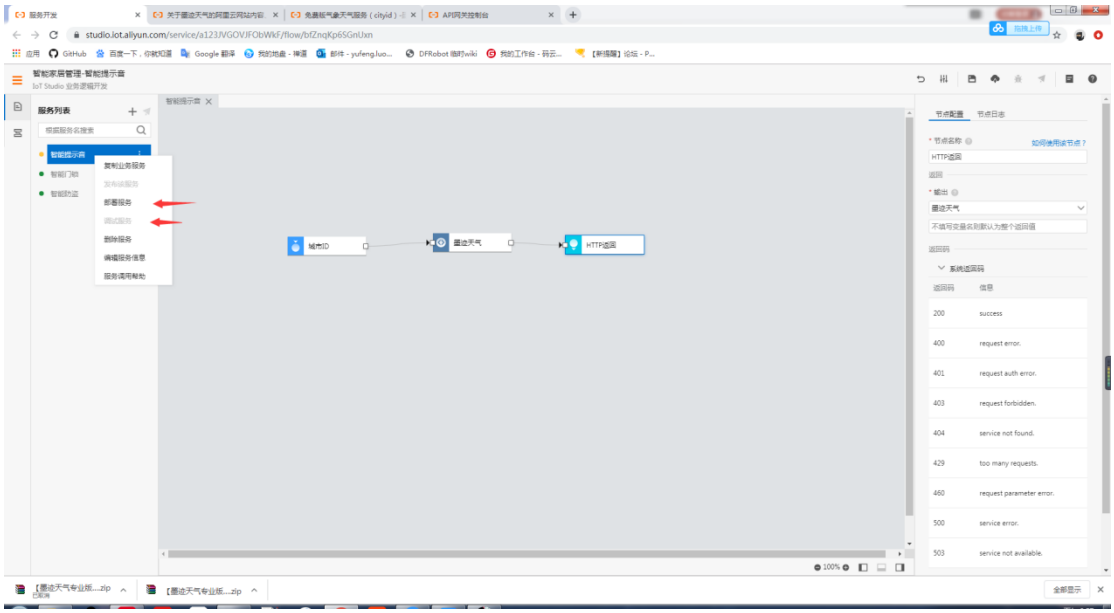


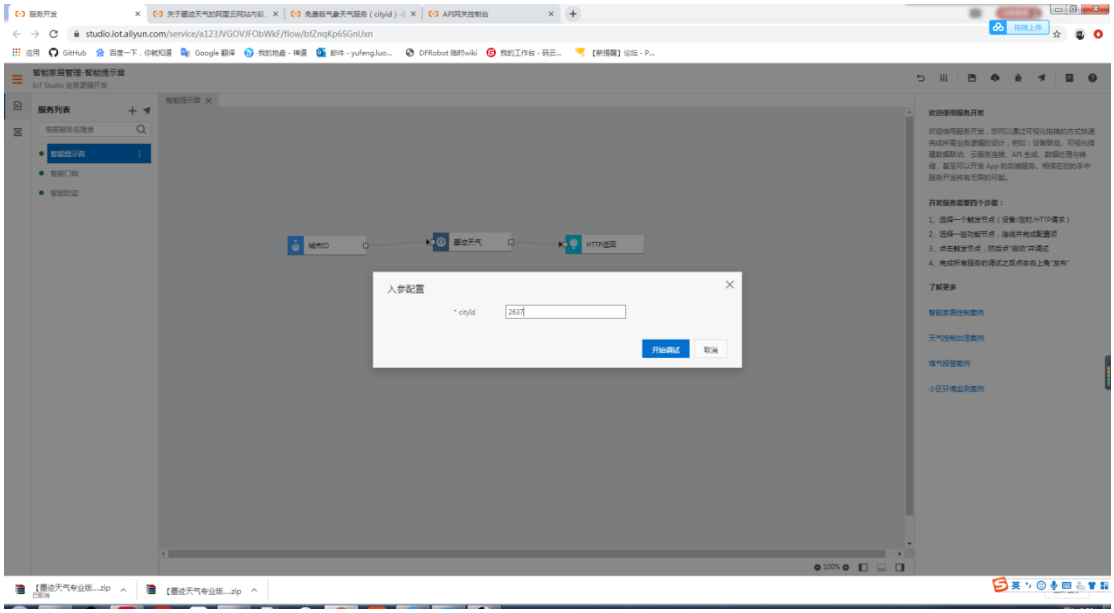
f.回到服务开发中配置“墨迹天气”的编码方式和参数。如下图所示，



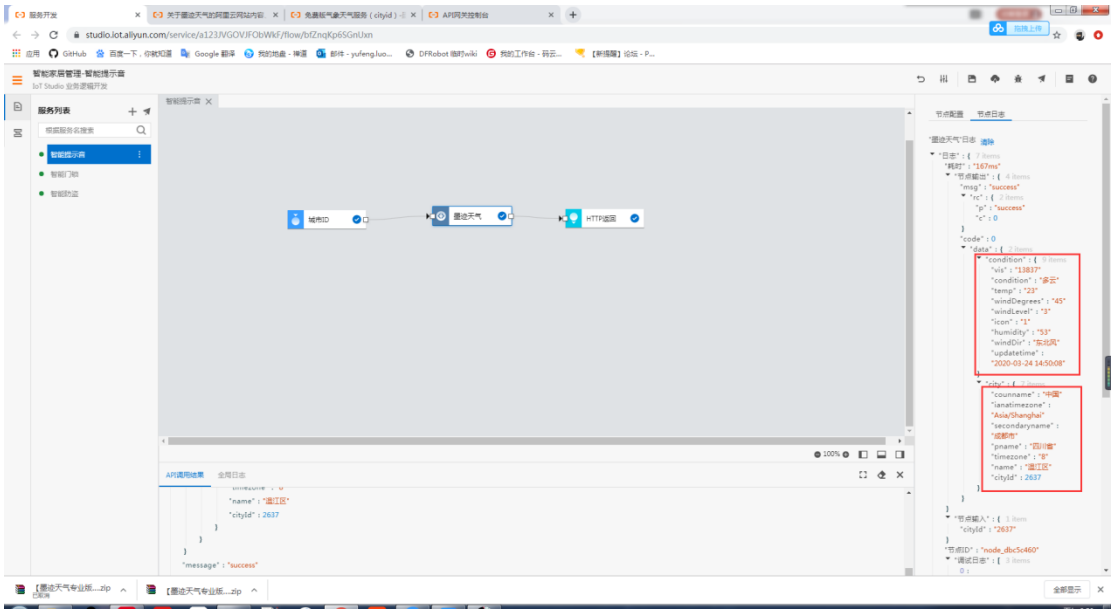
如此即可获取成都温江的天气信息了。由此“墨迹天气”节点则配置完成。

❹获取**墨迹天气**节点的输出信息。右键点击左侧导航栏的**智能提示音**，点击**部署服务->调试服务**,输入城市 ID：2637（温江），点击**开始调试**。

