

阿里云 物联网平台

云端开发指南

文档版本：20190523

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。

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。
##	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
[]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ }或者{a b}	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 云端SDK参考.....	1
1.1 下载云端SDK.....	1
1.2 Java SDK使用说明.....	1
1.3 Python SDK使用说明.....	3
1.4 PHP SDK使用说明.....	4
1.5 .NET SDK使用说明.....	5
2 云端API参考.....	7
2.1 API列表.....	7
2.2 概述.....	11
2.3 调用API.....	11
2.4 公共参数.....	13
2.5 签名机制.....	15
2.6 错误码.....	23
2.7 产品管理.....	41
2.7.1 CreateProduct.....	41
2.7.2 QueryProduct.....	46
2.7.3 QueryProductList.....	49
2.7.4 UpdateProduct.....	53
2.7.5 DeleteProduct.....	54
2.7.6 CreateProductTags.....	55
2.7.7 UpdateProductTags.....	57
2.7.8 DeleteProductTags.....	59
2.7.9 ListProductTags.....	60
2.7.10 ListProductByTags.....	62
2.8 设备管理.....	64
2.8.1 RegisterDevice.....	64
2.8.2 QueryDeviceDetail.....	67
2.8.3 QueryDevice.....	71
2.8.4 DeleteDevice.....	74
2.8.5 GetDeviceStatus.....	76
2.8.6 BatchGetDeviceState.....	78
2.8.7 DisableThing.....	80
2.8.8 EnableThing.....	82
2.8.9 BatchCheckDeviceNames.....	83
2.8.10 BatchRegisterDeviceWithApplyId.....	86
2.8.11 BatchRegisterDevice.....	88
2.8.12 BatchUpdateDeviceNickname.....	89
2.8.13 QueryBatchRegisterDeviceStatus.....	92

2.8.14 QueryPageByApplyId.....	94
2.8.15 QueryDeviceEventData.....	97
2.8.16 QueryDevicePropertyData.....	101
2.8.17 QueryDevicePropertiesData.....	105
2.8.18 QueryDeviceServiceData.....	109
2.8.19 InvokeThingService.....	113
2.8.20 InvokeThingsService.....	116
2.8.21 GetGatewayBySubDevice.....	118
2.8.22 QueryDevicePropertyStatus.....	121
2.8.23 SetDeviceProperty.....	124
2.8.24 SetDevicesProperty.....	127
2.8.25 QueryDeviceProp.....	129
2.8.26 SaveDeviceProp.....	130
2.8.27 DeleteDeviceProp.....	133
2.8.28 QueryDeviceByTags.....	135
2.8.29 GetThingTopo.....	137
2.8.30 NotifyAddThingTopo.....	140
2.8.31 RemoveThingTopo.....	143
2.8.32 QueryDeviceStatistics.....	144
2.8.33 SetDeviceDesiredProperty.....	146
2.8.34 QueryDeviceDesiredProperty.....	149
2.8.35 QueryLoRaJoinPermissions.....	153
2.8.36 CreateLoRaNodesTask.....	155
2.8.37 GetLoraNodesTask.....	157
2.8.38 QueryDeviceFileList.....	158
2.8.39 QueryDeviceFile.....	161
2.8.40 DeleteDeviceFile.....	163
2.9 分组管理.....	165
2.9.1 CreateDeviceGroup.....	165
2.9.2 UpdateDeviceGroup.....	166
2.9.3 DeleteDeviceGroup.....	167
2.9.4 BatchAddDeviceGroupRelations.....	168
2.9.5 BatchDeleteDeviceGroupRelations.....	170
2.9.6 QueryDeviceGroupInfo.....	171
2.9.7 QueryDeviceGroupList.....	172
2.9.8 SetDeviceGroupTags.....	175
2.9.9 QueryDeviceGroupTagList.....	177
2.9.10 QueryDeviceGroupByDevice.....	178
2.9.11 QuerySuperDeviceGroup.....	180
2.9.12 QueryDeviceListByDeviceGroup.....	181
2.9.13 QueryDeviceGroupByTags.....	183
2.10 Topic管理.....	186
2.10.1 CreateProductTopic.....	186
2.10.2 UpdateProductTopic.....	187
2.10.3 QueryProductTopic.....	189

2.10.4 DeleteProductTopic.....	191
2.10.5 CreateTopicRouteTable.....	191
2.10.6 DeleteTopicRouteTable.....	193
2.10.7 QueryTopicReverseRouteTable.....	194
2.10.8 QueryTopicRouteTable.....	195
2.11 规则引擎.....	196
2.11.1 CreateRule.....	196
2.11.2 CreateRuleAction.....	200
2.11.3 DeleteRule.....	209
2.11.4 DeleteRuleAction.....	210
2.11.5 GetRule.....	211
2.11.6 GetRuleAction.....	214
2.11.7 ListRule.....	216
2.11.8 ListRuleActions.....	219
2.11.9 StartRule.....	222
2.11.10 StopRule.....	223
2.11.11 UpdateRule.....	224
2.11.12 UpdateRuleAction.....	227
2.12 消息通信.....	229
2.12.1 Pub.....	229
2.12.2 PubBroadcast.....	231
2.12.3 RRpc.....	233
2.13 设备影子.....	235
2.13.1 GetDeviceShadow.....	235
2.13.2 UpdateDeviceShadow.....	236

