

阿里云 物联网平台

快速入门

文档版本：20190816

法律声明

阿里云提醒您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。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 创建产品与设备.....	1
2 为产品定义物模型.....	5
3 建立设备与平台的连接.....	16
4 服务端订阅设备消息.....	18
5 设备接收云端指令.....	21

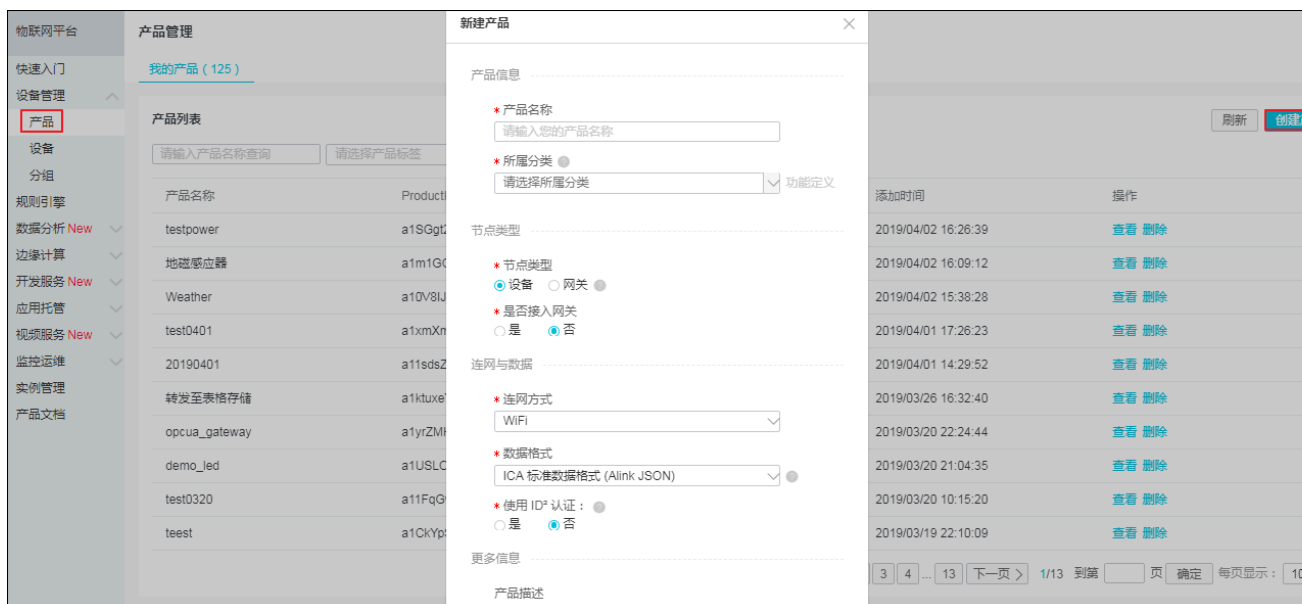
1 创建产品与设备

使用物联网平台的第一步是在云端创建产品和对应设备。产品相当于某一类设备的集合，该类设备具有相同的功能，您可以根据产品批量管理对应设备。

操作步骤

1. 登录[物联网平台控制台](#)。
2. 创建产品。
 - a) 左侧导航栏选择设备管理 > 产品。在产品管理页面，单击创建产品。
 - b) 配置参数。

具体配置细节可参见[创建产品](#)文档完成。



- c) 单击完成。

产品将自动出现在产品列表中。

3. 创建设备。

- a) 左侧导航栏选中设备管理 > 设备，进入设备管理页面。
- b) 单击添加设备。选中上一步创建的产品，输入设备名称（DeviceName），设置设备备注名，单击完成。

物联网平台

快速入门

设备管理 ^

产品

设备

分组

规则引擎

数据分析 v

边缘计算 v

开发服务 v

应用托管 v

视频服务 v

监控运维 v

产品文档

设备管理

全部产品 v

设备列表 批次管理

设备列表

DeviceName v 请输入

<input type="checkbox"/>	DeviceName/备注
<input type="checkbox"/>	Modbus_test_dev
<input type="checkbox"/>	test02
<input type="checkbox"/>	Xgateway1 0419
<input type="checkbox"/>	dht11 firsttry

c) 记录设备证书内容。

设备证书包含ProductKey、DeviceName和DeviceSecret。设备证书是设备后续与物联网平台交流的重要凭证，请妥善保管。

查看设备证书

i 设备证书用于云端对接入的设备做鉴权认证，请妥善保管！

ProductKey ?	a1zl[REDACTED]W76	复制
DeviceName ?	device1	复制
DeviceSecret ?	*****	显示