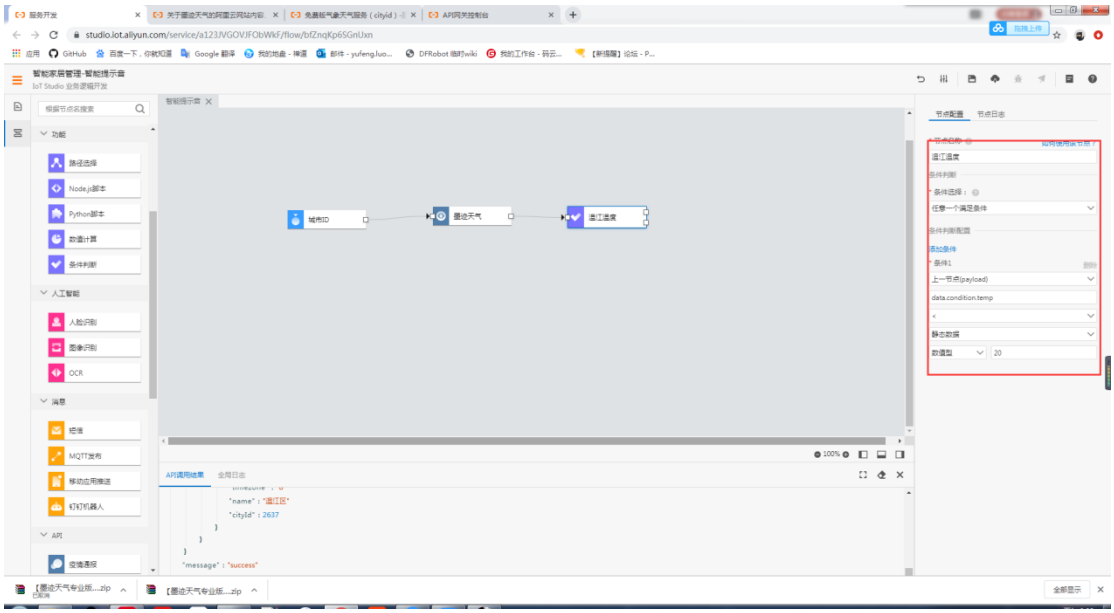




鼠标选中**墨迹天气**节点，点击右侧栏节点日志，即可获取**墨迹天气**节点的输出信息，如下图所示：

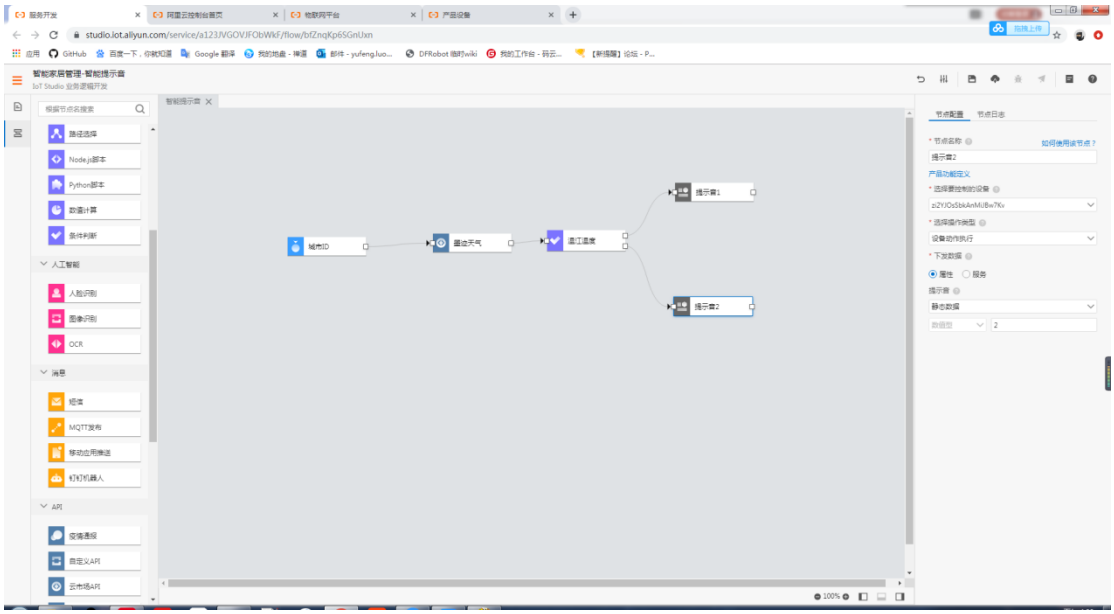


❺添加**条件判断**节点信息。选择功能->条件判断节点，并将其节点名称改为“**温江温度**”，选择**任意一个满足条件**，配置为：**上一个节点+ data.condition.temp**，**<**；**静态数据**，**数值型**，**20**

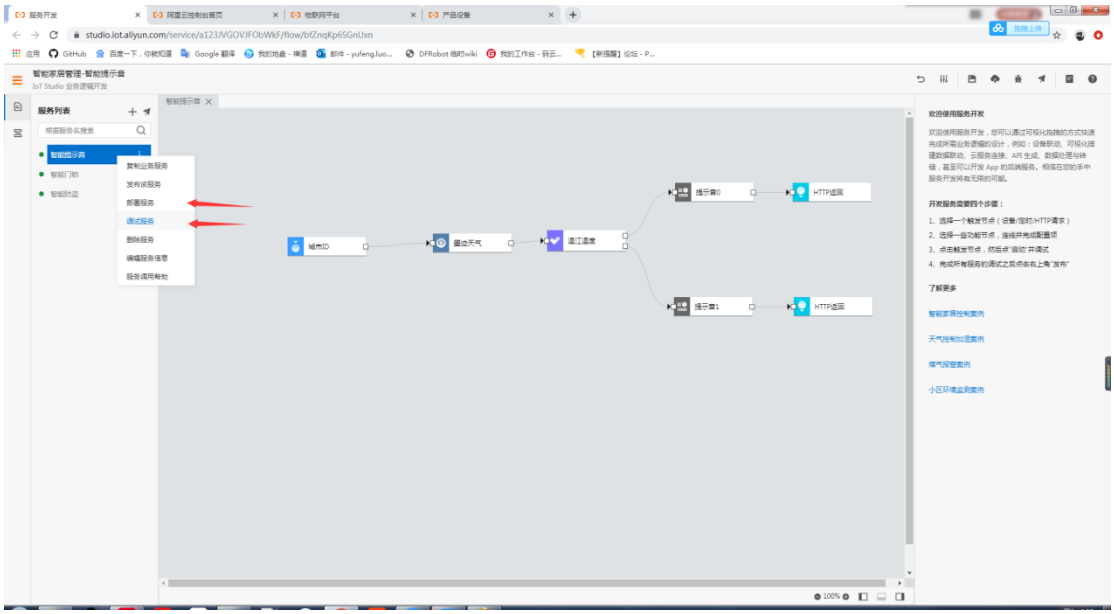


❻添加 2 个**智能提示音**设备。选中设备->智能提示音，一个连接到“**温江温度**”的满足条件接口，并将设备节点名改为“**提示音 0**”，将控制设备配置好，播放提示音为**数值型**，**1**；另一个连接到“**温江温度**”的不满足接口，并将歌曲改为“**提示音 1**”，将控制设备配

置好，播放提示音为**数值型**，**1**，配置如下所示：

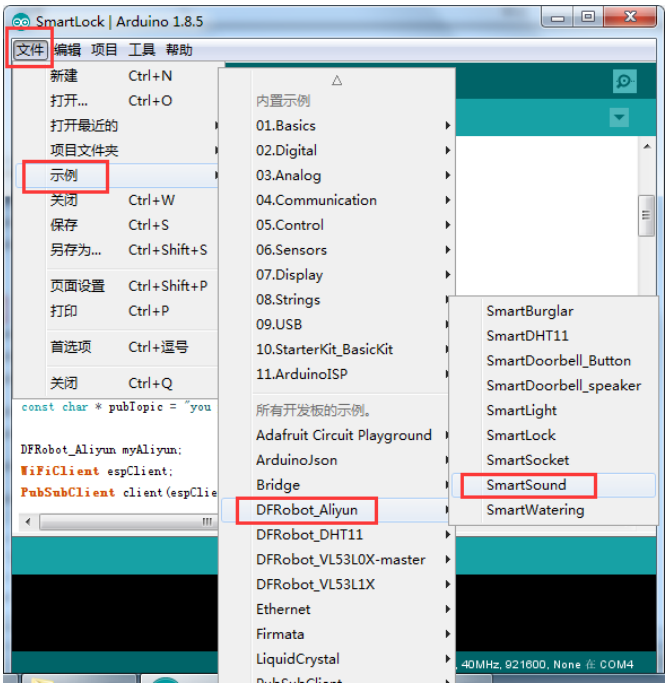


❼添加 **HTTP 返回**节点。配置为上一节点。将城市 ID 输入、墨迹天气、温江温度、提示音 0/1、HTTP 返回连接起来，右键点击**智能提示音，**再点击**部署服务->调试调试服务**，即可播放相应的提示音，如下图所示：



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 摄氏度时播放提示音 0，当检测到大于或等于 20 摄氏度时，播放提示音 1。

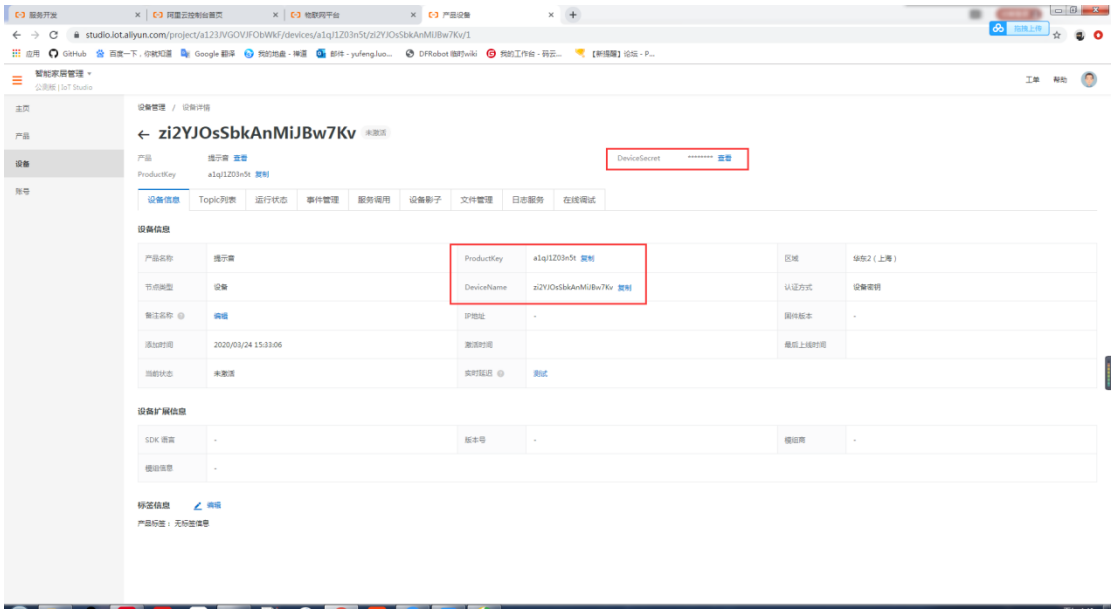
8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartSound** 项目，如下图所示：