1 云端SDK参考

1.1 下载云端SDK

物联网平台云端SDK用于调用云端API，以实现物联网平台的云端能力，如产品管理、设备管理、Topic管理、数据流转规则管理、消息通信等。



说明：

本章节仅介绍云端SDK的使用。设备端SDK开发，请参见[设备端SDK](#)。

云端SDK下载地址

物联网平台提供的云端SDK语言版本有：Java、Python、PHP、.NET。

单击以下链接，进入相应的云端SDK源码下载地址。

- [IoT Java SDK](#)
- [IoT Python SDK](#)
- [IoT PHP SDK](#)
- [IoT .NET SDK](#)

下载云端SDK Demo

阿里云提供物联网平台SDK使用 Demo。Demo 中包含Java、Python、PHP、.NET版本SDK。

点击下载[SDK Demo](#)

1.2 Java SDK使用说明

物联网平台的 Java SDK 让开发人员可以方便地使用 Java 程序操作物联网平台。开发者可以使用Maven依赖添加SDK，也可以下载安装包到本地直接安装。

安装 SDK

1. 安装Java开发环境。

您可以从 [Java 官方网站](#) 下载，并按说明安装Java开发环境。

2. 安装IoT Java SDK。

- a. 访问 [Apache Maven 官网](#) 下载Maven软件。
- b. 添加Maven项目依赖。

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>6.9.0</version>
</dependency>
```

依赖公共包

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

初始化SDK



说明:

以下示例以华东2地域及其服务接入地址为例。您在设置时，需使用您的物联网平台地域和对应的服务接入地址。

```
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); //初始化SDK客户端
```

accessKey即您的账号的AccessKeyId， accessSecret即AccessKeyId对应的AccessKeySecret。您可在[阿里云官网控制台AccessKey管理](#)中创建或查看您的AccessKey。

发起调用

物联网平台云端API，请参见[API列表](#)。

以调用Pub接口发布消息到Topic为例。

```
PubRequest request = new PubRequest();
request.setProductKey("productKey");
request.setMessageContent(Base64.encodeBase64String("hello world".
getBytes()));
request.setTopicFullName("/productKey/deviceName/get");
request.setQos(0); //目前支持QoS0和QoS1
```



```
try
{
    PubResponse response = client.getAcsResponse(request);
    System.out.println(response.getSuccess());
    System.out.println(response.getErrorMessage());
}
catch (ServerException e)
{
    e.printStackTrace();
}
catch (ClientException e)
{
    e.printStackTrace();
}
```

1.3 Python SDK使用说明

物联网平台提供Python语言的云端SDK供开发人员使用。本文介绍云端Python SDK的安装和配置，及使用Python SDK调用云端API的示例。

安装Python SDK

1. 安装Python开发环境。

访问[Python官网](#)下载Python安装包，并完成安装。支持Python 2.6.5及以上版本。

2. 安装Python的包管理工具 pip。（如果您已安装pip，请忽略此步骤。）

访问 [pip 官网](#) 下载pip安装包，并完成安装。

3. 安装IoT Python SDK。

以管理员权限执行以下命令，安装IoT Python SDK。请参见最新版[aliyun-python-sdk-iot](#)信息。

```
sudo pip install aliyun-python-sdk-core
sudo pip install aliyun-python-sdk-iot
```

4. 将IoT Python SDK相关文件引入Python文件。

```
from aliyunsdkcore import client
from aliyunskiot.request.v20180120 import RegisterDeviceRequest
from aliyunskiot.request.v20180120 import PubRequest
...
```

初始化SDK

```
accessKeyId = '<your accessKey>'
accessKeySecret = '<your accessSecret>'
clt = client.AcsClient(accessKeyId, accessKeySecret, 'cn-shanghai')
```

accessKeyId即您的账号的AccessKeyId， accessKeySecret即AccessKeyId对应的AccessKeySecret。您可在[阿里云官网控制台AccessKey管理](#)中创建或查看您的AccessKey。

发起调用

以调用Pub接口发布消息到设备为例。

```
request = PubRequest.PubRequest()
request.set_accept_format('json') #设置返回数据格式，默认为XML，此例中设置为JSON
request.set_ProductKey('productKey')
request.set_TopicFullName('/productKey/deviceName/get') #消息发送到的Topic全名
request.set_MessageContent('aGVsbG8gd29ybGQ=') #hello world Base64 String
request.set_Qos(0)
result = clt.do_action_with_exception(request)
print 'result : ' + result
```

1.4 PHP SDK使用说明

物联网平台提供PHP语言的云端SDK供开发人员使用。本文介绍云端PHP SDK的安装和配置，及使用PHP SDK调用云端API的示例。

安装IoT PHP SDK

1. 安装PHP开发环境。

访问[PHP官网](#)下载PHP安装包，并完成安装。

2. 下载并解压IoT PHP SDK软件包。

访问[IoT PHP SDK下载地址](#)下载PHP SDK 包，然后解压下载的软件包到指定的目录。PHP SDK是一个软件开发包，不需要进行安装操作。

初始化SDK

初始化SDK示例

```
<?php
include_once 'aliyun-php-sdk-core/Config.php';
use \Iot\Request\V20180120 as Iot;
//设置你的AccessKeyId/AccessSecret/ProductKey
$accessKeyId = "";
$accessSecret = "";
$iClientProfile = DefaultProfile::getProfile("cn-shanghai", $accessKeyId, $accessSecret);
$client = new DefaultAcsClient($iClientProfile);
```

accessKeyId即您的账号的AccessKeyId， accessSecret即AccessKeyId对应的AccessKeySecret。您可在[阿里云官网控制台AccessKey管理](#)中创建或查看您的AccessKey。

发起调用

物联网平台云端API，请参见[API列表](#)。

以调用Pub接口发布数据到设备为例。

