

阿里云

生活物联网平台 第三方平台介绍

文档版本：

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

目录

1.公版App使用天猫精灵控制设备	04
2.公版App使用Amazon Echo音箱控制设备	07
3.公版App使用Google Home音箱控制设备	11
4.公版App通过IFTTT搭建自动化情景	14
5.自有App接入天猫精灵教程	17
6.自有App定制Amazon Alexa技能	31
7.自有App定制Google Assistant技能	35

1.公版App使用天猫精灵控制设备

使用公版App的产品，只需定义了生活物联网平台的标准功能属性，并且在天猫精灵支持的品类中，即可快速连接天猫精灵，实现天猫精灵音箱对设备的控制。

背景信息

本示例中使用天猫精灵音箱，详细的官方文档请参见[天猫精灵官方推荐品类](#)和[天猫精灵设备案例](#)。

限制条件

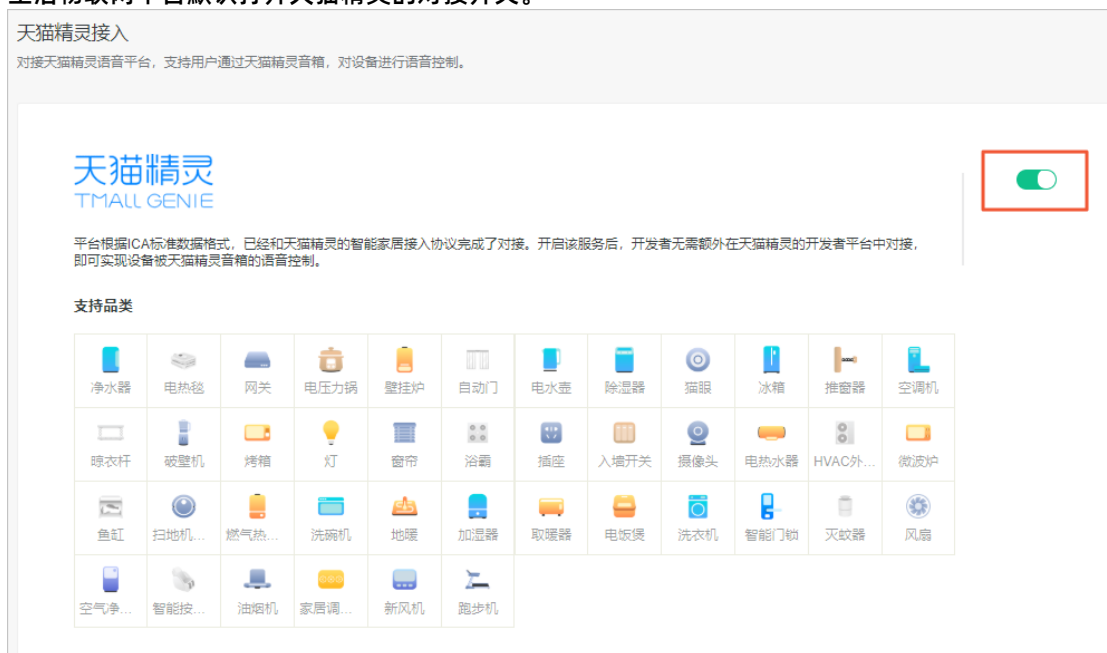
仅支持在中国境内激活的设备对接天猫精灵。

控制台开通

1. 确认已打开天猫精灵的控制开关。

- i. 选择一个现有的项目，或创建一个新项目。参见[创建项目](#)。
- ii. 创建一个新产品，参见[创建产品](#)。
- iii. 定义产品功能，参见[新增标准功能](#)。
- iv. 添加测试设备，参见[新增测试设备](#)。
- v. 配置App功能参数项。详情请参见[配置App功能参数](#)。

生活物联网平台默认打开天猫精灵的对接开关。



2. 单击品类图标，查看功能映射。

生活物联网平台已经默认完成了部分天猫精灵品类功能的映射。这些已经完成映射的指令，您无需再自行开发，只需确保设备使用了映射过的标准属性，即可被天猫精灵控制。

3. 调试设备。

使用公版App配网绑定了您的测试设备后，可以根据下方消费者使用步骤，绑定天猫精灵，即可实现音箱的控制。下载公版App请参见[公版App介绍](#)。

注意 首次注册账号时，国家和地区请选择中国内地。

消费者使用

阿里IoT的公版免开发App支持天猫精灵控制，使用方式如下步骤所示。

1. 用户拥有一台支持天猫精灵的设备，下载阿里云IoT提供的公版App云智能-智能家居助手，绑定该设备。
2. 用户拥有一台天猫精灵音箱，从手机端应用商店下载天猫精灵App并绑定该音箱。

3. 在公版App中，选择我的 > 第三方平台接入（或更多服务）> 天猫精灵 > 绑定账号，在弹出的界面中输入淘宝账号，操作步骤如下图所示。



4. 登录成功后，您可以通过天猫精灵音箱控制您的设备。
在天猫精灵App，即可在我家页签的设备列表中看到您的设备。



2. 公版App使用Amazon Echo音箱控制设备

使用公版App的产品，可以一键开通Amazon Alexa，实现Amazon Echo音箱对设备的控制。

背景信息

本示例中使用Amazon Alexa，详细的官方文档请参见[Amazon Alexa官方文档](#)。

限制条件

仅支持在以下地区激活的设备对接Amazon Alexa：

- 海外
- 中国香港
- 中国澳门
- 中国台湾

控制台开通

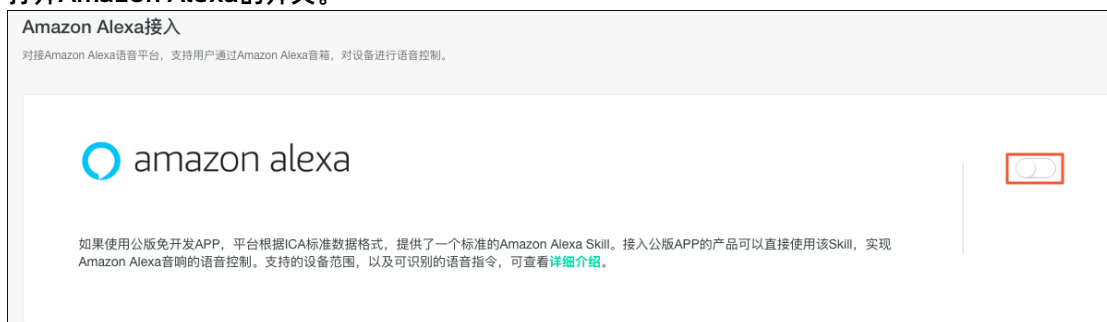
1. 确保自己的产品在Amazon Alexa可控范围内。

生活物联网平台已支持Amazon Alexa品类	Amazon Alexa官方推荐的品类
- 扫地机器人 - 灯 - 插座 - 香薰机 - 窗帘	- 灯和开关: Lights, switches and bulbs - 门锁: Door locks - 智能摄像头: Smart home cameras - 恒温调温器、风扇: Thermostats and fans - 微波炉: Microwave ovens

 **说明** 平台正在逐渐增加Amazon Alexa支持的品类，如果您的产品不在以上列表中，请在对接前[提交工单](#)，与我们的技术支持确认您的产品是否已支持。

2. 购买并开通Amazon Alexa服务。

- 进入服务中心，单击Amazon Alexa对应的开通服务。
- 分配一个项目，并在该项目中创建一个新产品。请参见[创建产品](#)。
- 定义产品功能。请参见[新增标准功能](#)。
- 在人机交互页面，打开使用公版App控制产品的开关，并单击Amazon Alexa对应的设置。打开Amazon Alexa的开关。



人机交互页面其余内容详情请参见[配置App](#)。

3. 设备调试。

- 烧录测试设备的证书信息。请参见[开发设备](#)。
- 下载公版App调试设备（请参见[公版App介绍](#)）。配网绑定了您的测试设备后，可以根据下方消费者使用步骤，绑定Amazon Alexa，即可实现音箱的控制。