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";
```

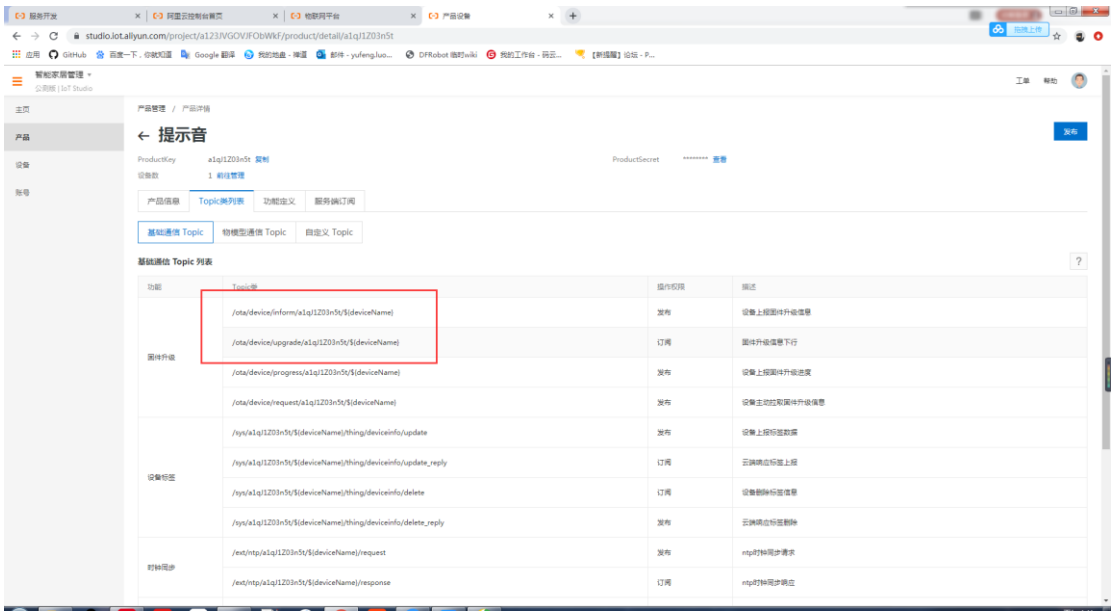
② 填写设备证书信息。获取设备 **SmartSound** 的设备信息，如下图所示：



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。

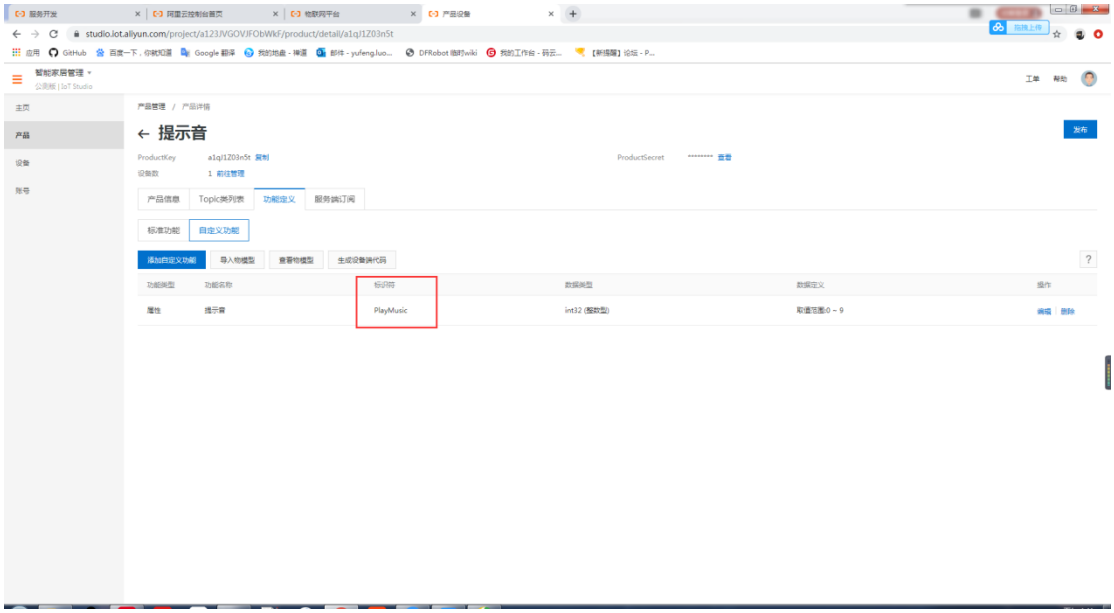
```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