```
$request = new Iot\PubRequest();
$request->setProductKey("productKey");
$request->setMessageContent("aGVsbG93b3JsZA="); //hello world Base64
String.
$request->setTopicFullName("/productKey/deviceName/get"); //消息发送到的
Topic全名.
$response = $client->getAcsResponse($request);
print_r($response);
```

1.5 .NET SDK使用说明

物联网平台提供.NET语言的云端SDK供开发人员使用。本文介绍云端.NET SDK的安装和配置，及使用.NET SDK调用云端API的示例。

安装 IoT .NET SDK

1. 安装.NET开发环境。

建议安装：

- Microsoft .NET Framework 4.0及以上版本。
- Visual Studio 2010 及以上版本。

2. 下载阿里云 .NET SDK核心库包和物联网平台.NET SDK包。

在[.NET SDK发布列表](#)中，分别单击 SDK 核心库和阿里云物联网平台对应的适用于.NET 4.0及以上的DLL引用，即可下载 .NET SDK的核心库和物联网平台.NET SDK的压缩包。解压后获得DLL文件。

3. 添加核心库的引用。

- a. 在Visual Studio的解决方案资源管理器中，选中您的项目，右键引用。
- b. 在弹出的菜单栏中，选择添加引用。
- c. 在弹出的添加引用对话框中，选择浏览。
- d. 选择之前下载并解压的物联网平台.NET SDK的DLL文件（*aliyun-net-sdk-Iot.dll*和*aliyun-net-sdk-Core.dll*），单击确认。

初始化SDK



说明：

以下示例以华东2地域及其服务接入地址为例。您在设置时，需使用您的物联网平台地域和对应的服务接入地址。

```
using Aliyun.Acs.Core;
using Aliyun.Acs.Core.Exceptions;
```

```
using Aliyun.Acs.Core.Profile;
DefaultProfile.AddEndpoint("cn-shanghai", "cn-shanghai", "Iot", "iot.
cn-shanghai.aliyuncs.com");
IClientProfile clientProfile = DefaultProfile.GetProfile("cn-shanghai",
"<your-access-key-id>", "<your-access-key-secret>");
DefaultAcsClient client = new DefaultAcsClient(clientProfile);
```

请在[阿里云官网控制台AccessKey管理](#)中创建或查看您的AccessKeyId和AccessKeySecret。

发起调用

物联网平台云端API，请参见[API列表](#)。

以调用Pub接口向Topic发布消息为例。