 **注意** 首次注册账号时，国家和地区请选择中国内地以外地区，例如选择美国。

消费者使用

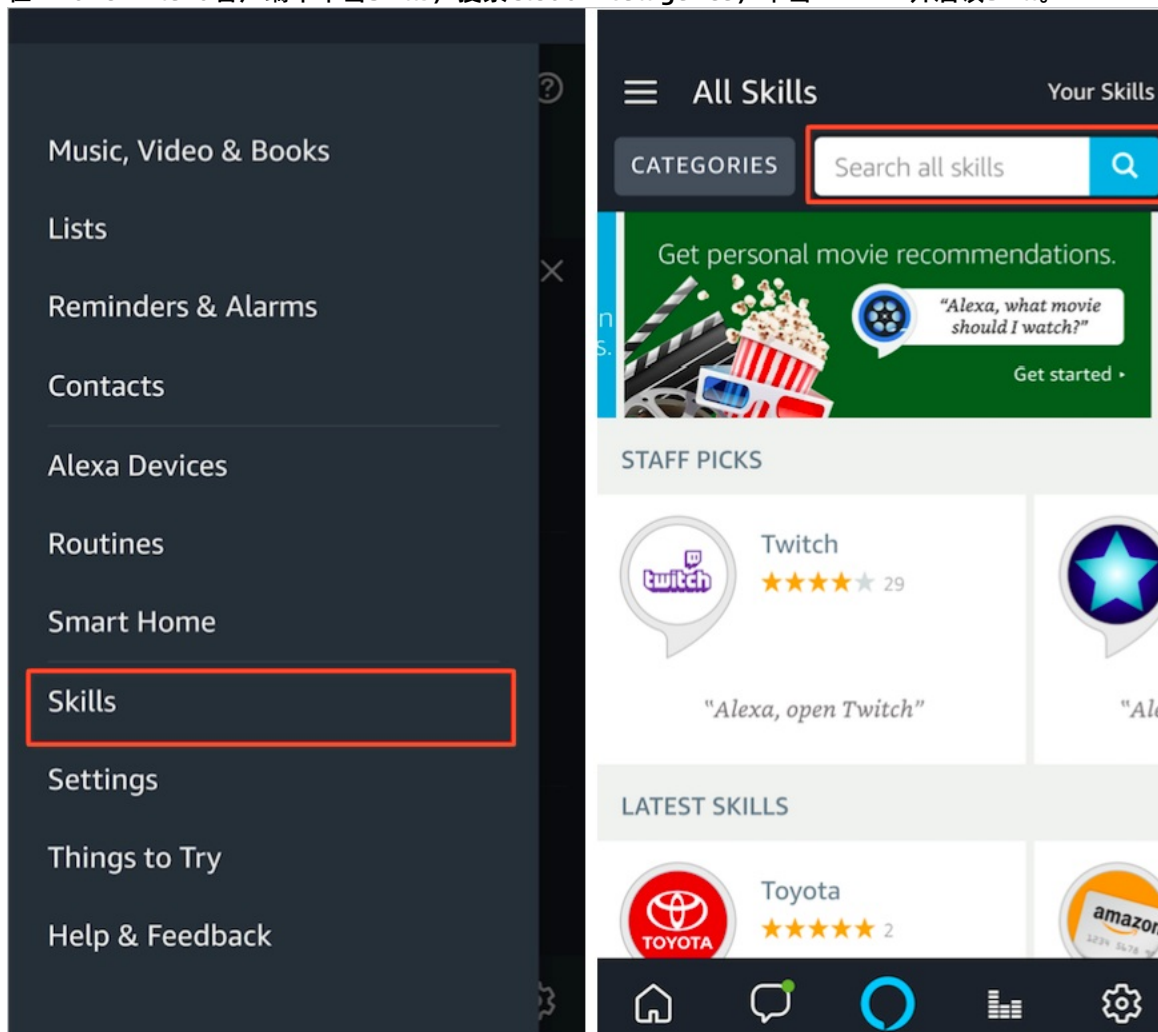
- 用户拥有一台接入生活物联网平台（国际站）的设备，下载阿里云IoT提供的公版App云智能-智能家居助手（中国内地手机端应用商店）或Cloud Intelligence（国外手机端应用商店），绑定该设备。
- 将设备昵称改为英文字母，例如My Light Device。

3. 用户拥有一台Amazon Echo音箱，从手机端应用商店下载Amazon Alexa App并绑定该音箱。

说明

- iOS系统用户请至美国区App Store搜索Amazon Alexa下载，Android用户请至Google Play搜索Amazon Alexa下载。
- 在中国内地地区下载Amazon Alexa需要使用国外的应用商店账号，所以国内测试建议通过网页来配置。

4. 在Amazon Alexa客户端中单击Skills，搜索Cloud Intelligence，单击ENABLE开启该Skill。



5. 输入您Cloud Intelligence客户端的登录账号和密码，并正确选择您的账号所属国家，然后单击Link Now绑定您的智能家居App设备。

6. 控制设备前，Amazon Echo音箱需要先发现设备。您可以对Amazon Echo音箱说：Alexa, discover devices。您也可以在Cloud Intelligence页面单击DISCOVER来发现设备，绑定过的设备会显示在列表中。

说明 如果在公版App Cloud Intelligence中修改了设备昵称，Amazon Echo音箱必须重新发现设备，否则不能对设备进行控制。

7. 您可以通过下列指令控制您的设备（假设设备昵称为My Light Device）。

- 打开设备： Alexa, turn on/off my light device.
- 调节灯亮度： Alexa, brighten/dim my light device.
- 调节灯颜色： Alexa, set my light device to yellow.
- 调节灯色温： Alexa, set my light device to warm white.

3.公版App使用Google Home音箱控制设备

使用公版App的产品，可以一键开通Google Assistant，实现Google Home音箱对设备的控制。

背景信息

本示例中使用Google Assistant，详细的官方文档请参见[Google Assistant官方文档](#)。

限制条件

仅支持在以下地区激活的设备对接Google Home：

- 海外
- 中国香港
- 中国澳门
- 中国台湾

控制台开通

1. 确保自己的产品在Google Assistant可控范围内。

生活物联网平台已支持Google Assistant品类	Google Assistant官方推荐的品类
扫地机器人、灯、插座、香薰机	灯、香薰机、插座、空调、空气净化器、咖啡机、洗碗机、烘干机、风扇、烧水壶、烤箱、冰箱、加湿器、开关、扫地机

 **说明** 平台正在逐渐增加Google Assistant支持的品类，如果您的产品不在以上列表中，请在对接前[提交工单](#)，与我们的技术支持确认您的产品是否已支持。

2. 购买并开通Google Assistant服务。

- 进入服务中心，单击Google Assistant对应的开通服务。
- 分配一个项目，并在该项目中创建一个新产品。请参见[创建产品](#)。
- 定义产品功能。请参见[新增标准功能](#)。
- 在人机交互页面，打开使用公版App控制产品的开关，并单击Google Assistant对应的设置。打开Google Assistant的开关。



人机交互页面其余内容详情请参见[配置App](#)。

3. 设备调试。

- 烧录测试设备的证书信息。请参见[开发设备](#)。
- 下载公版App调试设备（请参见[公版App介绍](#)）。配网绑定了您的测试设备后，可以根据下方消费者使用步骤，绑定Google Assistant，即可实现设备的控制。