一键复制 关闭

后续步骤

[#unique_5](#)。

2 为产品定义物模型

物联网平台支持为产品定义物模型，将实际产品抽象成由属性、服务、事件所组成的数据模型，便于云端管理和数据交互。产品创建完成后，您可以为它定义物模型，产品下的设备将自动继承物模型内容。

操作步骤

1. 产品列表中，选中创建的产品，单击查看，进入产品详情页。
2. 单击功能定义。
3. 在自定义功能栏，单击添加功能。

4. 如下图所示，将产品开关定义为属性。完成后单击确认。

* 功能类型：

属性 服务 事件 ?

* 功能名称：

?

* 标识符：

?

* 数据类型：

∨

* 布尔值：

0 -

1 -

读写类型：

读写 只读

描述：

0/100

5. 如下图所示，将计数器定义为属性。完成后单击确认。

* 功能类型：

属性 服务 事件 ?

* 功能名称：

?

* 标识符：

?

* 数据类型：

∨

* 取值范围：

~

* 步长：

单位：

∨

读写类型：

读写 只读

描述：

0/100

6. 如下图所示，将数值计算定义为服务。完成后单击确认。

* 功能类型：

属性 服务 事件 ?

* 功能名称：

?

* 标识符：

?

* 调用方式：

异步 同步 ?

输入参数：

[编辑](#) [删除](#)

[编辑](#) [删除](#)

[+增加参数](#)

输出参数：

[编辑](#) [删除](#)

[+增加参数](#)

描述：

0/100

- 输入参数中，数值A定义如下。

新增参数 ×

*** 参数名称:**
 ?

*** 标识符:**
 ?

*** 数据类型:**
 ▾

*** 取值范围 :**
 ~

*** 步长 :**

单位 :
 ▾

- 输入参数中，数值B定义如下。

新增参数 ✕

* 参数名称:
 ?

* 标识符:
 ?

* 数据类型:
 ▾

* 取值范围:
 ~

* 步长:

单位:
 ▾

- 输出参数为计算结果。

新增参数 ✕

*** 参数名称:**
 ?

*** 标识符:**
 ?

*** 数据类型:**
 ∨

*** 取值范围:**
 ~

*** 步长:**

单位:
 ∨

7. 如下图所示，将故障定义为事件。完成后单击确认。

* 功能类型：
 属性 服务 事件 ?

* 功能名称：
 ?

* 标识符：
 ?

* 事件类型：
 信息 告警 故障 ?

输出参数：
 参数名称：故障编号 编辑 删除

[+增加参数](#)