```
PubRequest request = new PubRequest();
request.ProductKey = "<productKey>";
request.TopicFullName = "/<productKey>/<deviceName>/get";
byte[] payload = Encoding.Default.GetBytes("Hello World.");
String payloadStr = Convert.ToBase64String(payload);
request.MessageContent = payloadStr;
request.Qos = 0;
try
{
    PubResponse response = client.GetAcsResponse(request);
    Console.WriteLine("publish message result: " + response.Success);
    Console.WriteLine(response.ErrorMessage);
}
catch (ServerException e)
{
    Console.WriteLine(e.ErrorCode);
    Console.WriteLine(e.ErrorMessage);
}
catch (ClientException e)
{
    Console.WriteLine(e.ErrorCode);
    Console.WriteLine(e.ErrorMessage);
}
```

2 云端API参考

2.1 API列表

以下是物联网平台 API 列表。

产品管理相关 API

API	描述
CreateProduct	创建产品。
UpdateProduct	修改产品信息。
QueryProductList	查询产品列表。
QueryProduct	查询产品详细信息。
DeleteProduct	删除指定产品。
CreateProductTags	创建产品标签。
UpdateProductTags	更新产品标签。
DeleteProductTags	删除产品标签。
ListProductTags	查询产品的所有标签。
ListProductByTags	根据标签查询产品。

设备管理相关 API

API	描述
RegisterDevice	注册设备。
QueryDeviceDetail	查询设备详情。
QueryDevice	查询产品的设备列表。
DeleteDevice	删除设备。
GetDeviceStatus	获取设备的运行状态。
BatchGetDeviceState	批量获取设备状态。
DisableThing	禁用设备。
EnableThing	解禁设备。
BatchCheckDeviceNames	批量检查设备名称。

API	描述
BatchRegisterDeviceWithApplyId	根据 ApplyId 批量申请设备。
BatchRegisterDevice	批次申请特定数量设备。
QueryBatchRegisterDeviceStatus	查询批量注册设备状态。
QueryPageByApplyId	查询批次设备列表。
QueryDeviceEventData	查询设备的事件历史数据。
QueryDevicePropertyData	查询设备的属性历史数据。
QueryDevicePropertiesData	批量查询指定设备的多个属性的历史数据。
QueryDeviceServiceData	获取设备的服务记录历史数据。
InvokeThingService	调用设备的服务。
InvokeThingsService	批量调用设备的服务。
QueryDevicePropertyStatus	查询设备的属性快照。
SetDeviceProperty	设置设备的属性。
SetDevicesProperty	批量设置设备属性。
SaveDeviceProp	设置设备标签。
QueryDeviceProp	查询设备标签列表。
DeleteDeviceProp	删除设备标签。
GetThingTopo	查询网关设备或子设备所具有的拓扑关系。
NotifyAddThingTopo	通知云端增加设备拓扑关系。
RemoveThingTopo	移除网关设备或子设备所具有的拓扑关系。
QueryDeviceStatistics	获取设备的统计数量。
GetGatewayBySubDevice	根据挂载的子设备信息查询对应的网关设备信息。
QueryDeviceByTags	根据标签查询设备。
SetDeviceDesiredProperty	为指定设备批量设置期望属性值。
QueryDeviceDesiredProperty	查询指定设备的期望属性值。
QueryDeviceFileList	查询指定设备上传到物联网平台的所有文件。
QueryDeviceFile	查询指定设备上传到物联网平台的指定文件信息。
DeleteDeviceFile	删除指定设备上传到物联网平台的指定文件。

API	描述
BatchUpdateDeviceNickname	批量更新设备备注名称。
QueryLoRaJoinPermissions	查询账号下的LoRaWAN入网凭证列表。
CreateLoRaNodesTask	创建批量注册LoRaWAN设备的任务。
GetLoraNodesTask	查询批量注册LoRaWAN设备任务的状态。

分组管理相关API

API	描述
CreateDeviceGroup	创建分组。
DeleteDeviceGroup	删除分组。
UpdateDeviceGroup	修改分组信息。
QueryDeviceGroupInfo	查询分组详情。
QueryDeviceGroupList	分页查询分组列表。
BatchAddDeviceGroupRelations	添加设备到分组。
BatchDeleteDeviceGroupRelations	删除分组中已添加的指定设备。
SetDeviceGroupTags	添加或更新分组标签。
QueryDeviceGroupTagList	查询分组标签列表。
QueryDeviceGroupByDevice	查询指定设备所在的分组列表。
QuerySuperDeviceGroup	根据子分组ID查询父分组信息。
QueryDeviceListByDeviceGroup	查询分组中的设备列表。
QueryDeviceGroupByTags	根据标签查询设备分组。

规则引擎相关 API

API	描述
ListRule	查询规则列表。
CreateRule	创建规则。
GetRule	查询规则信息。
UpdateRule	修改规则。
DeleteRule	删除规则。
ListRuleActions	查询规则动作列表。

API	描述
GetRuleAction	查询规则动作信息。
CreateRuleAction	创建规则动作。
UpdateRuleAction	更新规则动作。
DeleteRuleAction	删除规则动作。
StartRule	启动规则。
StopRule	停止规则。

Topic 管理相关 API

API	描述
QueryProductTopic	查询产品Topic类。
CreateProductTopic	创建产品Topic类。
UpdateProductTopic	修改产品Topic类。
DeleteProductTopic	删除产品Topic类。
CreateTopicRouteTable	添加Topic路由表。
QueryTopicRouteTable	查询Topic路由表。
QueryTopicReverseRouteTable	查询Topic反向路由表。
DeleteTopicRouteTable	删除Topic路由表。

消息通信相关 API

API	描述
Pub	发布消息到Topic。
RRpc	发消息给设备，并同步返回响应。
PubBroadcast	发布广播消息。

设备影子相关 API

API	描述
GetDeviceShadow	查询设备影子。
UpdateDeviceShadow	更新设备影子。

2.2 概述

通过向 API 的服务端地址发送 HTTPS/HTTP GET/POST 请求，并按照 API 接口说明，在请求中加入相应请求参数来调用 API 接口。根据请求的处理情况，系统会返回处理结果。

我们将从以下方面介绍 API 的调用。

- [调用API](#)
- [公共参数](#)

为更好的保护您的阿里云账号安全，建议使用RAM子账号用户身份来调用API。

为子账号授予IoT API 访问权限，请参见 [IoT API 授权映射表](#)。

2.3 调用API

本文档主要介绍调用物联网平台云端API的请求结构和请求示例。

请求结构

您可以通过发送 HTTP /HTTPS 请求调用物联网平台 API。

请求结构如下：

```
http://Endpoint/?Action=xx&Parameters
```

参数	说明
Endpoint	调用的云服务的接入地址。物联网平台的接入地址格式：<code>iot.\${RegionId}.aliyuncs.com</code>。其中，变量<code>\${RegionId}</code>需替换为您的物联网平台服务的地域代码。有关阿里云地域代码，请参见地域和可用区文档 接入地址示例： · 华东2（上海）：<code>iot.cn-shanghai.aliyuncs.com</code> · 新加坡：<code>iot.ap-southeast-1.aliyuncs.com</code> · 美国（硅谷）：<code>iot.us-west-1.aliyuncs.com</code> · 日本（东京）：<code>iot.ap-northeast-1.aliyuncs.com</code> · 德国（法兰克福）：<code>iot.eu-central-1.aliyuncs.com</code>
Action	要执行的操作，即云端API接口的名称。例如，调用Pub接口向指定Topic发布消息，Action对应的值就是Pub，即Action=Pub。

参数	说明
Parameters	请求参数。每个参数之间用（&）符号分隔。 请求参数由公共请求参数和API自定义参数组成。公共参数中包含API版本号、身份验证等信息。

下面以调用Pub接口向指定Topic发布消息为例：



说明：

本文档示例均使用华东2（上海）地域的接入地址。为了便于阅读，本文档中的示例均做了格式化处理。