 **注意** 首次注册账号时，国家和地区请选择中国内地以外地区，例如选择美国。

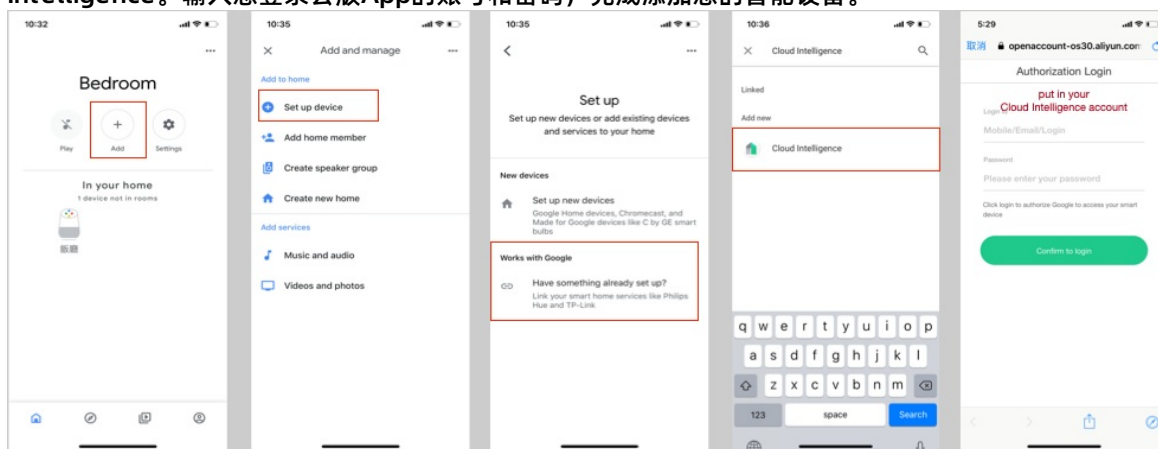
消费者使用

阿里云IoT的公版免开发APP支持Google Home和Google Assistant的控制，使用方式如下步骤所示。

1. 用户拥有一台接入生活物联网平台（国际站）的设备，下载阿里云IoT提供的公版App云智能（中国内地手机端应用商店）或Cloud Intelligence（国外手机端应用商店），绑定该设备。
2. 用户拥有一台Google Home音箱，从手机端应用商店下载Google Home或者 Google AssistantApp并绑定该音箱。

❓ 说明 iOS系统用户请至美国区App Store搜索Google Home下载，Android用户请至Google Play搜索Google Home下载。

3. 在Google Home客户端中单击“+”，Set up devices中选择添加已有设备，搜索 Cloud Intelligence。输入您登录公版App的账号和密码，完成添加您的智能设备。



4. 您可以通过下列指令控制您的设备（如Light）。
 - 打开或关闭灯： Hey Google, turn on/off the light.
 - 调节灯强度： Hey Google, brighten/dim the light.
 - 调节灯颜色： Hey Google, set the light to yellow.
 - 调节灯色温： Hey Google, set the light to warm white.

4. 公版App通过IFTTT搭建自动化情景

使用公版App的产品，可以一键开通IFTTT，可以实现利用IFTTT三方App，搭建各种有趣的自动化场景，联动设备。

背景信息

 **注意** 由于平台国际站整体方案升级，IFTTT暂时无法使用，本文方案仅供调试使用。如果您近期需量产或向中国内地以外地区（包括港澳台地区）交付设备，请通过控制台右上角的工单联系我们，并请密切关注文档更新。

限制条件

仅支持在以下地区激活的设备对接IFTTT：

- 海外
- 中国香港
- 中国澳门
- 中国台湾

控制台开通

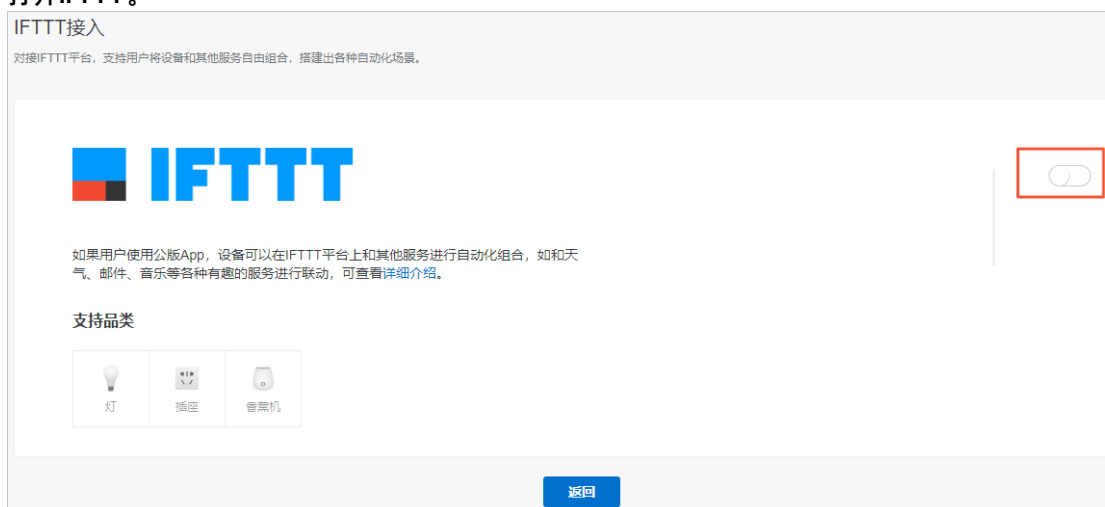
1. 确保自己的产品在IFTTT可控范围内。

生活物联网平台已支持部分IFTTT品类有：插座、香薰机、灯。

 **说明** 若不确定您的品类是否能够被IFTTT控制，您可以[提交工单](#)，联系我们的技术支持确认。

2. 购买并开通IFTTT服务。

- i. 进入服务中心，单击IFTTT对应的开通服务。
- ii. 分配一个项目，并在该项目中创建一个新产品。请参见[创建产品](#)。
- iii. 定义产品功能。请参见[新增标准功能](#)。
- iv. 在人机交互页面，打开使用公版App控制产品的开关，并单击IFTTT对应的设置。打开IFTTT。



3. 设备调试。

- i. 烧录测试设备的证书信息。请参见[开发设备](#)。
- ii. 下载公版App调试设备（请参见[公版App介绍](#)）。配网绑定了您的测试设备后，可以根据下方消费者使用步骤，绑定IFTTT，即可实现设备的控制。

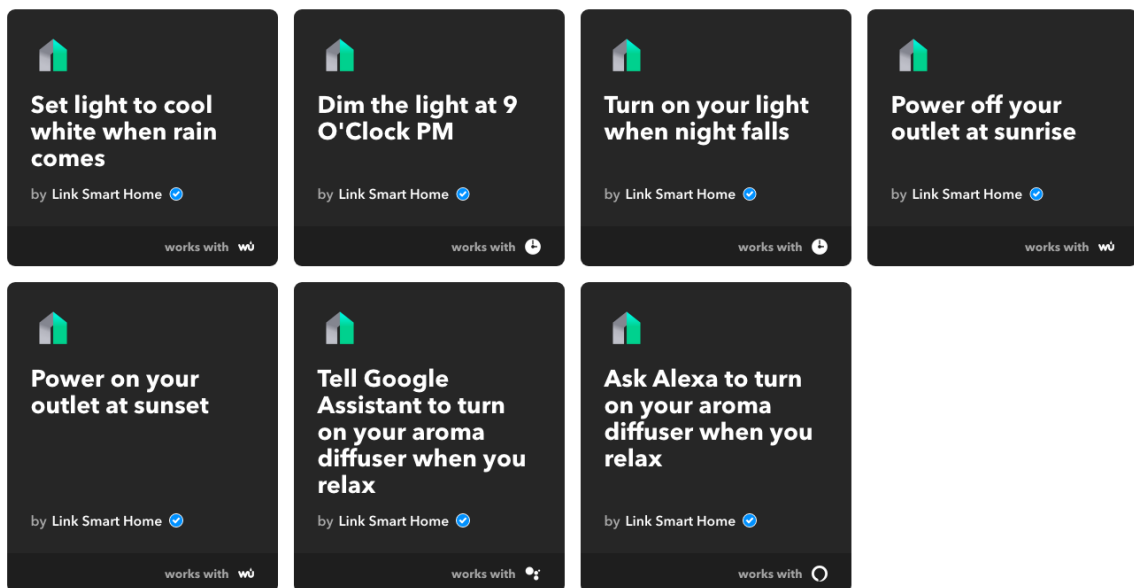
 **注意** 首次注册账号时，国家和地区请选择中国内地以外地区，例如选择美国。