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。




```
/*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。（可能会有一些时间的延迟，属于正常现象）

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

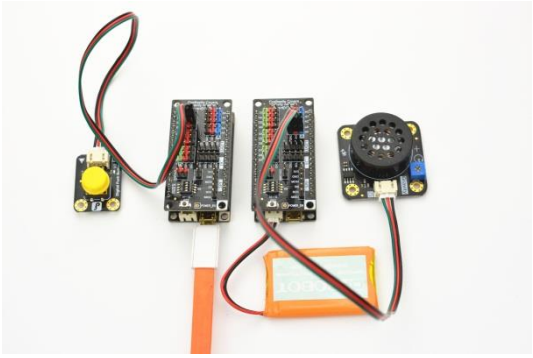
FireBettle Board-ESP32 x2

FireBettle Gravity 扩展板 x2

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。

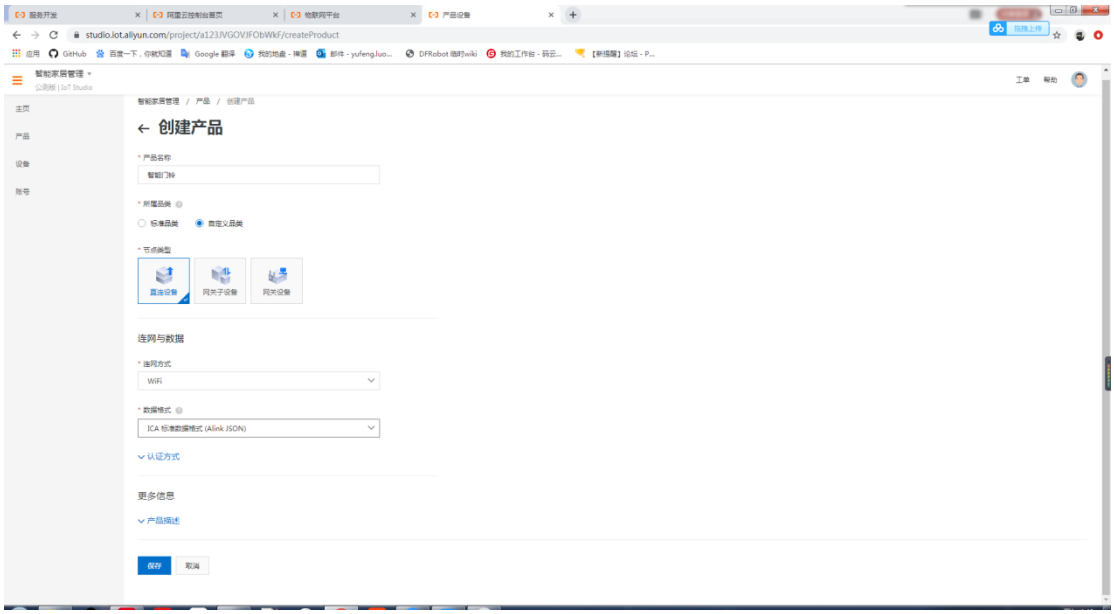


应用分析：通过前面应用的学习，智能门铃项目中有 2 个物理设备，我们需要创建 2 个设备，对于按钮来说，只需要上报属性至阿里云，门铃只需从阿里云下发指令。其次是创建一个钉钉服务来关联按钮和喇叭。

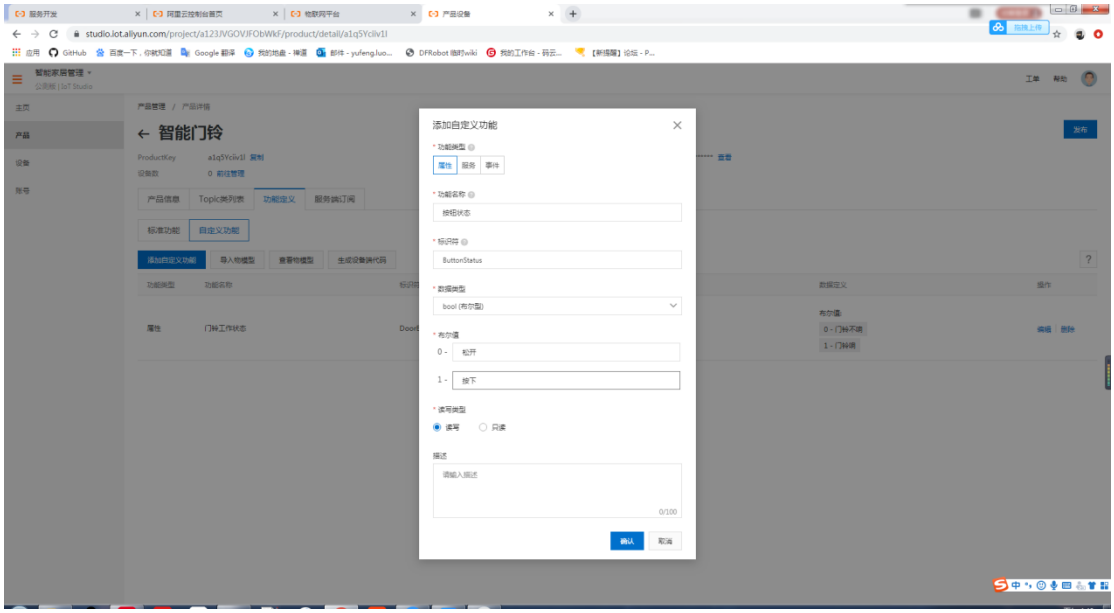
制作步骤

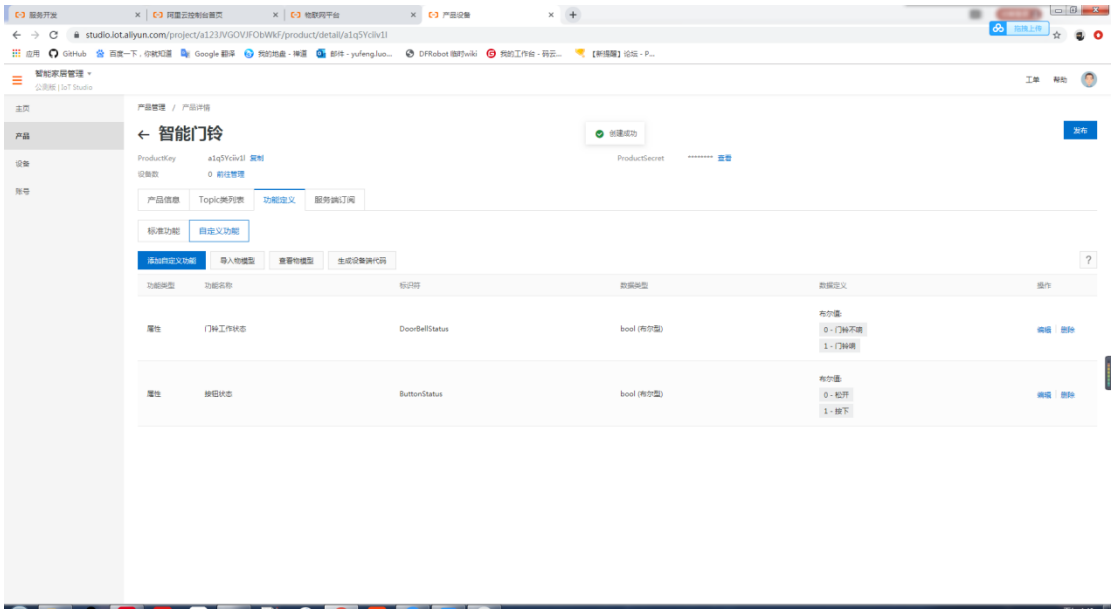
1.打开浏览器，输入网址 https://iot.aliyun.com,登录并进入**物联网平台**；

2.在**智能家居管理**下创建一个名为**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；

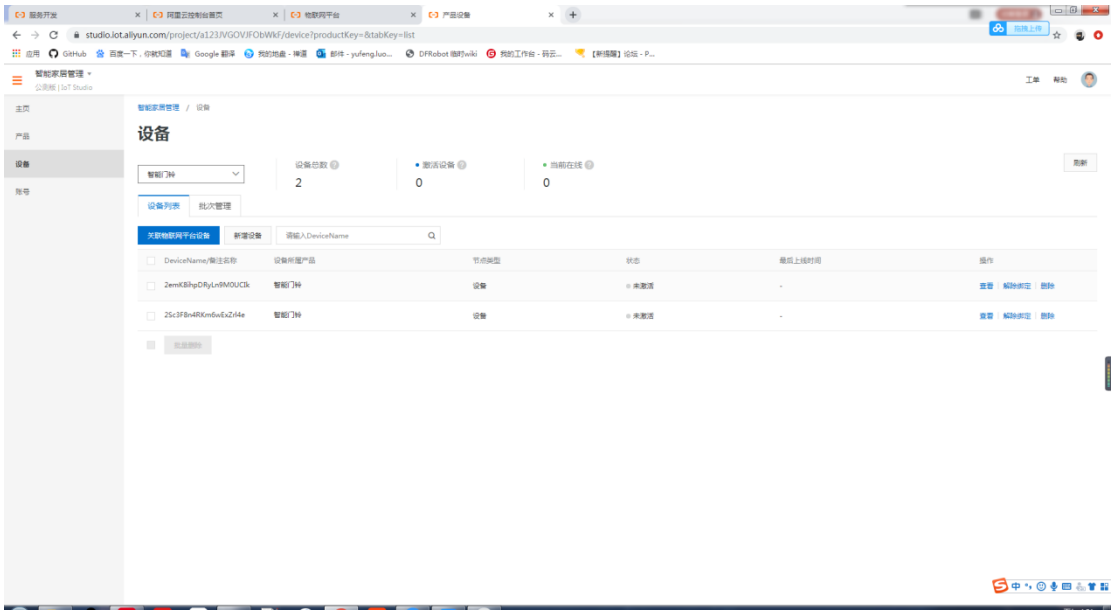


3. 为智能门铃产品定义一个功能名为“**门铃工作状态**”和“**按钮状态**”的属性，自定义产品标识符，配置，如下图所示：

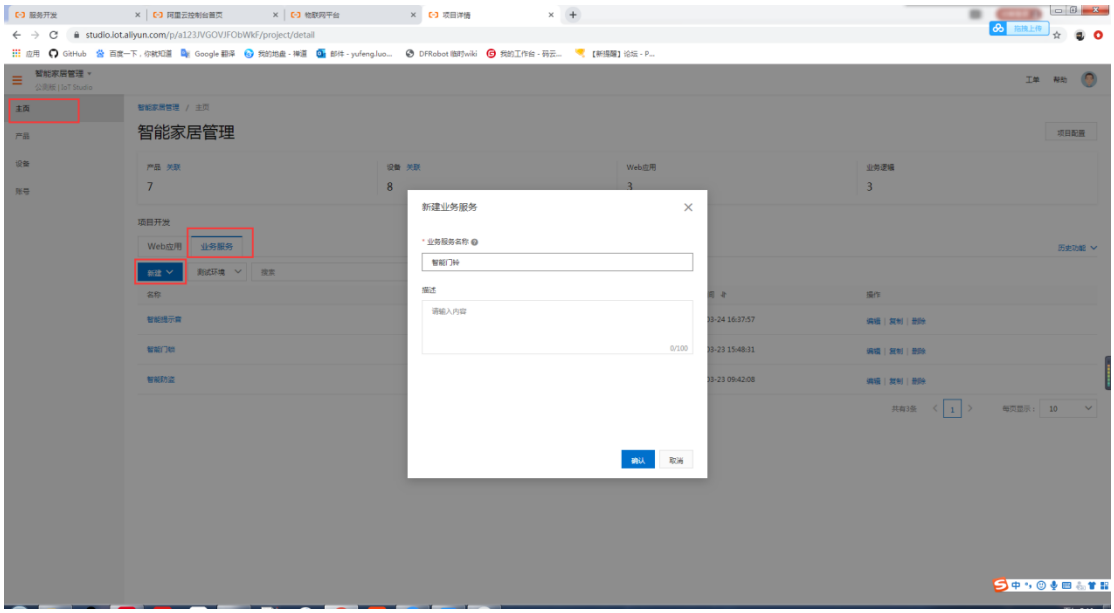




4.为**智能门铃**添加 2 个设备，如下图所示：



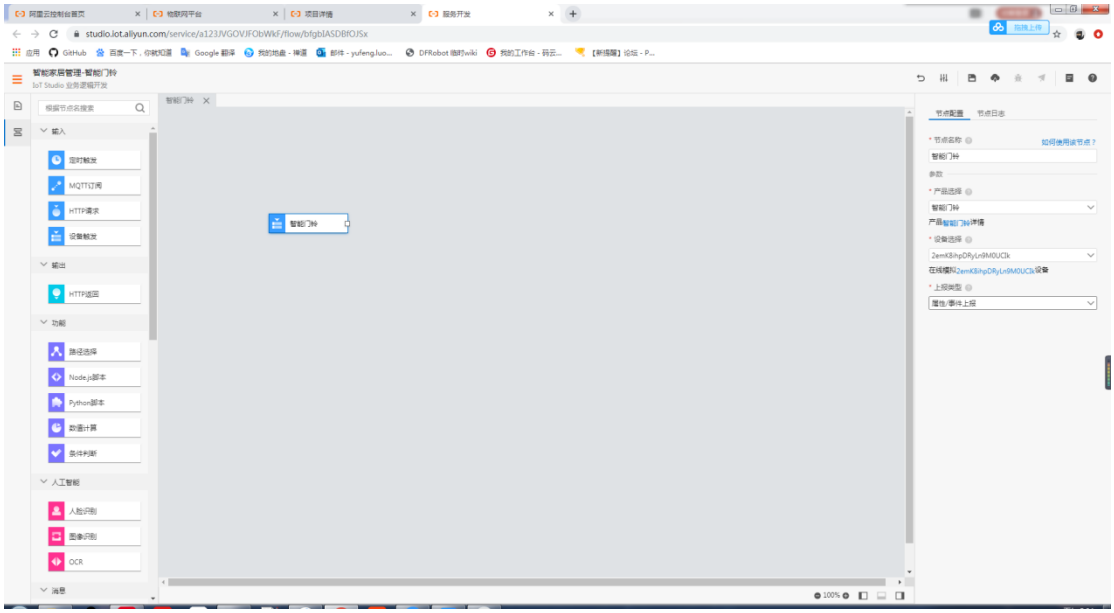
5. 选中左侧导航栏的 **IoT Studio->服务开发**,点击**新建服务**，根据页面提示，创建一个名为“**智能门铃**”的服务，该服务的主要功能为：当阿里云平台检测到按钮被按下，则下发播放门铃指令，让喇叭播放乐曲。如下图所示：



6.点击**完成**，进入**服务编辑器**。具体开发技巧请参考网址:

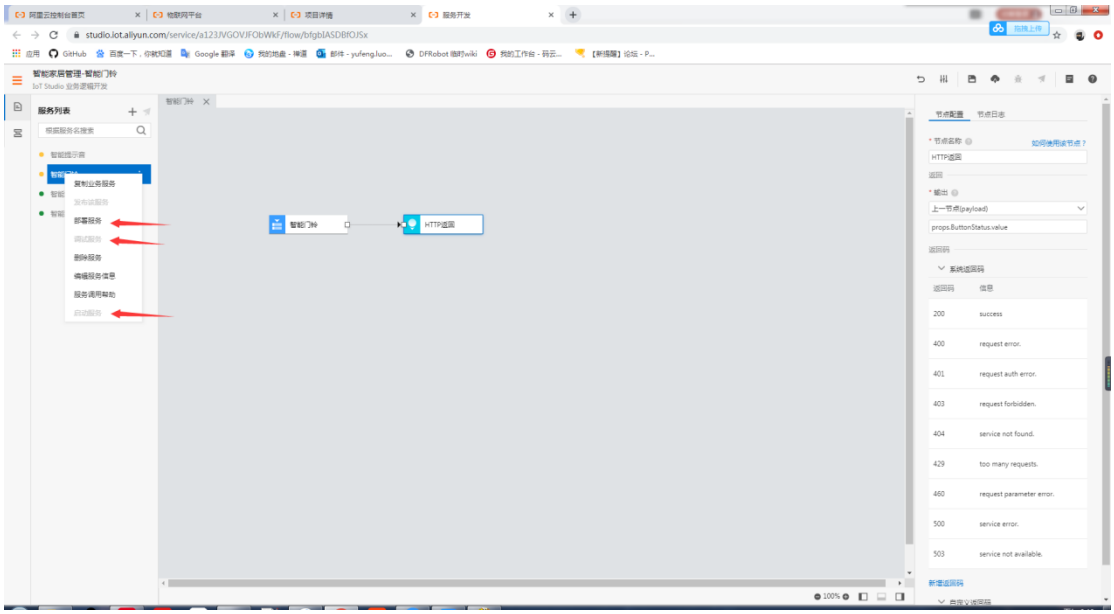
https://linkdevelop.aliyun.com/studioservice-doc#index.html

❶ 点击左侧导航栏的**节点**，在节点列表中选择**设备->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，选择第一个设备，并选择监听“属性或事件上报”上报，如下图所示:

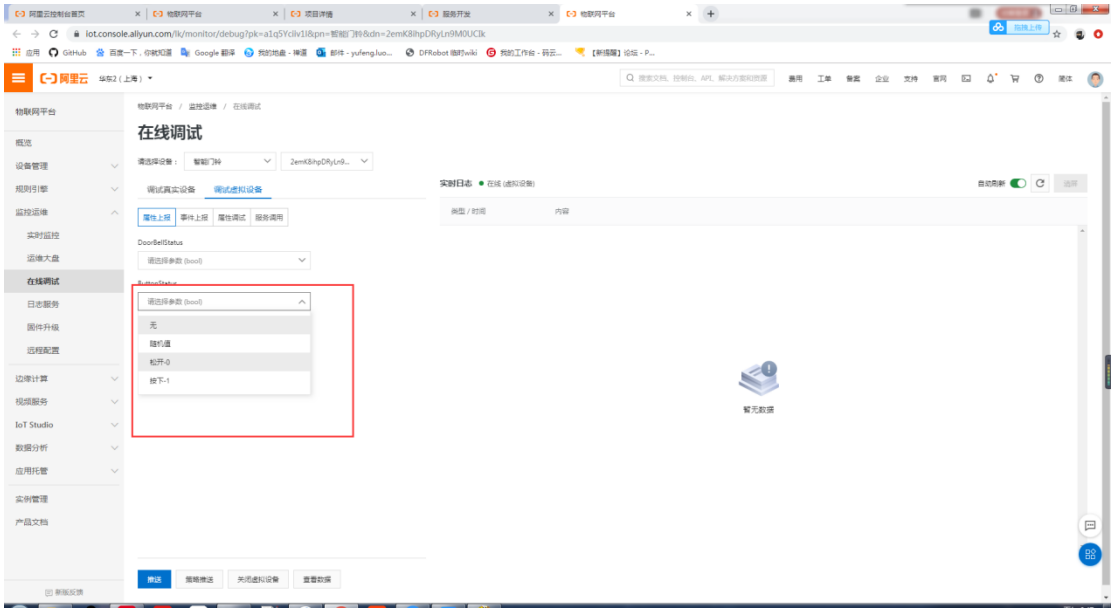


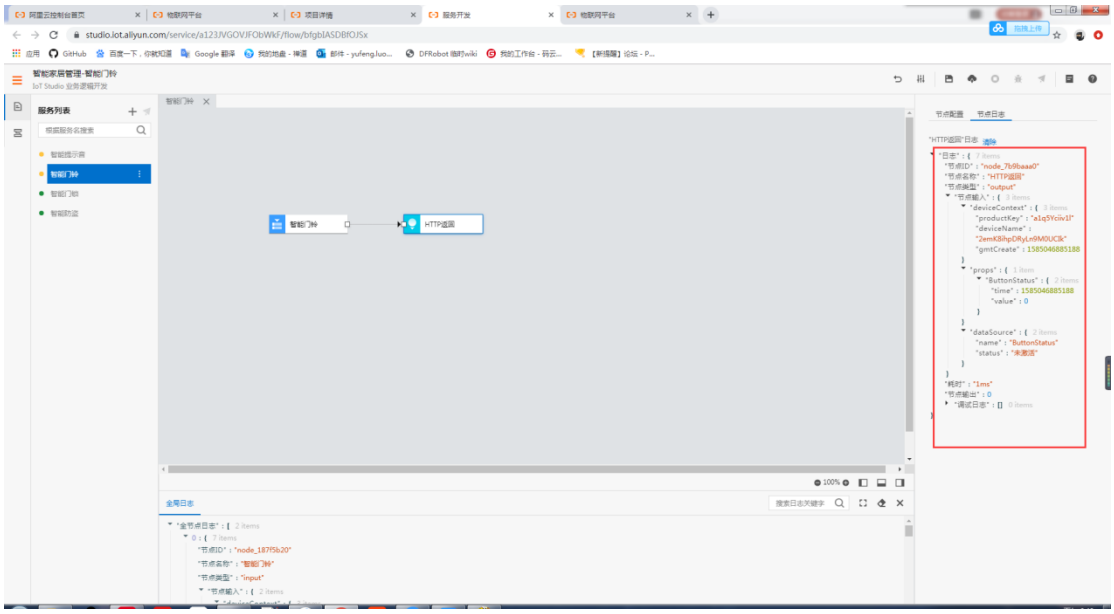
❷获取“按钮状态”节点的节点输出信息。“按钮状态”节点配置完成后，选中左侧节点列表的**功能->HTTP 返回**，输出“**选择上一节点**”+“**props.ButtonStatus.value**”，右键

点击**智能门铃->部署服务->启动服务->调试服务**，如下图所示：

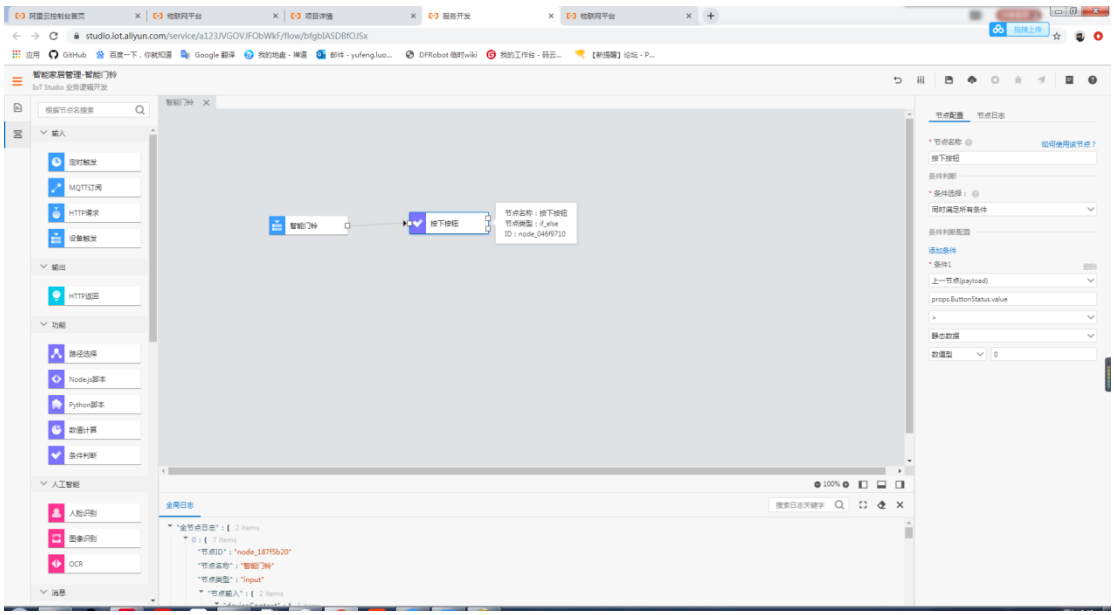