```
https://iot.cn-shanghai.aliyuncs.com/?Action=Pub
&Format=XML
&Version=2017-04-20
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgI%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=...
&Timestamp=2017-07-19T12:00:00Z
&RegionId=cn-shanghai
...
```

API在线调试

阿里云提供API在线调试工具 [OpenAPI Explorer](#)。在OpenAPI Explorer页，您可以快速检索和试验调用API。系统会根据您输入的参数同步生成各语言SDK的Demo代码。各语言SDK Demo显示在页面右侧示例代码页签下供您参考。在调试结果页签下，查看API调用的真实请求URL和JSON格式的返回结果。

The screenshot displays the OpenAPI Explorer web interface. On the left, a sidebar lists various Alibaba Cloud services under '全部产品' (All Products). The '阿里云物联网平台' (Alibaba Cloud IoT Platform) is selected. A search bar contains the word 'Create'. The main panel shows the 'CreateProduct' API endpoint. It includes a 'RegionId' dropdown set to '华东2 (上海)'. Below are input fields for 'NodeType', 'IotInstanceId', 'ProductName', 'DataFormat', and 'Description'. A '发起调用' (Initiate Call) button is at the bottom right. On the right side, the '示例代码' (Example Code) tab is active, showing a Java SDK demo code snippet for the 'CreateProduct' API. The code includes imports for the IoT SDK and Gson, and a 'main' method that demonstrates how to create a product and print the response.

API授权

为了确保您的账号安全，建议您使用子账号的身份凭证调用API。如果您使用RAM子账号调用物联网平台API，您需要为该RAM子账号创建、授予相应的授权策略。

为子账号授权调用API，请参见[IoT API 授权映射表](#)。

2.4 公共参数

本文档列举了物联网平台API的公共请求参数和公共返回参数。

公共请求参数

公共请求参数是调用每个API时都需要使用的请求参数。

名称	类型	是否必需	描述
Format	String	否	返回值的类型，支持JSON和XML类型。默认为XML。
Version	String	是	API版本号，为日期形式：YYYY-MM-DD，最新版本为2018-01-20。每个接口可以存在多个版本。
AccessKeyId	String	是	阿里云颁发给用户的访问服务所用的密钥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，即可创建和查看AccessKey。
Signature	String	是	签名结果串。
SignatureMethod	String	是	签名方式，目前支持HMAC-SHA1。
Timestamp	String	是	请求的时间戳。日期格式按照ISO8601标准表示，并需要使用UTC时间。格式为YYYY-MM-DDThh:mm:ssZ 例如，2016-01-04T12:00:00Z（为北京时间2016年01月04日20点0分0秒）
SignatureVersion	String	是	签名算法版本。目前版本是1.0。

名称	类型	是否必需	描述
SignatureNonce	String	是	唯一随机数。用于防止网络重放攻击。用户在不同请求中要使用不同的随机数值
RegionId	String	是	设备所在地域（与控制台上的Region对应），如 cn-shanghai。

示例

```
https://iot.cn-shanghai.aliyuncs.com/  
?Format=XML  
&Version=2018-01-20  
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgI%3D  
&SignatureMethod=HMAC-SHA1  
&SignatureNonce=15215528852396  
&SignatureVersion=1.0  
&AccessKeyId=...  
&Timestamp=2018-05-20T12:00:00Z  
&RegionId=cn-shanghai
```

公共返回参数

API返回结果采用统一格式，返回2xx HTTP状态码代表调用成功；返回4xx或5xx HTTP状态码代表调用失败。调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式。默认为XML格式。

每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId

- 调用成功

- XML格式

```
<?xml version="1.0" encoding="UTF-8"?>  
<!--结果的根结点-->  
<接口名称+Response>  
  <!--返回请求标签-->  
  <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>  
  <!--返回结果数据-->  
</接口名称+Response>
```

- JSON格式

```
{  
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216"  
  /* 返回结果数据 */  
}
```

```
}
```

- 调用失败

调用接口出错后，将不会返回结果数据。调用方可根据错误码来定位错误原因。

当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。另外，还包含一个全局唯一的请求ID：RequestId。在您不能确认错误的情况下，可以联系阿里云客服，并提供该RequestId，以便我们尽快帮您解决问题。

- XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <Code>UnsupportedOperation</Code>
  <Message>The specified action is not supported.</Message>
</Error>
```

- JSON格式

```
{
  "RequestId": "8906582E-6722-409A-A6C4-0E7863B733A5",
  "Code": "UnsupportedOperation",
  "Message": "The specified action is not supported."
}
```

2.5 签名机制

物联网平台会对每个接口访问请求的发送者进行身份验证，所以无论使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名（Signature）信息。

签名方法

签名时，您需在控制台 [AccessKey 管理](#) 页面查看您的阿里云账号的 AccessKeyId 和 AccessKeySecret，然后进行对称加密。其中，AccessKeyId 用于标识访问者身份；AccessKeySecret 是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密。



说明：

物联网平台提供了Java、Python、PHP等语言的服务端SDK。使用我们提供的SDK，可以免去签名过程。请参见[下载云端SDK](#)及各SDK的使用说明。

请按照下面的方法对请求进行签名：

1. 构造规范化的请求字符串（Canonicalized Query String）。

a. 排序参数。

按参数名的字典顺序，对请求参数进行排序，包括 [公共请求参数](#)（不包括 Signature 参数）和接口的自定义参数。



说明:

当使用 GET 方法提交请求时，这些参数就是请求 URL 中的参数部分，即 URL 中?之后由&连接的部分。

b. 对参数名称和参数值进行 URL 编码。

使用UTF-8字符集按照 [RFC3986](#) 规则编码请求参数名和参数值。编码规则如下：

- 字符 A~Z、a~z、0~9 以及字符-、_、.、~不编码。
- 其它字符编码成%XY的格式，其中XY是字符对应 ASCII 码的 16 进制表示。比如英文的双引号"对应的编码为%22。
- 扩展的UTF-8字符，编码成%XY%ZA...的格式。
- 英文空格要编码成%20，而不是加号+。

该编码方式与application/x-www-form-urlencoded MIME格式编码算法相似，但又有所不同。

如果您使用的是Java标准库中的java.net.URLEncoder，可以先用标准库中percentEncode编码，随后将编码后的字符中加号+替换为%20、星号*替换为%2A、%7E替换为波浪号~，即可得到上述规则描述的编码字符串。