描述：
 0/100

- 输出参数为故障编号。

新增参数 ✕

*** 参数名称:**
 ?

*** 标识符:**
 ?

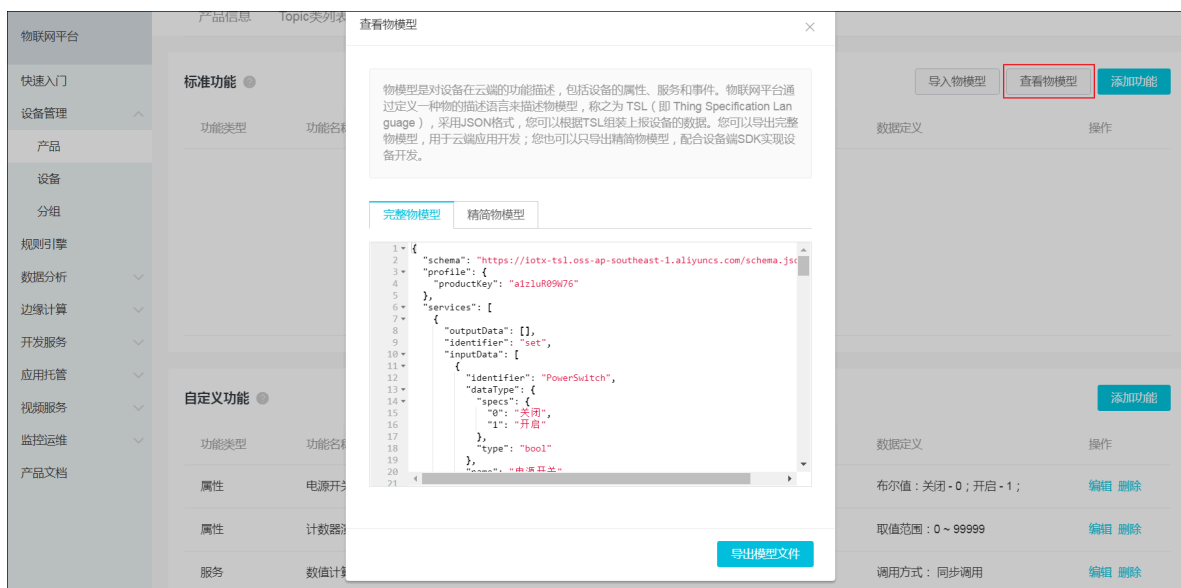
*** 数据类型:**
 ▾

*** 枚举项:**

参数值 ?	~	参数描述 ?	
<input type="text" value="0"/>	~	<input type="text" value="故障类型0"/>	删除
<input type="text" value="1"/>	~	<input type="text" value="故障类型1"/>	删除
<input type="text" value="2"/>	~	<input type="text" value="故障类型2"/>	删除

[+添加枚举项](#)

8. 单击查看物模型，在完整物模型栏下，可看到该产品的完整物模型JSON文件。



后续步骤

[#unique_7](#)

3 建立设备与平台的连接

阿里云物联网平台提供设备端SDK，设备使用SDK与平台建立通信。在这一步里，我们使用平台提供的样例程序linkkit-example-solo模拟设备进行开发，实现设备与物联网平台的通信。

背景信息

- 本文使用Linux下的C语言SDK，该SDK的编译环境推荐使用64位的Ubuntu16.04。
- SDK的开发编译环境会用到以下软件：

```
make-4.1, git-2.7.4, gcc-5.4.0, gcov-5.4.0, lcov-1.12, bash-4.3.48, tar-1.28, mingw-5.3.1
```

可以使用如下命令行安装：

```
sudo apt-get install -y build-essential make git gcc
```

操作步骤

1. 登录Linux虚拟机。
2. 获取[Link Kit SDK](#)。
3. 使用unzip命令解压压缩包。
4. 设备身份信息将通过HAL调用返回给SDK。因此，需要将wrappers/os/ubuntu/HAL_OS_linux.c中的设备证书信息修改为[#unique_9](#)中测试产品的设备证书，完成后保存退出。

如下所示，在此处填入ProductKey、DeviceName和DeviceSecret，设备使用证书进行身份认证并连接物联网平台。



说明：

快速入门中使用[#unique_10](#)的认证方式，ProductSecret可以不填。

```
#ifdef DEVICE_MODEL_ENABLED
char _product_key[IOTX_PRODUCT_KEY_LEN + 1] = "a1zluR09W76";
char _product_secret[IOTX_PRODUCT_SECRET_LEN + 1] = "";
char _device_name[IOTX_DEVICE_NAME_LEN + 1] = "device1";
char _device_secret[IOTX_DEVICE_SECRET_LEN + 1] = "ynNuxxxxxx
xxxx7RMUJD9WZhEvd7ijx0";
```

5. 在SDK顶层目录，执行make命令，完成样例程序的编译。

```
$ make distclean
```

```
$ make
```

生成的样例程序 `linkkit-example-solo` 存放在 `./output/release/bin` 目录下。

6. 运行样例程序。在控制台观察到设备上线。此时，设备与物联网平台成功连接。

```
./output/release/bin/linkkit-example-solo
```

设备上线成功后，会自动向物联网平台上报消息。您可以通过查看日志，获取具体内容。

后续步骤

[服务端订阅设备消息](#)