在弹出的“在线调试”页面中选择某个参数，点击**推送**，回到服务开发页面，点击节点日志，如图：



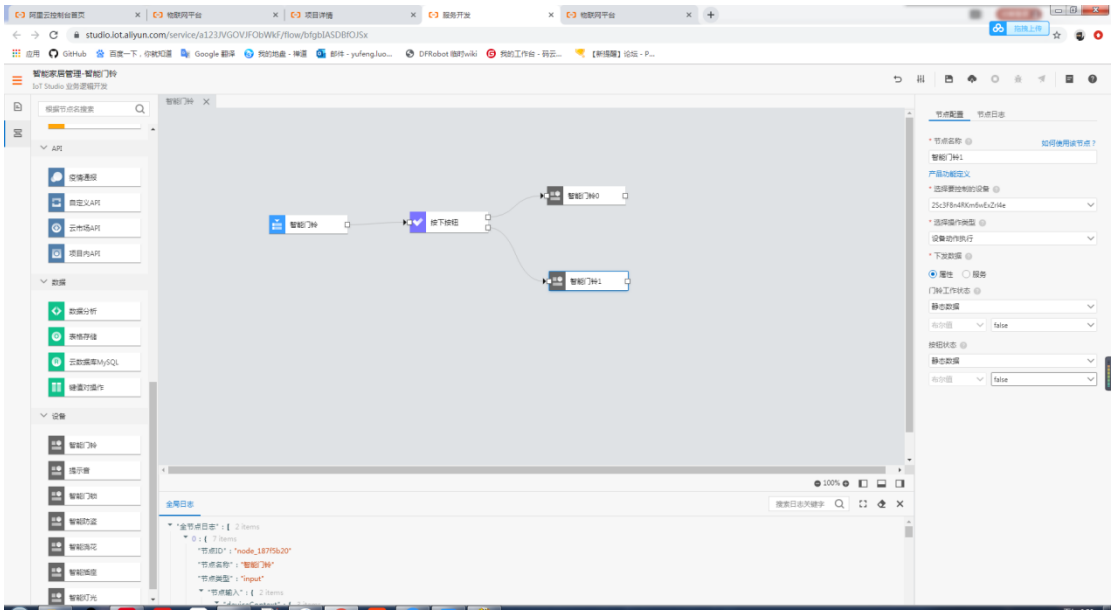


❸添加一个“**条件判断**”节点，条件节点相当于一个 if-else 判断。选中左侧节点列表的功能->条件节点，将其名称改为“**按下按钮**”，将其配置为**同时满足所有条件**，“**上一个节点**”+“**props.ButtonStatus.value**”；**>**，**静态数据**，**数值型**，**0**。可以参考**如何使用该节点?->如何配置条件节点?**



❹将“HTTP 返回”节点删除。添加**设备->智能门铃**，添加两个，一个与满足接口连接，将其进行配置，**智能门铃 0**，**控制设备选择 Speak2**,设备动作执行，门铃工作状态选择**静态数据**，**布尔值**，**true**,按钮状态选择**静态数据**，**布尔值 true**.，另一个与不满足接口连接。将

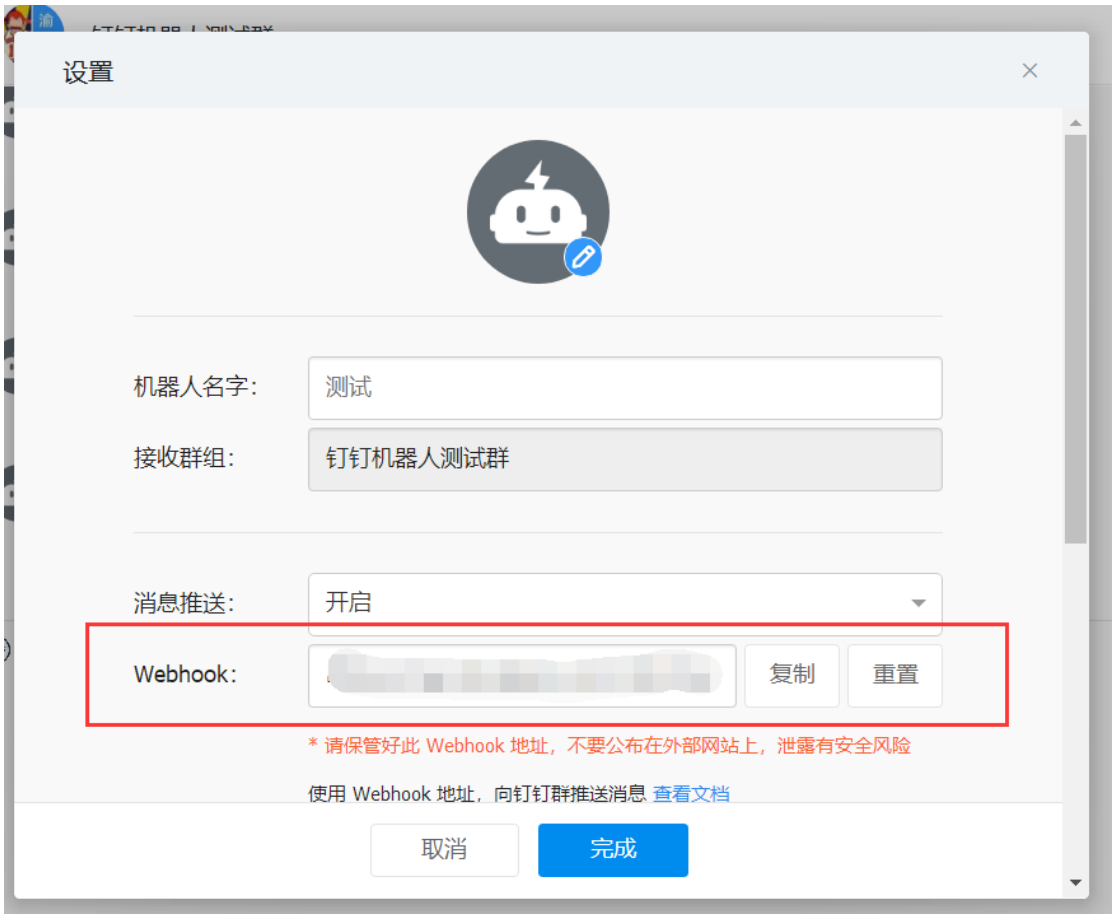
其进行配置，**智能门铃 1，控制设备选择 Speak2**,设备动作执行，门铃工作状态选择**静态数据，布尔值，false**,按钮状态选择**静态数据，布尔值，false**.并将按钮状态、按下按钮、智能门铃连接起来



最后我们还需要将门铃响的信息推送至钉钉。

❺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？**

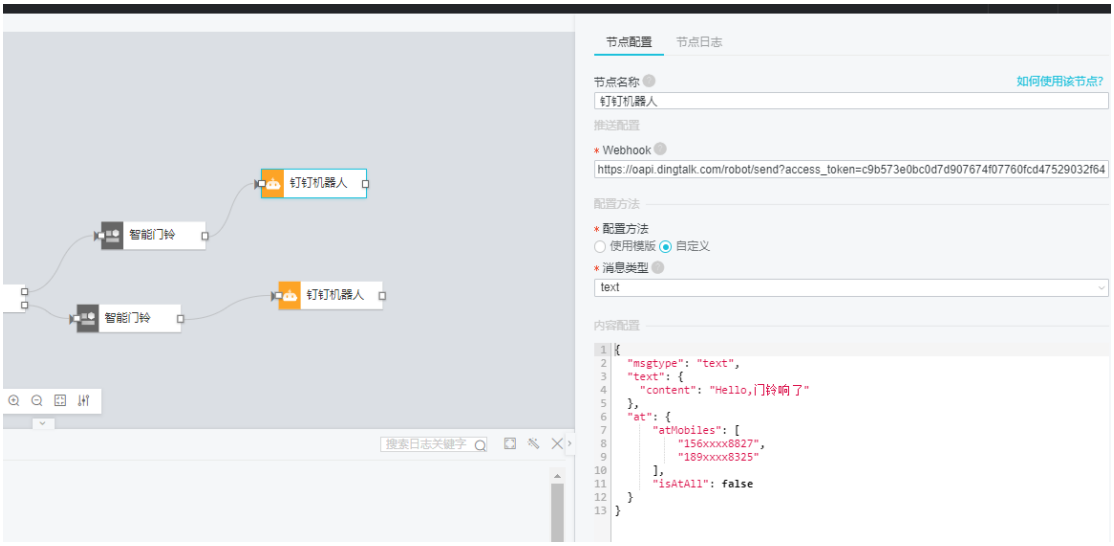
a.获取 **Webhook**。在获取之前，你需要在钉钉群中，添加一个钉钉机器人(注意查看钉钉机器人的 **Webhook** 时只能用电脑端的钉钉)，具体添加方法请参考**如何使用该节点？->如何配置钉钉机器人节点？**最终获取的 **Webhook**，复制粘贴至右侧栏的 **Webhook** 处，如下图所示：

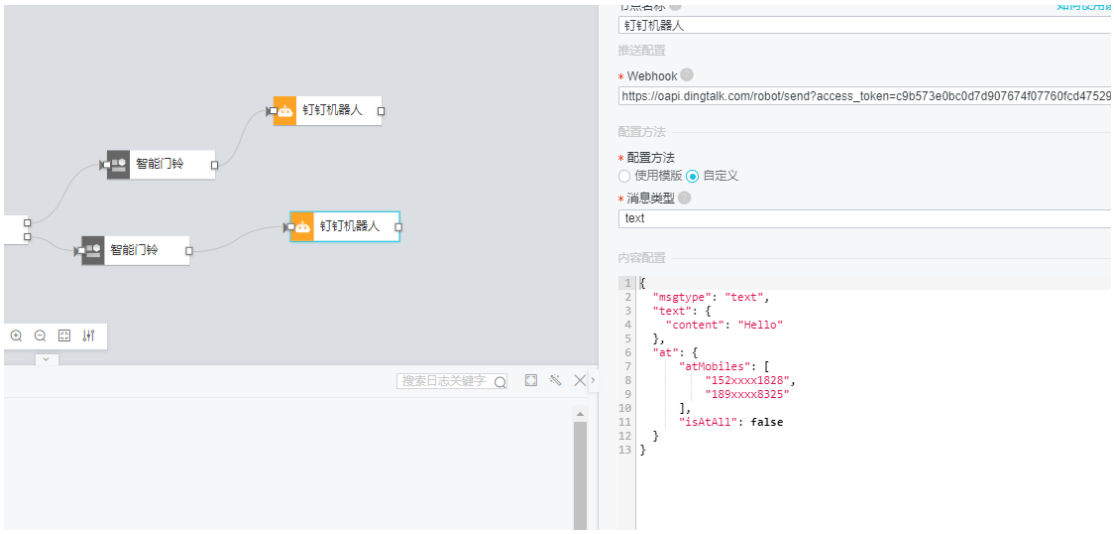

设置
机器人名字:
测试
接收群组:
钉钉机器人测试群
消息推送:
开启
Webhook:
复制
重置
* 请保管好此 Webhook 地址，不要公布在外部网站上，泄露有安全风险
使用 Webhook 地址，向钉钉群推送消息 查看文档
取消
完成

设置安全关键词:

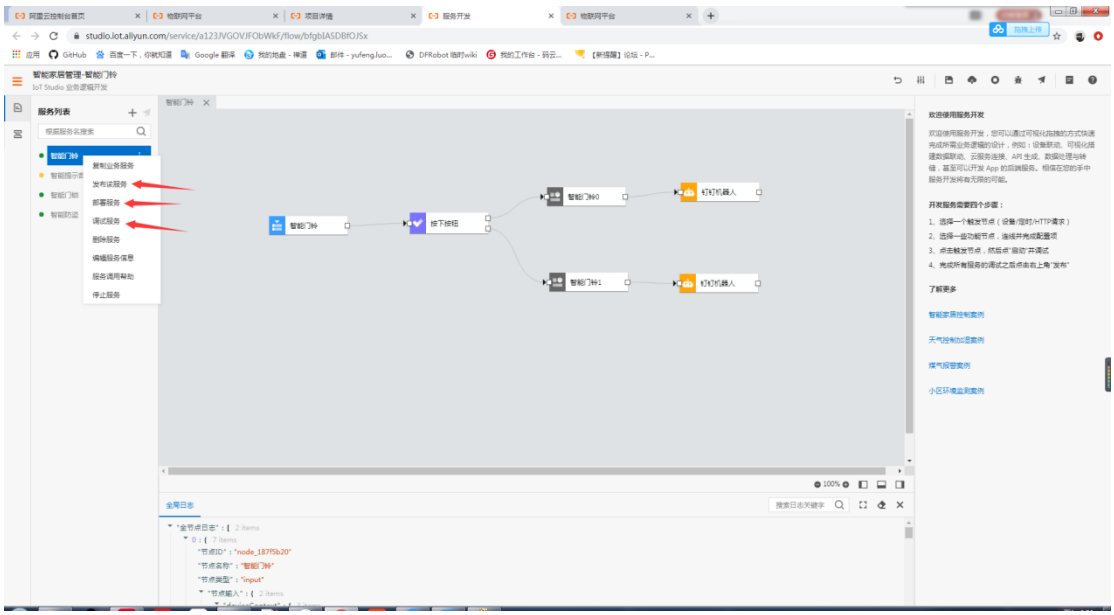
d. 上一个钉钉机器人，选择自定义，“**text**”作为消息类型，并在 **content** 的双引号内填入信息“**hello,门铃响了**”。

下一个钉钉机器人，选择自定义，“**text**”作为消息类型，并在 **content** 的双引号内填入信息“**hello**”。



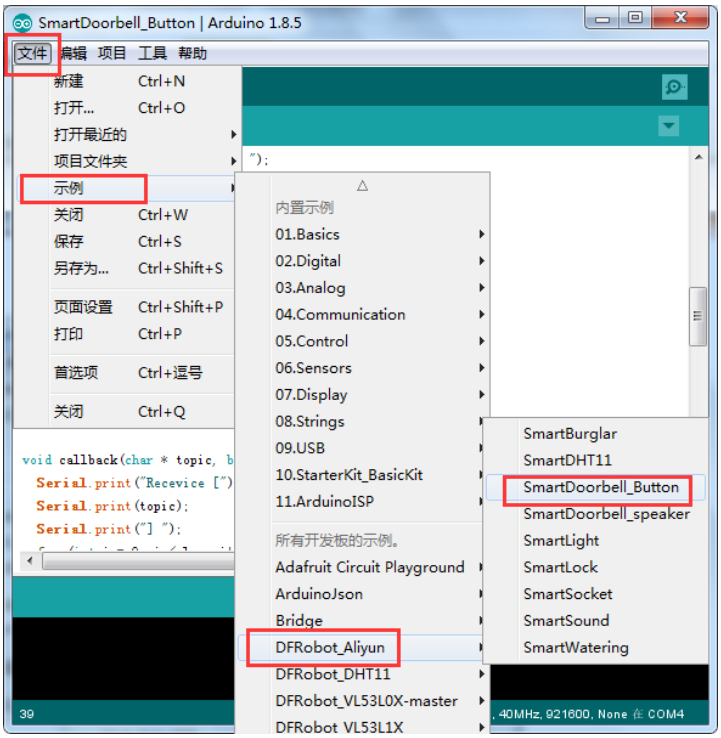


❻点击设备触发节点的**部署**按钮，点击**启动**，如下图所示：



如此，一个名为“智能门铃”的服务就开发完毕了。

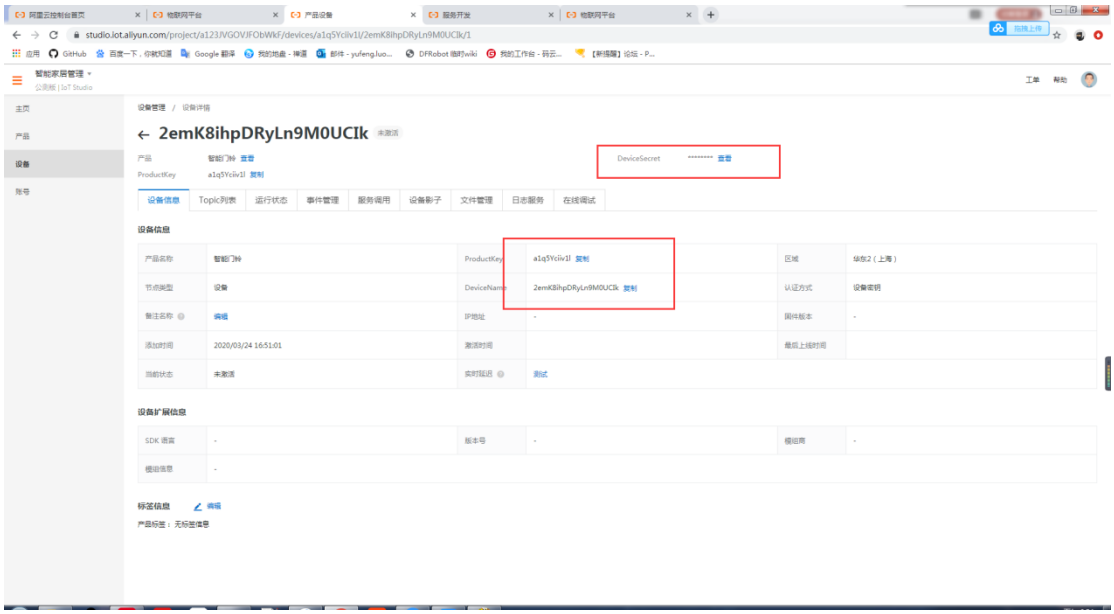
8.Arduino 程序修改。点击**文件->示例->DFRobot_Aliyun->SmartDoorBell_Button**项目，如下图所示：



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";
```