消费者使用

1. 用户拥有一台接入生活物联网平台（国际站）的设备，下载阿里云IoT提供的公版App云智能-智能家居助手（中国内地手机端应用商店）或Cloud Intelligence（国外手机端应用商店），绑定该设备。
2. 用户从手机端应用商店下载IFTTT App并绑定该音箱。

❓ 说明 iOS系统用户请至美国区App Store搜索IFTTT下载，Android用户请至Google Play搜索IFTTT下载。

3. 在IFTTT客户端中搜索Cloud Intelligence，单击Connect授权公版App。
4. 完成授权后，可以自由配置您的Applets（自动化场景）。



5. 自有App接入天猫精灵教程

生活物联网平台提供了免费的天猫精灵快捷通道。您的产品只需定义了平台的标准功能属性，且在天猫精灵支持的品类中，在完成相应的配置后，即可快速连接天猫精灵，实现天猫精灵音箱对设备的控制。

限制条件

- 生活物联网平台遵循ICA数据标准，为了确保您的设备可以被天猫精灵准确识别和控制，请确保使用标准功能。
- 仅支持在中国境内激活的设备对接天猫精灵。

消费者使用

阿里IoT的公版免开发App支持天猫精灵控制，使用方式如下步骤所示。

1. 用户拥有一台支持天猫精灵的设备，下载厂家自有品牌App，绑定该设备。
2. 用户拥有一台天猫精灵音箱，从手机端应用商店下载天猫精灵App并绑定该音箱。
3. 在厂家自有品牌App中，用户授权淘宝账号登录，成功绑定天猫精灵。

 说明 App账号和淘宝账号是一一对应的。

4. 用户在天猫精灵App的设备列表中，在我家页签中可以看到绑定的设备。



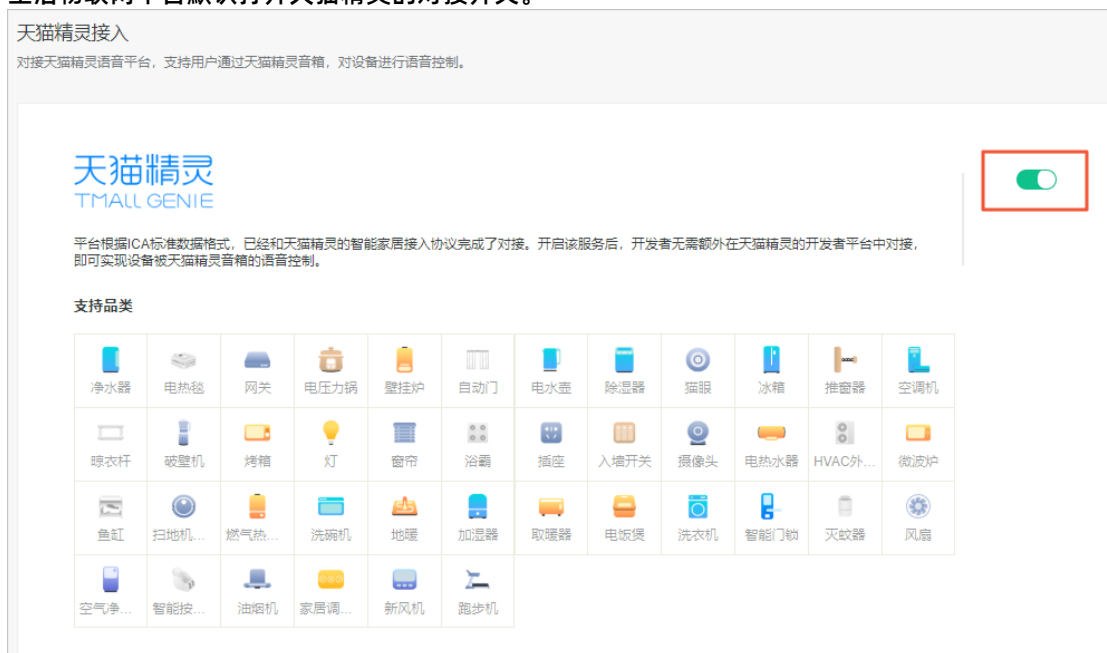
5. 完成以上步骤后，您可以通过天猫精灵音箱控制您的设备。

控制台开通

1. 确认已打开天猫精灵的控制开关。

- 选择一个现有的项目，或创建一个新项目。参见[创建项目](#)。
- 创建一个新产品，参见[创建产品](#)。
- 定义产品功能，参见[新增标准功能](#)。
- 添加测试设备，参见[新增测试设备](#)。
- 配置App功能参数项。详情请参见[配置App功能参数](#)。

生活物联网平台默认打开天猫精灵的对接开关。



2. 单击品类图标，查看功能映射。

生活物联网平台已经默认完成了部分天猫精灵品类功能的映射。这些已经完成映射的指令，您无需再自行开发，只需确保设备使用了映射过的标准属性，即可被天猫精灵控制。

3. 调试设备。

使用公版App配网绑定了您的测试设备后，可以根据下方消费者使用步骤，绑定天猫精灵，即可实现音箱的控制。下载公版App请参见[公版App介绍](#)。

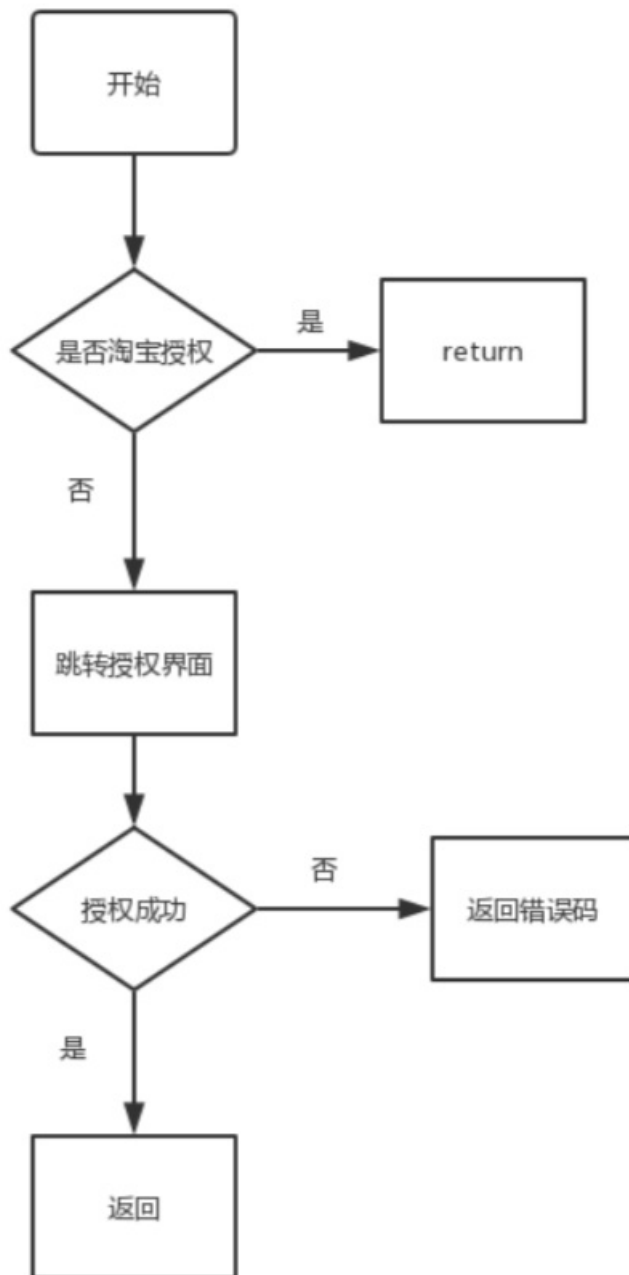
 **注意** 首次注册账号时，国家和地区请选择中国内地。

4. 创建一个自有App，参见[创建自有App](#)。

开发Android自有App

由于天猫精灵使用淘宝账号体系，需要在您的自有品牌App中，集成淘宝的账号授权，从而实现和天猫精灵的对接。

1. 调用淘宝登录授权页面的唤起接口。
开发步骤如下图所示。



2. 单击授权按钮，进入授权界面。
加载WebView，授权网址URL如下（需要拼接AppKey）。

```
String url = "https://oauth.taobao.com/authorize?response_type=code&client_id=<项目的appKey>&r
```