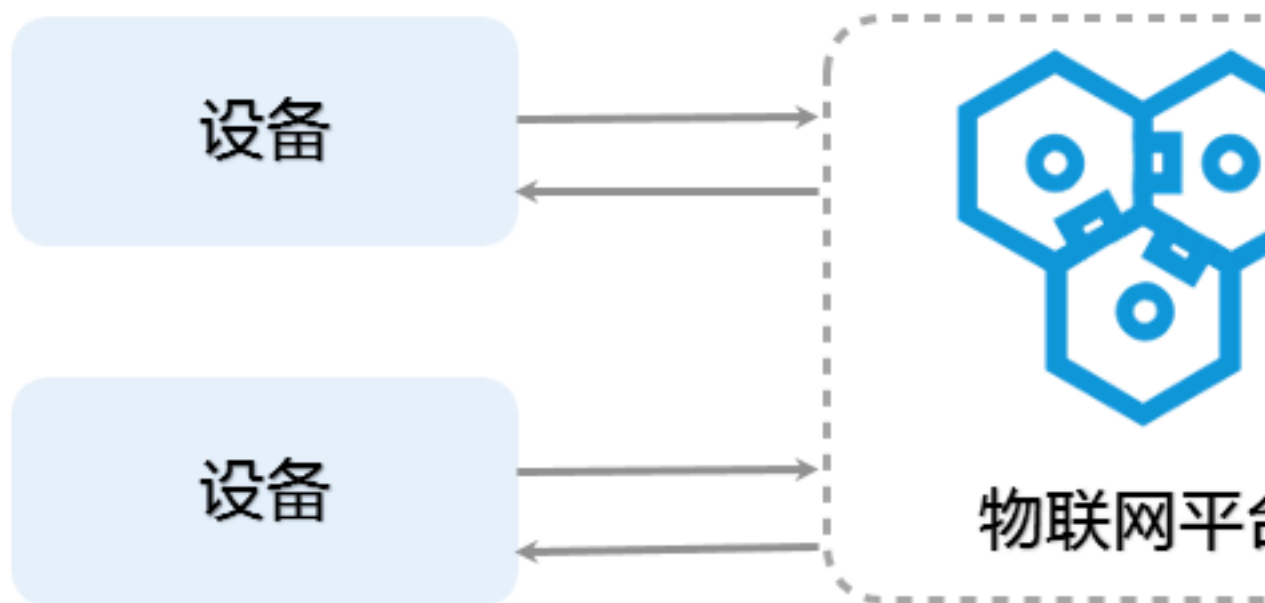
4 服务端订阅设备消息

设备连接物联网平台后，数据直接上报至物联网平台。平台上的数据可以通过HTTP/2通道流转至您的服务器。这一步中，我们将配置HTTP/2服务端订阅功能。您的服务器可以通过接入HTTP/2 SDK，接收设备数据。

背景信息

服务端订阅设备消息流程

图：



操作步骤

1. 在[物联网平台控制台](#)中，为产品配置服务端订阅。
 - a) 在产品管理页，单击已创建的测试产品对应的查看按钮。
 - b) 在产品详情页，单击服务端订阅 > 设置。
 - c) 勾选要订阅的消息类型，然后单击保存。

消息类型	说明
设备上报消息	包括设备上报的自定义数据和物模型数据（包括属性上报、事件上报、属性设置响应和服务调用响应）。
设备状态变化通知	指当产品下设备的状态发生变化时，系统发出的通知消息。例如，设备上下线消息。
设备生命周期变更	包括设备创建、删除、禁用和启用的通知消息。
网关发现子设备上报	网关可以将发现的子设备信息上报给物联网平台。需要网关上的应用程序支持。网关产品特有消息类型。

消息类型	说明
设备拓扑关系变更	指子设备和网关之间的拓扑关系建立和解除消息。网关产品特有消息类型。

2. 接入HTTP/2 SDK。

如果您使用Apache Maven来管理Java项目，需在项目的pom.xml文件中加入以下依赖项。



说明:

请确保JDK版本为8。HTTP/2更多信息，请参见[#unique_13](#)。

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>iot-client-message</artifactId>
  <version>1.1.3</version>
</dependency>

<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.7.1</version>
</dependency>
```

3. 基于您的阿里云账号AccessKey进行身份认证，建立SDK与物联网平台的连接。