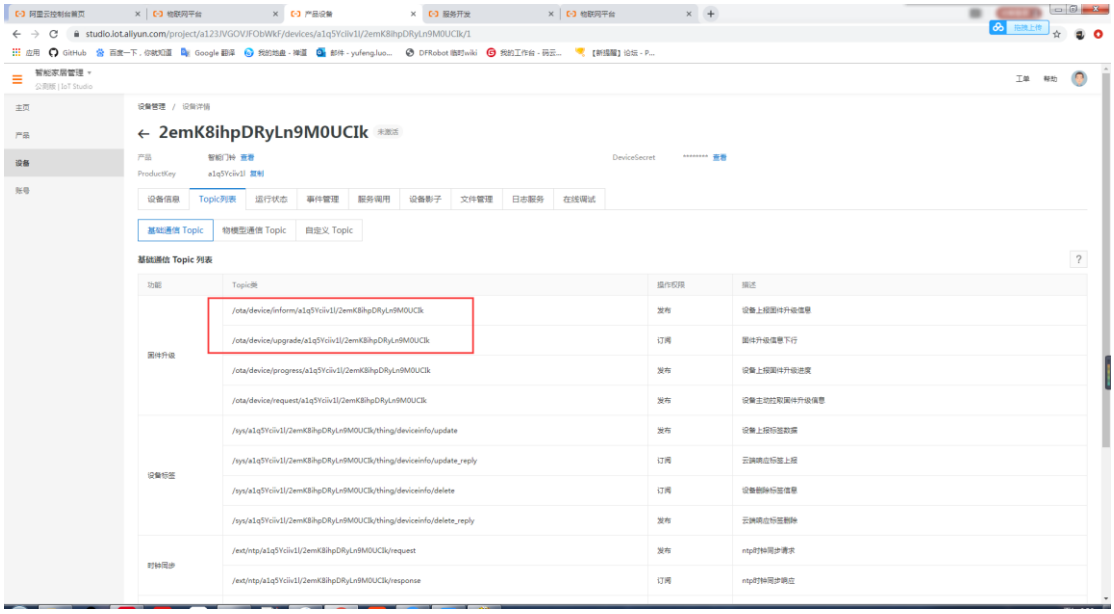
② 填写设备证书信息。获取设备 **Button** 的设备信息，如下图所示:



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
```

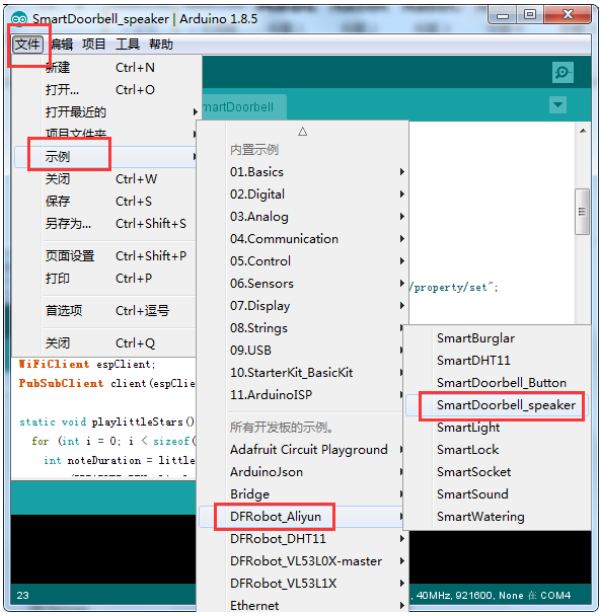
④替换产品标识符。在设备管理的产品项，找到名为智能门铃的产品，点击查看，点击功能定义，即可在自定义功能栏看到标识符，如下图所示，将其替换到程序中的产品标识符即可。