3. 在shouldOverrideUrlLoading中判断地址是否包含code，获取code并传到上一步访问URL后的界面。

```
mWebView.setWebViewClient(new WebViewClient() {  
    //设置结束加载函数  
    @Override  
    public void onPageFinished(WebView view, String url) {  
        topbar.setTitle(view.getTitle());  
    }  
    @Override  
    public boolean shouldOverrideUrlLoading(WebView view, String url) {  
        if (isTokenUrl(url)) {  
            Intent intent = new Intent();  
            intent.putExtra("AuthCode", mAuthCode);  
            setResult(RESULT_CODE, intent);  
            finish();  
            return true;  
        }  
        view.loadUrl(url);  
        return false;  
    }  
});  
  
private boolean isTokenUrl(String url) {  
    if (!TextUtils.isEmpty(url)) {  
        if (url.contains("code=")) {  
            String[] urlArray = url.split("code=");  
            if (urlArray.length > 1) {  
                String[] paramArray = urlArray[1].split("&");  
                if (paramArray.length > 1) {  
                    mAuthCode = paramArray[0];  
                    return true;  
                }  
            }  
        }  
    }  
    return false;  
}
```

4. 根据code绑定淘宝账号。

示例代码如下，其中 /account/taobao/bind 接口的说明，请参见[账号服务](#)。

```
public void bindAccount(String authCode) {

    JSONObject params = new JSONObject();
    if (null != authCode) {
        params.put("authCode", authCode);
    }
    Map<String, Object> requestMap = params.getInnerMap();

    IoTRequest iOTRequest = new IoTRequestBuilder()
        .setAuthType("iotAuth")
        .setApiVersion("1.0.5")
        .setPath("/account/taobao/bind")
        .setParams(requestMap)
        .setScheme(Scheme.HTTPS)
        .build();
    new IoTAPIClientFactory().getClient().send(iOTRequest, new IoTCallback() {
        @Override
        public void onFailure(IOTRequest iOTRequest, Exception e) {

        }

        @Override
        public void onResponse(IOTRequest iOTRequest, IoTResponse iOTResponse) {

        }
    });
}
```

5. （可选）调用接口解除绑定。

```
// 用户解绑淘宝Id
IoTRequestBuilder builder = new IoTRequestBuilder();
builder.setPath("/account/thirdparty/unbind");
builder.setApiVersion("1.0.5");
builder.setAuthType("iotAuth");
builder.addParam("accountType","TAOBAO");

IoTRequest iOTRequest = builder.build();
new IoTAPIClientFactory().getClient().send(iOTRequest, new IoTCallback() {
    @Override
    public void onFailure(IOTRequest iOTRequest, final Exception e) {}
    @Override
    public void onResponse(IOTRequest iOTRequest, final IoTResponse iOTResponse) {}
});
//account/thirdparty/unbind接口的说明，请参见账号服务
//此处para = @{"accountType":"TAOBAO"}
```

6. 判断是否已绑定。

/account/thirdparty/get 接口的说明，请参见[账号服务](#)。

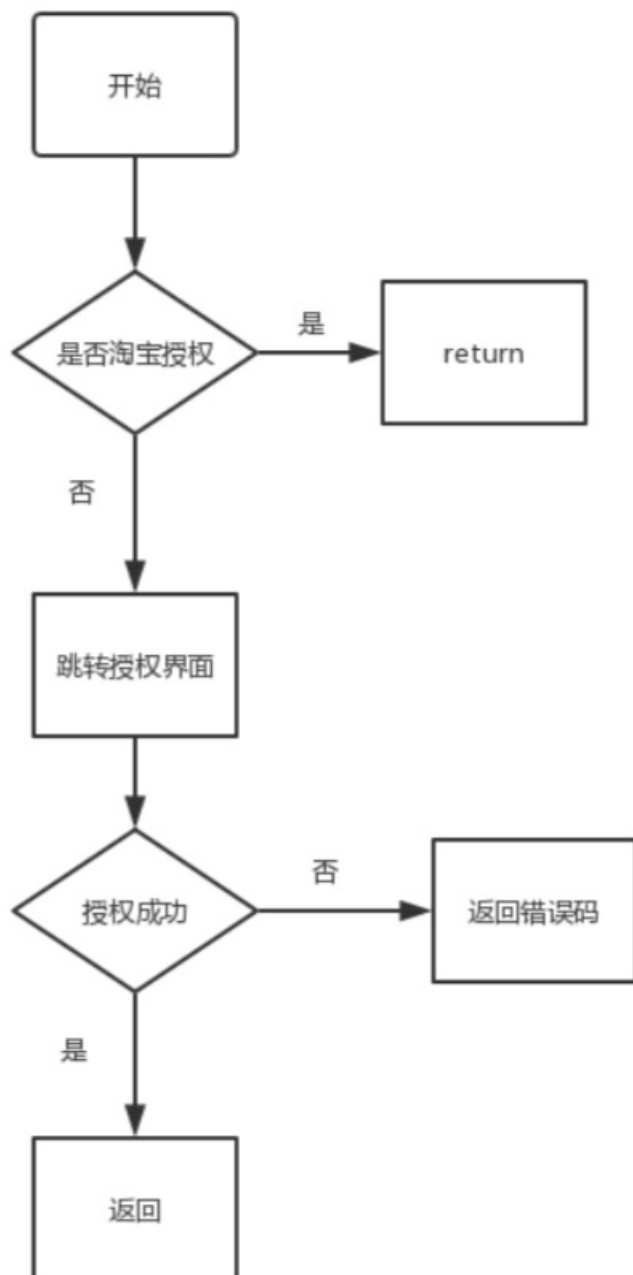
```
// 查询当前用户绑定淘宝Id
IoTRequestBuilder builder = new IoTRequestBuilder();
builder.setPath("/account/thirdparty/get");
builder.setApiVersion("1.0.5");
builder.setAuthType("iotAuth");
builder.addParam("accountType","TAOBAO");

IoTRequest iOTRequest = builder.build();
new IoTAPIClientFactory().getClient().send(iOTRequest, new IoTCallback() {
    @Override
    public void onFailure(IOTRequest iOTRequest, final Exception e) {}
    @Override
    public void onResponse(IOTRequest iOTRequest, final IoTResponse iOTResponse) {}
});
```

开发iOS自有App

由于天猫精灵使用淘宝账号体系，需要在您的自有品牌App中，集成淘宝的账号授权，从而实现和天猫精灵的对接。

1. 调用淘宝登录授权页面的唤起接口。
开发步骤如下图所示。



2. 单击授权按钮，进入授权界面。

加载WebView，授权网址URL如下（需要拼接AppKey）。

```
https://oauth.taobao.com/authorize?response_type=code&client_id=<项目的appKey>&redirect_uri=
```

初始化一个WebView，并设置代理加载。示例代码如下。

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    [super viewDidLoad];  
    WKWebView *webView = [[WKWebView alloc] init];  
    [self.view addSubview:webView];  
    [webView mas_makeConstraints:^(MASConstraintMaker *make) {  
        make.left.equalTo(self.view);  
        make.right.equalTo(self.view);  
        make.top.equalTo(self.view);  
        make.bottom.equalTo(self.view);  
    }];  
    webView.navigationDelegate = self; //代理：WKNavigationDelegate  
    [webView loadRequest:[NSURLRequest requestWithURL:[NSURL URLWithString:[NSString stringWith]]
```

3. 加载页面成功后，使用真实的淘宝账号登录。





4. 登录成功后，处理您在控制台填写的回调地址。

天猫精灵快速通道

天猫精灵接入

自有品牌授权天猫精灵，无需开发天猫精灵技能，根据文档进行账号授权即可。

[查看文档](#)

App信息：

上传icon图片
图片尺寸512*512

回调地址：

保存

重置