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws
UnsupportedEncodingException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace(
        "+", "%20").replace("*", "%2A").replace("%7E", "~") : null;
}
```

c. 使用等号=连接编码后的请求参数名和参数值。

d. 使用与号&连接编码后的请求参数。参数排序与步骤 a 的排序一致。

完成后，即获得规范化请求字符串（CanonicalizedQueryString）。

2. 构造签名字符串。

可以使用 percentEncode 处理步骤 1 得到的规范化字符串，构造签名字符串。可参考如下规则：

```
StringToSign=
```

```
HTTPMethod + "&" + //HTTPMethod: 发送请求的 HTTP 方法, 例如 GET。  
percentEncode("/") + "&" + //percentEncode("/"): 字符 (/) UTF-8 编码得  
到的值, 即 %2F。  
percentEncode(CanonicalizedQueryString) //您的规范化请求字符串。
```

3. 计算 HMAC 值。

按照 [RFC2104](#) 的定义, 使用步骤 2 得到的字符串StringToSign计算签名 HMAC 值。示例使用的是Java Base64编码方法。

```
Signature = Base64( HMAC-SHA1( AccessSecret, UTF-8-Encoding-Of(  
StringToSign) ) )
```



说明:

计算签名时, 使用的 Key 就是您的 AccessKeySecret 并加上一个与号&字符 (ASCII:38)。使用的哈希算法是 SHA1。

4. 计算签名值。

按照 Base64 编码规则把步骤 3 中的 HMAC 值编码成字符串, 即得到签名值 (Signature)。

5. 添加签名。

将得到的签名值作为 Signature 参数, 按照 [RFC3986](#) 的规则进行 URL 编码后, 再添加到请求参数中, 即完成对请求签名的过程。

签名示例

以调用Pub接口为例。假设您的AccessKeyId=testid, AccessKeySecret=testsecret。

1. 签名前的请求URL:

```
http://iot.cn-shanghai.aliyuncs.com/?MessageContent=aGVsbG93b3JsZA%3D&Action=Pub&Timestamp=2017-10-02T09%3A39%3A41Z&SignatureVersion=1.0&ServiceCode=iot&Format=XML&Qos=0&SignatureNonce=0715a395-aedf-4a41-bab7-746b43d38d88&Version=2017-04-20&AccessKeyId=testid&SignatureMethod=HMAC-SHA1&RegionId=cn-shanghai&ProductKey=12345abcdeZ&TopicFullName=%2FproductKey%2Ftestdevice%2Fget
```

2. 计算得到的待签名字符串StringToSign:

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DPub%26Format%3DXML%26MessageContent%3DaGVsbG93b3JsZA%253D%26ProductKey%3D12345abcdeZ%26Qos%3D0%26RegionId%3Dcn-shanghai%26ServiceCode%3Diot%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D0715a395-aedf-4a41-bab7-746b43d38d88%26SignatureVersion%3D1.0%26Timestamp%3D2017-10-02T09
```

```
%253A39%253A41Z%26TopicFullName%3D%252FproductKey%252Ftestdevice%252Fget%26Version%3D2017-04-20
```

3. 计算签名值。

因为AccessKeySecret=testsecret，用于计算的Key为testsecret&，计算得到的签名值为：

```
Y9eWn4nF8QPh3c4zAFkM/k/u7eA=
```

4. 将签名作为Signature参数加入到URL请求中，最后得到的URL为：

```
http://iot.cn-shanghai.aliyuncs.com/?MessageContent=aGVsbG93b3JsZA%3D&Action=Pub&Timestamp=2017-10-02T09%3A39%3A41Z&SignatureVersion=1.0&ServiceCode=iot&Format=XML&Qos=0&SignatureNonce=0715a395-aedf-4a41-bab7-746b43d38d88&Version=2017-04-20&AccessKeyId=testid&Signature=Y9eWn4nF8QPh3c4zAFkM%2Fk%2Fu7eA%3D&SignatureMethod=HMAC-SHA1&RegionId=cn-shanghai&ProductKey=12345abcdeZ&TopicFullName=%2FproductKey%2Ftestdevice%2Fget
```

JAVA代码示例

以下为签名的JAVA demo 供您参考。

1. 配置文件 Config.java

```
/*
 * Copyright © 2018 Alibaba. All rights reserved.
 */
package com.aliyun.iot.demo.sign;

/**
 * 服务端API签名配置文件
 *
 * @author: ali
 * @version: 0.1 2018-08-08 08:23:54
 */
public class Config {

    // AccessKey 信息
    public static String accessKey = "1234567890123456";
    public static String accessKeySecret = "123456789012345678901234567890";

    public final static String CHARSET_UTF8 = "utf8";
}
```