```
/*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。

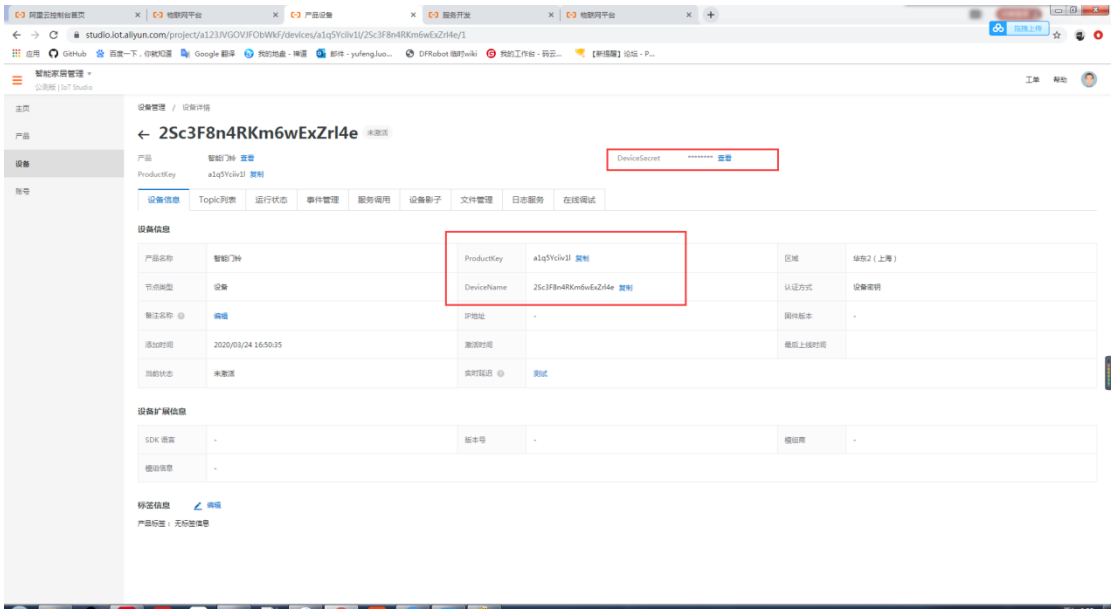
9 Arduino 程序修改，点击**文件->示例->DFRobot_Aliyun->SmartDoorBell_Speaker** 目，如下图所示：



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";
```

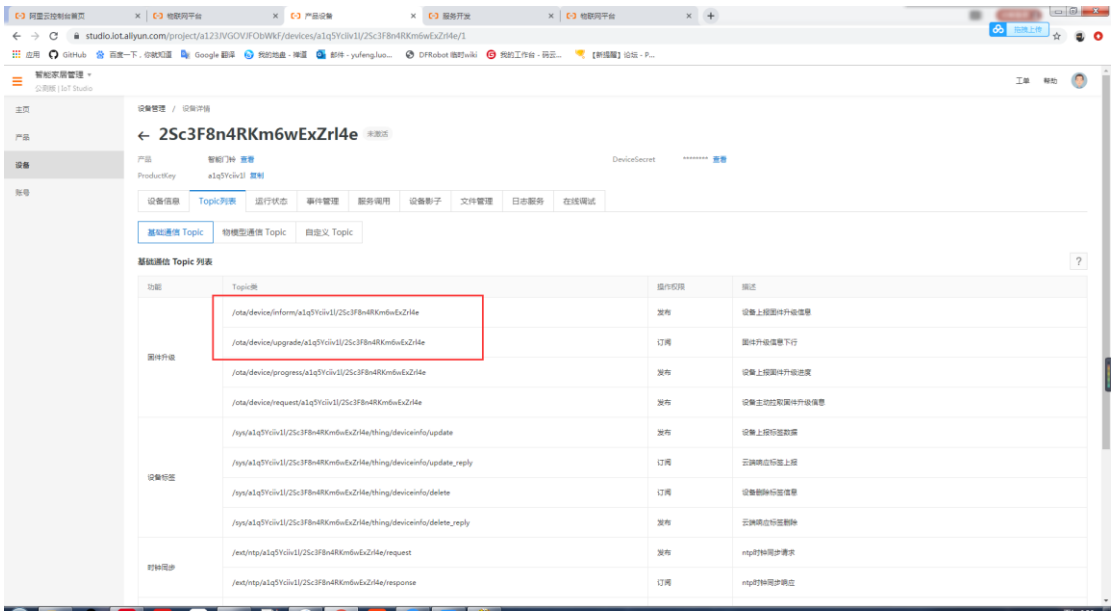
② 填写设备证书信息。获取设备 **Speaker2** 的设备信息，如下图所示：



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。

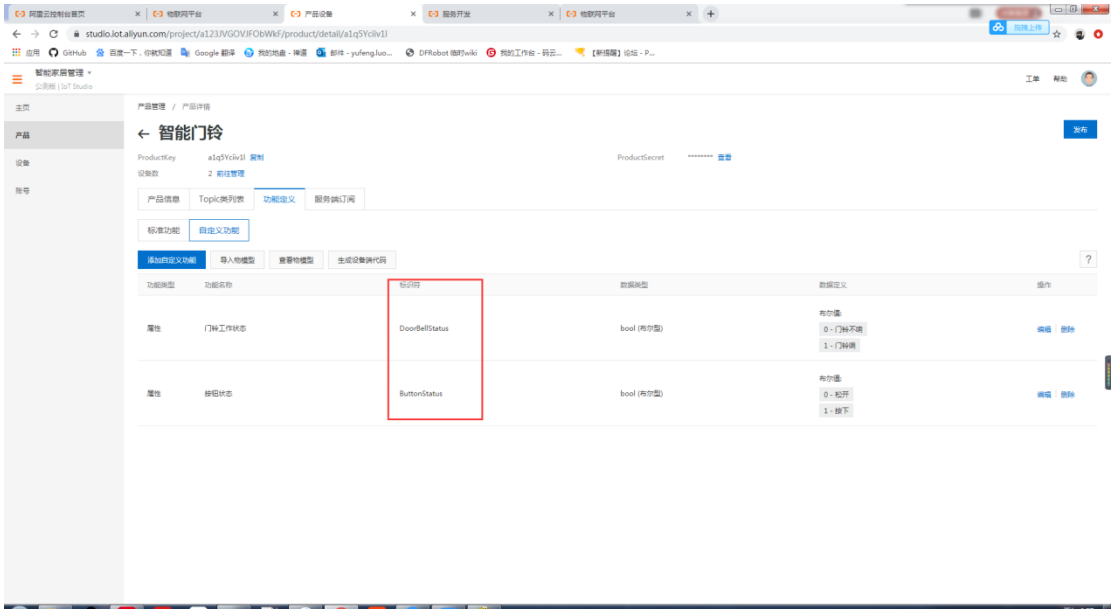
```
/*配置设备证书信息*/
String ProductKey = "you Product Key";
String ClientId = "12345";
String DeviceName = "you Device Name";
String DeviceSecret = "you Device Secret";
```

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：



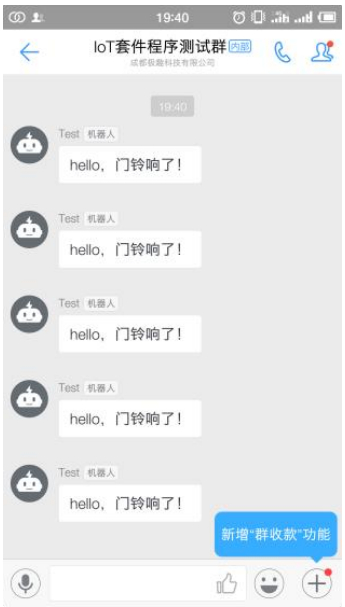
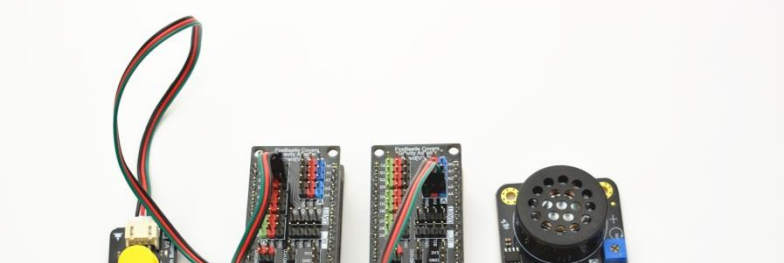
```
/*需要订阅的TOPIC*/
const char * subTopic = "you sub Topic";//****set
```

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：