```
- (void)webView:(WKWebView *)webView decidePolicyForNavigationAction:(WKNavigationAction *)n
```

```

NSRange range = [navigationAction.request.URL.absoluteString rangeOfString:@"控制台定义的回调
if (range.location){
    //允许跳转
    decisionHandler(WKNavigationActionPolicyAllow);
} else {
    //不允许跳转
    decisionHandler(WKNavigationActionPolicyCancel);
    NSURLComponents *components = [NSURLComponents componentsWithString:navigationAction

for (NSURLQueryItem *item in components.queryItems){
    if ([item.name isEqualToString:@"code"]){
        //用户绑定淘宝ID请求:此处IMSTmallSpeakerApi 通过下面代码封装一个基础请求类
        [IMSTmallSpeakerApi bindTaobaoidWithParams:@{@"authCode":item.value} completion:^(N

        if (self.completion){
            self.completion(err, result);
        }
    }
}

```

```

    }
    [self.navigationController popViewControllerAnimated:YES];
  }];
  break;
}
}
}
}

// 封装的IMSTmallSpeakerApi请求类，依赖请求类#import <IMSApiClient/IMSApiClient.h>
// .h中
+ (void)requestTmallSpeakerApi:(NSString *)api
    version:(NSString *)ver
    params:(NSDictionary *)para
    completion:(void (^)(NSError *, id))completion;

// .m中
+ (void)requestTmallSpeakerApi:(NSString *)api
    version:(NSString *)ver
    params:(NSDictionary *)para
    completion:(void (^)(NSError *, id))completion {

    IMSIoTRequestBuilder *builder = [[IMSIoTRequestBuilder alloc] initWithPath:api
                                     apiVersion:ver
                                     params:para];

    [builder setScheme:@"https"];

    IMSRequest *request = [[builder setAuthenticationType:IMSAuthenticationTypeIoT] build];
    [IMSRequestClient asyncSendRequest:request responseHandler:^(NSError * _Nullable error, IMSRe
    if (completion) {
        // 返回请求过期后，需要重新登录；重新登录后重新初始化主框架，不需要重新请求
        if (response.code == 401) {
            if (NSClassFromString(@"IMSAccountService") != nil) {
                // 先退出登录
                if ([[IMSAccountService sharedInstance] isLogin]) {
                    [[IMSAccountService sharedInstance] logout];
                }
                return;
            }
        }
    }
}
}

```

```

        if (!error && response.code == 200) {
            completion(error, response.data);
            return ;
        }

        NSError *bError = [NSError errorWithDomain:NSURLErrorDomain
                                           code:response.code
                                           userInfo:@{NSLocalizedDescriptionKey: response.localizedMsg ? : @"服务器"}];
        completion(bError, nil);
        return;
    }
};
}

```

```

// 用户绑定淘宝Id
+ (void)bindTaobaoldWithParams:(NSDictionary *)para
    completion:(void (^)(NSError *, NSDictionary *))completion{
    [self requestTmallSpeakerApi:@" /account/taobao/bind" version:@"1.0.5" params:para completion:completion];
    ///account/taobao/bind接口说明请参见账号服务
}
//此处para = @{@"authCode":@"xxxx"}, 其中xxxx为网页回调的code，具体查看登录成功后的回调处理步骤

```

处理回调后返回code200，表示授权成功。此时请刷新网页。

5. （可选）调用接口解除绑定。

```

// 用户解绑淘宝Id
+ (void)unbindTaobaoldWithParams:(NSDictionary *)para
    completion:(void (^)(NSError *, NSDictionary *))completion{
    [self requestTmallSpeakerApi:@" /account/thirdparty/unbind" version:@"1.0.5" params:para completion:completion];
    ///account/thirdparty/unbind接口的说明，请参见账号服务
    //此处para = @{@"accountType":@"TAOBAO"}
}

```

6. 判断是否已绑定。

/account/thirdparty/get 接口的说明，请参见[账号服务](#)。

```
// 查询用户绑定的淘宝Id
+ (void)getTaobaoldWithParams:(NSDictionary *)para
    completion:(void (^)(NSError *, NSDictionary *))completion{
    [self requestTmallSpeakerApi:@"account/thirdparty/get" version:@"1.0.5" params:para completion:]
}
//此处para = @{@"accountType":@"TAOBAO"}
```

返回数据存在以下内容代表已绑定。