2. UrlUtil.java

```
/*
 * Copyright © 2018 Alibaba. All rights reserved.
 */
package com.aliyun.iot.demo.sign;

import java.net.URLEncoder;
import java.util.Map;

import org.apache.commons.lang3.StringUtils;
```



```
/**
 * URL处理类
 *
 * @author: ali
 * @version: 0.1 2018-06-21 20:40:52
 */
public class UrlUtil {

    private final static String CHARSET_UTF8 = "utf8";

    public static String urlEncode(String url) {
        if (!StringUtils.isEmpty(url)) {
            try {
                url = URLEncoder.encode(url, "UTF-8");
            } catch (Exception e) {
                System.out.println("Url encode error:" + e.getMessage());
            }
        }
        return url;
    }

    public static String generateQueryString(Map<String, String> params
, boolean isEncodeKV) {
        StringBuilder canonicalizedQueryString = new StringBuilder();
        for (Map.Entry<String, String> entry : params.entrySet()) {
            if (isEncodeKV)
                canonicalizedQueryString.append(percentEncode(entry.getKey())).
append("=")
                .append(percentEncode(entry.getValue())).append("&");
            else
                canonicalizedQueryString.append(entry.getKey()).append("=").
append(entry.getValue()).append("&");
        }
        if (canonicalizedQueryString.length() > 1) {
            canonicalizedQueryString.setLength(canonicalizedQueryString.
length() - 1);
        }
        return canonicalizedQueryString.toString();
    }

    public static String percentEncode(String value) {
        try {
            // 使用URLEncoder.encode编码后, 将"+","*","%7E"做替换即满足 API规定的编
            码规范
            return value == null ? null
                : URLEncoder.encode(value, CHARSET_UTF8).replace("+", "%20").
replace("*", "%2A").replace("%7E",
                "~");
        } catch (Exception e) {
        }
        return "";
    }
}
```

3. SignatureUtils.java

```
/*
 * Copyright © 2018 Alibaba. All rights reserved.
 */
package com.aliyun.iot.demo.sign;

import java.io.IOException;
```

```
import java.io.UnsupportedEncodingException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URLDecoder;
import java.net.URLEncoder;
import java.util.Map;
import java.util.TreeMap;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

import org.apache.commons.codec.binary.Base64;
import org.apache.commons.lang3.StringUtils;

/**
 * 服务端API签名
 *
 * @author: ali
 * @version: 0.1 2018-06-21 20:47:05
 */
public class SignatureUtils {

    private final static String CHARSET_UTF8 = "utf8";
    private final static String ALGORITHM = "UTF-8";
    private final static String SEPARATOR = "&";

    public static Map<String, String> splitQueryString(String url)
        throws URISyntaxException, UnsupportedEncodingException {
        URI uri = new URI(url);
        String query = uri.getQuery();
        final String[] pairs = query.split("&");
        TreeMap<String, String> queryMap = new TreeMap<String, String>();
        for (String pair : pairs) {
            final int idx = pair.indexOf("=");
            final String key = idx > 0 ? pair.substring(0, idx) : pair;
            if (!queryMap.containsKey(key)) {
                queryMap.put(key, URLDecoder.decode(pair.substring(idx + 1),
                    CHARSET_UTF8));
            }
        }
        return queryMap;
    }

    public static String generate(String method, Map<String, String>
        parameter, String accessKeySecret)
        throws Exception {
        String signString = generateSignString(method, parameter);
        System.out.println("signString---" + signString);
        byte[] signBytes = hmacSHA1Signature(accessKeySecret + "&",
            signString);
        String signature = newStringByBase64(signBytes);
        System.out.println("signature----" + signature);
        if ("POST".equals(method))
            return signature;
        return URLEncoder.encode(signature, "UTF-8");
    }

    public static String generateSignString(String httpMethod, Map<
        String, String> parameter) throws IOException {
        TreeMap<String, String> sortParameter = new TreeMap<String, String>
            >();
        sortParameter.putAll(parameter);
        String canonicalizedQueryString = UrlUtil.generateQueryString(
            sortParameter, true);
    }
}
```

```

    if (null == httpMethod) {
        throw new RuntimeException("httpMethod can not be empty");
    }
    StringBuilder stringToSign = new StringBuilder();
    stringToSign.append(httpMethod).append(SEPARATOR);
    stringToSign.append(percentEncode("/")).append(SEPARATOR);
    stringToSign.append(percentEncode(canonicalizedQueryString));
    return stringToSign.toString();
}

public static String percentEncode(String value) {
    try {
        return value == null ? null
            : URLEncoder.encode(value, CHARSET_UTF8).replace("+", "%20").
            replace("*", "%2A").replace("%7E",
                "~");
    } catch (Exception e) {
    }
    return "";
}

public static byte[] hmacSHA1Signature(String secret, String
baseString) throws Exception {
    if (StringUtils.isEmpty(secret)) {
        throw new IOException("secret can not be empty");
    }
    if (StringUtils.isEmpty(baseString)) {
        return null;
    }
    Mac mac = Mac.getInstance("HmacSHA1");
    SecretKeySpec keySpec = new SecretKeySpec(secret.getBytes(
CHARSET_UTF8), ALGORITHM);
    mac.init(keySpec);
    return mac.doFinal(baseString.getBytes(CHARSET_UTF8));
}

public static String newStringByBase64(byte[] bytes) throws
UnsupportedEncodingException {
    if (bytes == null || bytes.length == 0) {
        return null;
    }
    return new String(Base64.encodeBase64(bytes, false), CHARSET_UTF8
);
}
}

```

4. 主入口 Main.java

```

/*
 * Copyright © 2018 Alibaba. All rights reserved.
 */
package com.aliyun.iot.demo.sign;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;

/**
 * 签名工具主入口
 *
 * @author: ali
 * @version: 0.1 2018-09-18 15:06:48