```
// 阿里云accessKey
String accessKey = "xxxxxxxxxxxxxxxx";
// 阿里云accessSecret
String accessSecret = "xxxxxxxxxxxxxxxx";
// regionId
String regionId = "cn-shanghai";
// 阿里云uid
String uid = "xxxxxxxxxxxx";
// endPoint: https://{uid}.iot-as-http2.{region}.aliyuncs
.com
String endPoint = "https://" + uid + ".iot-as-http2." +
regionId + ".aliyuncs.com";

// 连接配置
Profile profile = Profile.getAccessKeyProfile(endPoint,
regionId, accessKey, accessSecret);

// 构造客户端
MessageClient client = MessageClientFactory.messageClient(
profile);

// 数据接收
client.connect(messageToken -> {
  Message m = messageToken.getMessage();
  System.out.println("receive message from " + m);
  return MessageCallback.Action.CommitSuccess;
});
```

```
});
```

参数及获取方式:

参数	获取途径
accessKey	您的账号AccessKey ID。 登录阿里云控制台，将光标移至账号头像上，然后单击accesskeys，跳转至用户信息管理页，即可获取。
accessSecret	您的账号AccessKey Secret。获取方式同accessKey。
uid	您的账号ID。 用主账号登录阿里云控制台，单击账号头像，跳转至账号管理控制台，即可获取账号UID。
regionId	您物联网平台服务所在地域代码。 在物联网平台控制台页，右上方即可查看地域（Region）。RegionId的表达方法，请参见文档 地域与可用区 。

4. 确认HTTP/2 SDK可以接收到设备消息。

若监听成功，可以通过SDK的消息回调获得以下数据:

参数	说明
messageId	物联网平台生成的消息ID。
topic	消息来源Topic。
payload	消息数据。请参见 #unique_14 。
generateTime	消息生成时间戳，以毫秒表示。
qos	<ul style="list-style-type: none"> · 0: 消息最多投递1次。 · 1: 消息最少投递1次。

5 设备接收云端指令

设备成功上报消息后，您可以尝试从云端下发指令到设备端。本文档讲解了如何通过云端应用调用SetDeviceProperty接口，设置产品属性。同时，设备成功接收云端指令。

操作步骤

1. 在Maven项目中加入依赖项。

IoT Java SDK的Maven依赖坐标

```
<!-- https://mvnrepository.com/artifact/com.aliyun/aliyun-java-sdk-iot -->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-iot</artifactId>
  <version>6.8.0</version>
</dependency>
```

依赖公共包

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.5.1</version>
</dependency>
```

2. 初始化SDK。

此处地域与产品地域保持一致，使用华东2。

```
String accessKey = "<your accessKey>";
String accessSecret = "<your accessSecret>";
DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Iot", "iot.cn-shanghai.aliyuncs.com");
IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", accessKey, accessSecret);
DefaultAcsClient client = new DefaultAcsClient(profile);
```

3. 云端应用向设备下发属性设置指令，将开关属性设置为1。

示例如下：

```
SetDevicePropertyRequest request = new SetDevicePropertyRequest();
request.setProductKey("a1zluR09W76");
request.setDeviceName("device1");
JSONObject itemJson = new JSONObject();
itemJson.put("PowerSwitch", 1);
request.setItems(itemJson.toString());

try {
  SetDevicePropertyResponse response = client.getAcsResponse(request);
  System.out.println(response.getRequestId() + ", success: " + response.getSuccess());
}
```

```
} catch (ClientException e) {  
    e.printStackTrace();  
}
```

**说明:**

属性设置的具体调用方法，请参见[SetDeviceProperty](#)。

预期结果

若设备成功接收到云端下发的信息。日志示例如下：

```
< {  
<   "method": "thing.service.property.set",  
<   "id": "432801169",  
<   "params": {  
<     "PowerSwitch": 1  
<   },  
<   "version": "1.0.0"  
< }  
  
user_report_reply_event_handler.94: Message Post Reply Received,  
Message ID: 646, Code: 200, Reply: {}  
user_property_set_event_handler.114: Property Set Received, Request:  
{"PowerSwitch":1}  
  
> {  
>   "id": "647",  
>   "version": "1.0",  
>   "params": {  
>     "PowerSwitch": 1  
>   },  
>   "method": "thing.event.property.post"  
> }
```