```
{
  accountId = 90xxx335;
  accountType = TAOBAO;
  linkIdentityId = 50ebop9xxxxxxxxxxxxx8dbc58643423092968;
}
```

6. 自有App定制Amazon Alexa技能

生活物联网平台支持您的自有App接入Amazon Alexa，并定制上架专属于您自己独特品牌的技能，提升品牌价值和竞争力。

Amazon Alexa技能可以让带有您品牌标识的技能在Amazon Alexa技能列表中展示，终端用户搜索后即可使用。助您树立品牌形象，提升品牌竞争力。

您可以在生活物联网平台控制台的人机交互页面Amazon Alexa参数项中，查看您的产品是否在Amazon Alexa支持的品类中，以及哪些功能支持Amazon Alexa的语音控制。

您可以自行接入Amazon Alexa，也可以联系我们定制。定制Amazon Alexa技能是一项收费的增值服务（详细请参见[服务计费](#)）。服务提供接入Amazon Alexa的一系列技术协助，包括开发、测试与技能上架等（技能上架的审核结果与所需时间取决于Amazon平台）。

一、准备工作

- 已创建了产品，并完成产品的开发。详细请参见[概述](#)。
- 已创建了自有App，且已集成了App SDK，完成了对产品的配网和控制。详细请参见[创建自有App](#)和App端开发文档。
- 已在iOS App Store或Google Play上架您的自有App。
- 已完成Amazon Alexa和AWS的账号注册。
- 已完成[亚马逊开发者](#)账号的注册。
- 您已开通Amazon Alexa接入功能，并分配到产品所在的项目。
- 在产品的人机交互页面中，已打开产品的Amazon Alexa开关。



二、申请Amazon Alexa能力定制

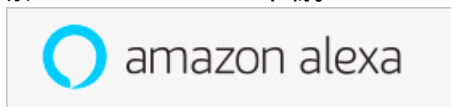
完成准备工作后，您可以通过控制台右上角的工单或您专属的客户代表联系我们，填写Amazon Alexa技能定制申请表。

我们收到申请表后会有技术人员跟您联系，协助您完成Amazon Alexa能力定制。

三、体验效果

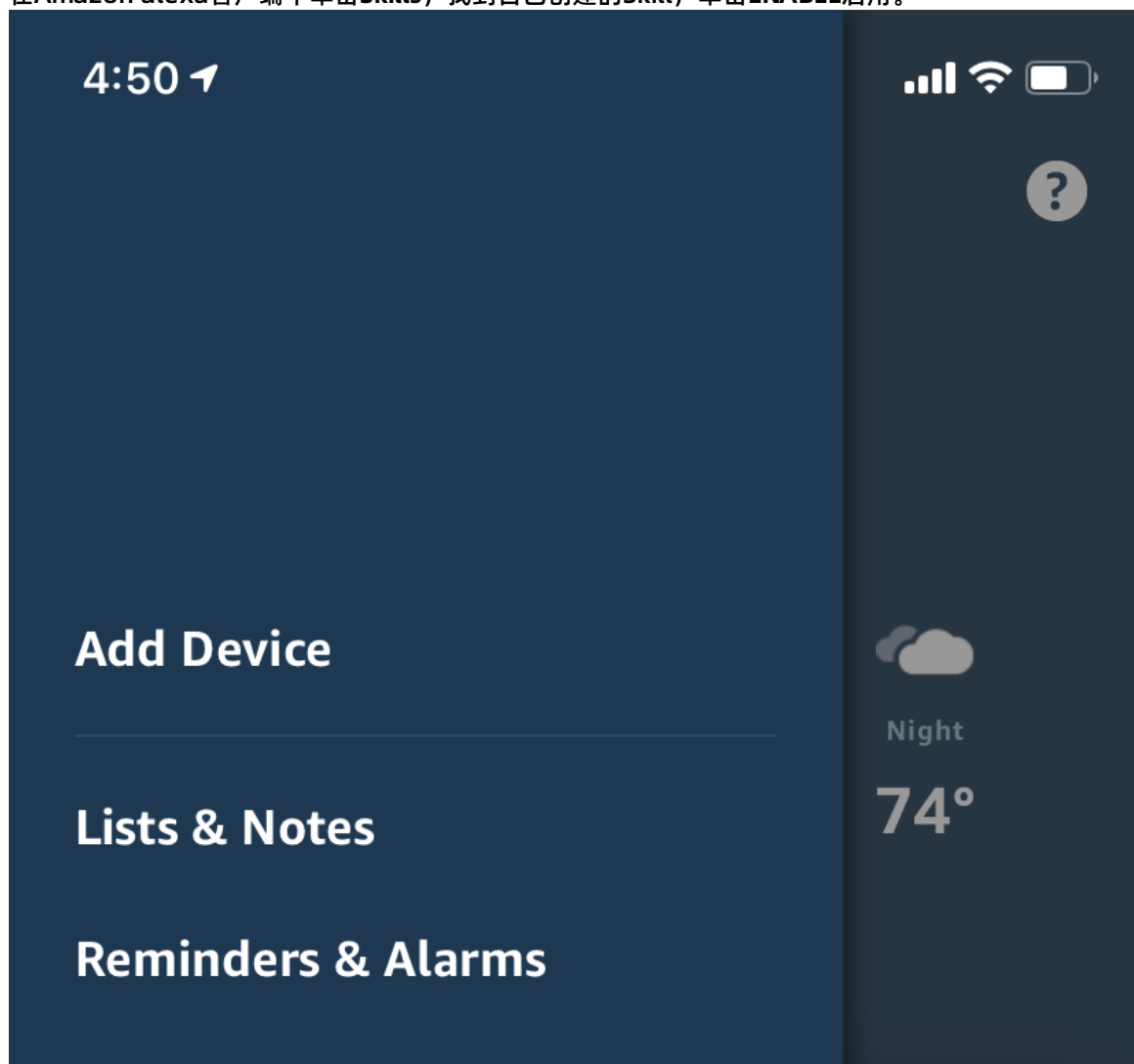
请您根据以下步骤，使用Amazon Echo音箱控制您自有App绑定的设备，从而体验Amazon Echo定制技能。

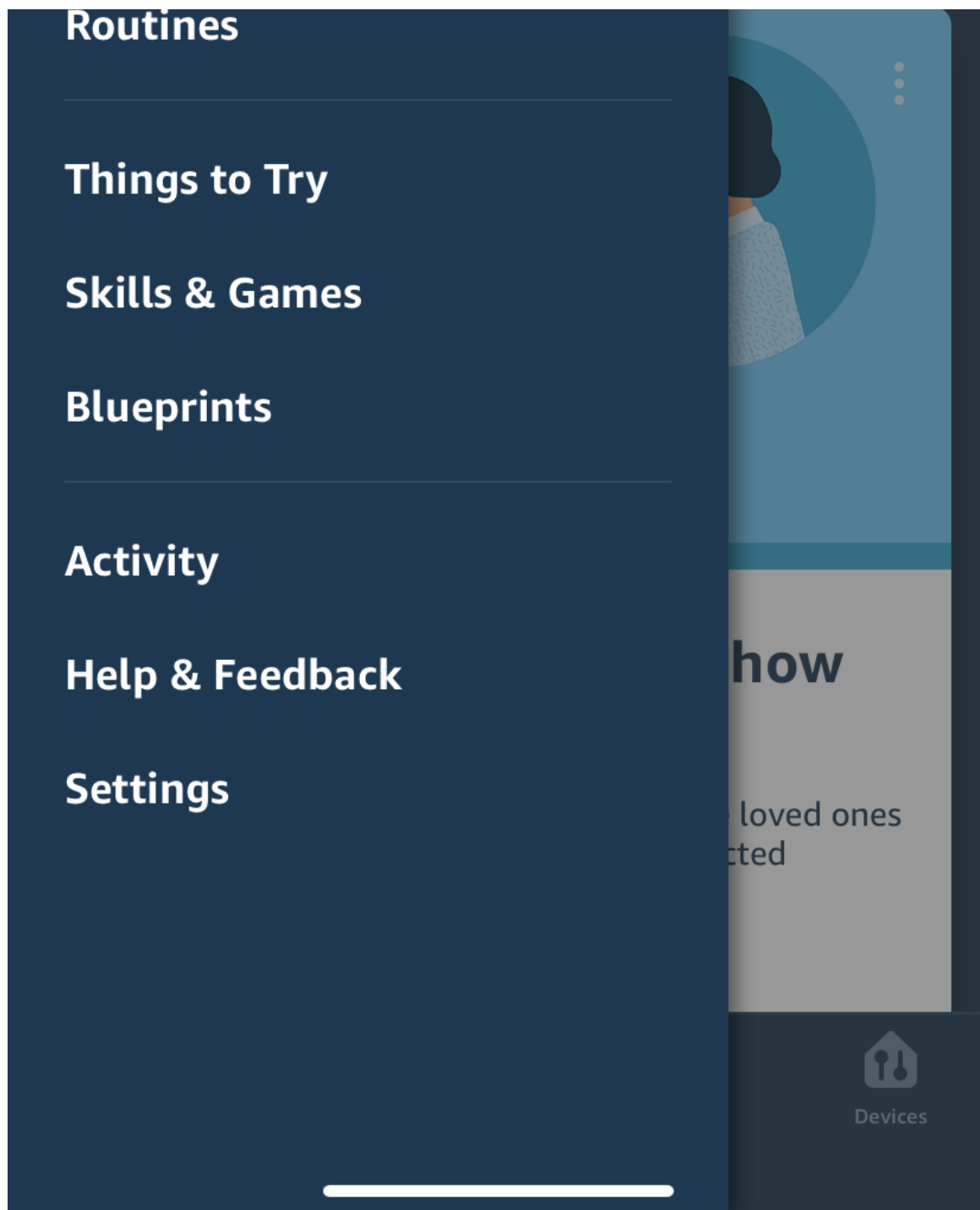
1. 准备一台Amazon Echo音箱，下载Amazon alexa客户端并安装绑定您的Echo音箱。
iOS用户请至美国区App Store搜索“Amazon Alexa”下载，Android客户请至Google Play搜索“Amazon Alexa”下载。



在大陆地区下载Alexa App需要使用国外的应用市场账号，所以国内测试建议通过网页来配置。网页配置地址为：<http://alexa.amazon.com/spa/index.html>

2. 准备一台设备，使用您的自有品牌App进行配网绑定。
3. 在Amazon alexa客户端中单击Skills，找到自己创建的Skill，单击ENABLE启用。





4. 在弹出的界面中输入您自有App登录的账号和密码，并正确选择您的账号所属国家，然后单击**Link Now**来绑定您自有App绑定的设备。
5. 发现设备。
控制设备前，Echo需要先发现设备。
 - 您可以对Echo说：“Alexa, discover devices”。
 - 您也可以在Skill页面单击**DISCOVER**来发现设备。您绑定过的设备就会出现在列表里。



说明 如果您在自有App中修改了设备昵称，必须重新发现设备，否则不能对设备进行控制。

6. （以控制灯为例）通过下列指令控制您的设备。

- 打开/关闭灯： Alexa, turn on/off the light.
- 调节灯亮度： Alexa, brighten/dim the light.
- 调节灯颜色： Alexa, set the light to yellow.
- 调节灯色温： Alexa, set the light to warm white.

7. 自有App定制Google Assistant技能

生活物联网平台支持您的自有App接入Google Assistant，并定制上架专属于您自己独特品牌的Action，提升品牌价值和竞争力。

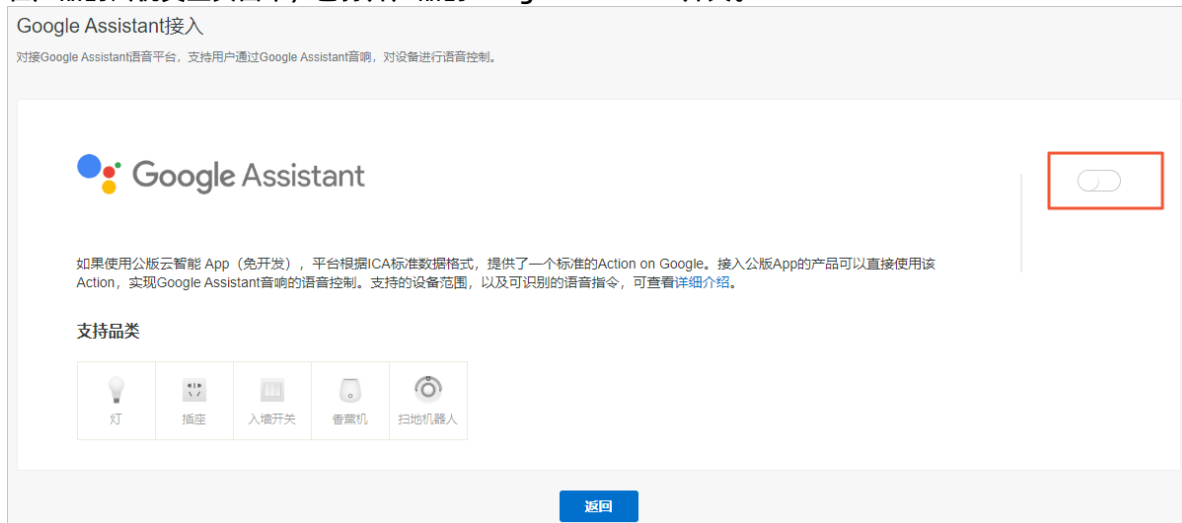
Google Assistant技能可以让带有您品牌标识的技能在Google Assistant的Action列表中展示，终端用户搜索后即可使用。助您树立品牌形象，提升品牌竞争力。

您可以在生活物联网平台控制台的人机交互页面Google Assistant参数项中，查看您的产品是否在Google Assistant支持的品类中，以及哪些功能支持Google Assistant的语音控制。

您可以自行接入Google Assistant，也可以联系我们定制。定制Google Assistant技能是一项收费的增值服务（详细请参见[服务计费](#)）。服务提供接入Google Assistant Action的一系列技术协助，包括开发、测试、技能上架等（技能上架的审核结果与所需时间取决于Google Assistant平台）。

一、准备工作

- 已创建了产品，并完成产品的开发。详细请参见[概述](#)。
- 已创建了自有App，且已集成了App SDK，完成了对产品的配网和控制。详细请参见[创建自有App](#)和App端开发文档。
- 已在iOS App Store或Google Play上架您的自有App。
- 已完成[Google开发者](#)账号的注册。
- Assistant您已开通Google，并分配至产品所在的项目。
- 在产品的人机交互页面中，已打开产品的Google Assistant开关。



二、申请Google Assistant能力定制

完成准备工作后，您可以通过控制台右上角的工单或您专属的客户代表联系我们，填写Google Assistant技能定制申请表。

我们收到申请表后会有技术人员跟您联系，协助您完成Google Assistant能力定制。

三、体验效果

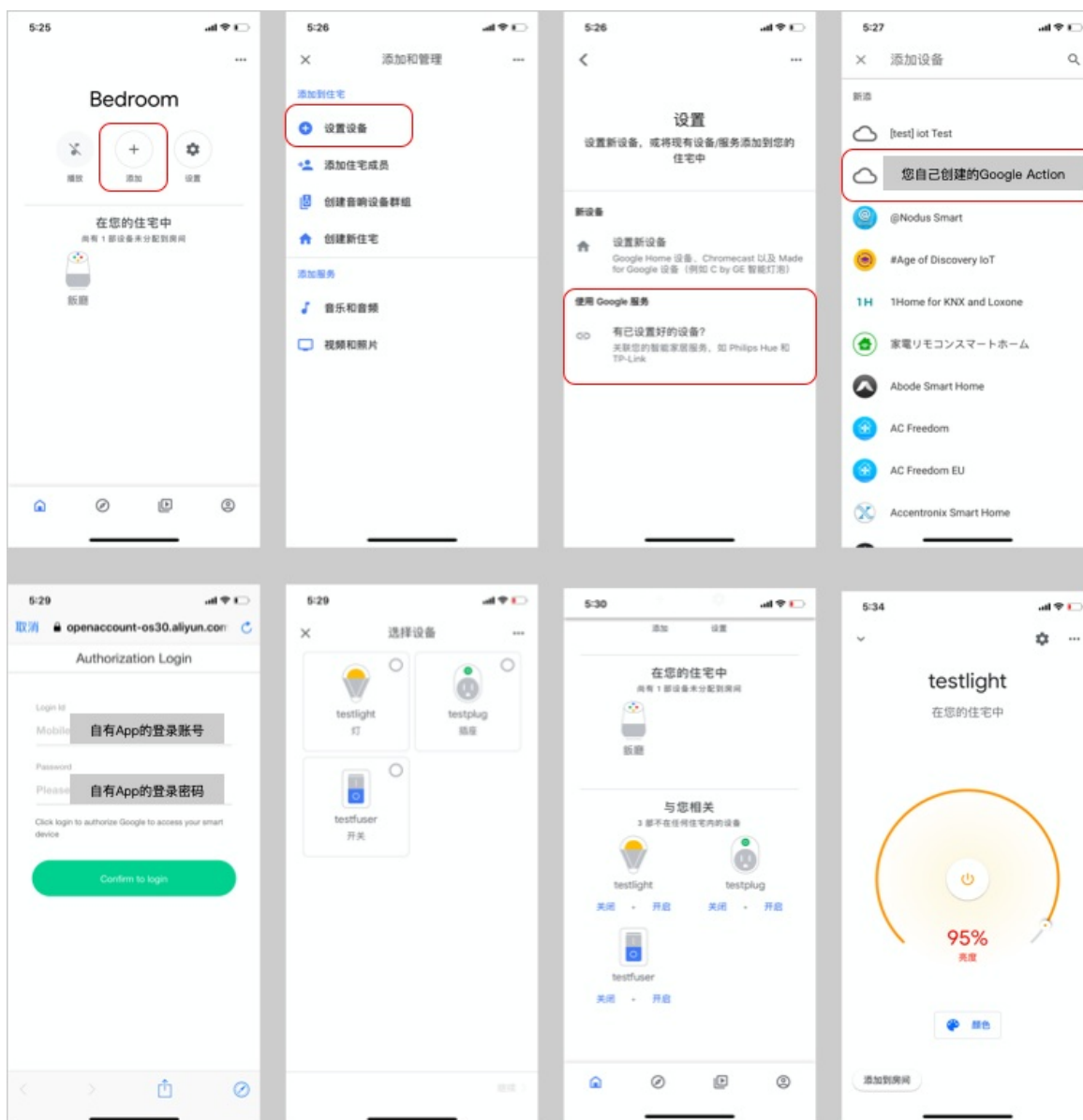
请您根据以下步骤，使用Google Home音箱控制您自有App绑定的设备，从而体验Google Assistant定制技能。

1. 准备一台Google Home音箱，下载Google Home或者Google AssistantApp，安装并绑定到您的音箱。

❓ 说明 iOS系统用户请至美国区App Store搜索Google Home下载，Android用户请至Google Play搜索Google Home下载。



2. 准备一台设备，使用您的自有品牌App进行配网绑定。
3. 在Google Home客户端中添加设备，找到您创建的Google Action。输入您自有品牌App登录的账号和密码，完成添加您的智能设备。



4. 您可以通过下列指令控制您的设备（以操控灯为例）。

- 打开/关闭灯： Hey Google, turn on/off the light.
- 调节灯亮度增强/减弱： Hey Google, brighten/dim the light.
- 调节灯颜色： Hey Google, set the light to yellow.
- 调节灯色温： Hey Google, set the light to warm white.