```

```
*/
public class Main {

    // 1.需求修改Config.java中的AccessKey信息
    // 2.建议使用方法二，所有参数都需要一一填写
    // 3."最终signature"才是你需要的签名最终结果
    public static void main(String[] args) throws UnsupportedEncodingException {

        // 方法一
        System.out.println("方法一: ");
        String str = "GET&%2F&AccessKeyId%3D" + Config.accessKey
            + "%26Action%3DRegisterDevice%26DeviceName%3D1533023037%
26Format%3DJSON%26ProductKey%3DaxxxUtgaRLB%26RegionId%3Dcn-shanghai
%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D1533023037%
26SignatureVersion%3D1.0%26Timestamp%3D2018-07-31T07%253A43%253A57Z%
26Version%3D2018-01-20";
        byte[] signBytes;
        try {
            signBytes = SignatureUtils.hmacSHA1Signature(Config.accessKeyS
ecret + "&", str.toString());
            String signature = SignatureUtils.newStringByBase64(signBytes);
            System.out.println("signString---" + str);
            System.out.println("signature----" + signature);
            System.out.println("最终signature: " + URLEncoder.encode(signature
, Config.CHARSET_UTF8));
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println();

        // 方法二
        System.out.println("方法二: ");
        Map<String, String> map = new HashMap<String, String>();
        // 公共参数
        map.put("Format", "JSON");
        map.put("Version", "2018-01-20");
        map.put("AccessKeyId", Config.accessKey);
        map.put("SignatureMethod", "HMAC-SHA1");
        map.put("Timestamp", "2018-07-31T07:43:57Z");
        map.put("SignatureVersion", "1.0");
        map.put("SignatureNonce", "1533023037");
        map.put("RegionId", "cn-shanghai");
        // 请求参数
        map.put("Action", "RegisterDevice");
        map.put("DeviceName", "1533023037");
        map.put("ProductKey", "axxxUtgaRLB");
        try {
            String signature = SignatureUtils.generate("GET", map, Config.
accessKeySecret);
            System.out.println("最终signature: " + signature);
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println();
    }
}
```

```
}
```

2.6 错误码

本文档列举调用物联网平台API出错时，返回的错误信息。入参数据格式错误、超出限定值、入参缺少必需参数等错误修改，请参见具体API文档的请求参数描述。

系统错误码

以`iot.system`开头的错误码为系统相关错误码。

错误码	描述
<code>iot.system.SystemException</code>	系统异常。 请稍后重试。

公共错误码

以`iot.common`开头的错误码为公共错误码。

错误码	描述
<code>iot.common.InvalidPageParams</code>	分页大小或者分页页号不合法。 请参见具体API文档的分页相关参数描述，如 <code>PageSize</code> 。
<code>iot.common.InvalidTenant</code>	不合法的租户。 请确认阿里云账号信息和账号权限。
<code>iot.common.QueryDeviceActionError</code>	查询设备失败。 请确认入参信息正确，然后重试。
<code>iot.common.QueryDevicePropertyActionError</code>	查询设备属性失败。 请确认入参信息正确，然后重试。
<code>iot.common.QueryProductActionError</code>	查询产品失败。 请确认入参信息正确，然后重试。
<code>iot.common.QueryProductCountActionError</code>	查询产品总数失败。 请确认入参信息正确，然后重试。

错误码	描述
iot.common.RamActionPermissionDeny	没有资源访问控制（RAM）权限。 请参见 子账号授权文档 。
iot.common.AuthActionPermissionDeny	鉴权失败。 原因可能是入参的设备信息不属于当前账号。请确认AccessKey信息和设备信息。

产品（Product）相关错误码

以iot.prod开头的错误码为产品相关错误码。

错误码	描述
iot.prod.AlreadyExistedProductName	已经存在相同的产品名称。一个阿里云账号下的产品名称不能重复。
iot.prod.CreateProductFailed	创建产品失败。 请确认入参信息正确，然后重试。
iot.prod.CreateProductTopicFailed	创建产品的Topic类失败。 请确认入参信息正确，然后重试。
iot.prod.InvalidAliyunCommodityCode	入参AliyunCommodityCode值错误。 AliyunCommodityCode的可选值只有iothub_senior和iothub。
iot.prod.InvalidFormattedCatId	入参CategoryId（产品的设备类型）错误。
iot.prod.InvalidFormattedProductkey	入参产品ProductKey格式错误。 请核对输入的ProductKey值。
iot.prod.InvalidFormattedProductName	入参产品名称格式错误。 产品名应满足以下限制：由中文、英文字母、数字和下划线（_）组成，长度为4-30位（一个中文字符占两位）。
iot.prod.LongProductDesc	产品描述字符数超出限定值。 描述信息应在100字符以内。

错误码	描述
iot.prod.InvalidNodeType	产品的节点类型错误。 节点类型支持的可选值： · 0：设备 · 1：网关
iot.prod.NotExistedProduct	产品不存在。 输入的ProductKey值在当前账号下不存在。
iot.prod.NotOpenID2Service	没有开通ID ² 服务。 该产品在创建时没有开通ID ² 安全认证服务。 ID ² 安全认证服务只能在创建产品时开通，并且，产品创建成功后，不能更改是否使用ID ² 认证的状态。
iot.prod.NotSeniorProduct	产品不是高级版产品。
iot.prod.NullProductKey	入参产品ProductKey不能为空。
iot.prod.NullProductName	入参产品名称不能为空。
iot.prod.ProductCountExceedMax	产品总数已超过最大限制数量。 一个阿里云账号下最多可有1,000个产品。
iot.prod.QueryDeviceCountActionError	查询产品下的设备总数失败。 请确认入参信息正确，然后重试。
iot.prod.QueryProductAbilitiesFailed	获取产品功能失败。 请确认入参信息是否正确，如Identifier值等。
iot.prod.QueryProductAbilityFailed	查询产品功能失败。 请确认入参信息是否正确，如Identifier值等。
iot.prod.QueryProductListActionError	获取产品列表数据失败。 请确认入参信息正确，然后重试。

错误码	描述
iot.prod.UpdateProductFailed	更新产品信息失败。 请确认入参信息正确，然后重试。

设备（Device）相关错误码

以*iot.device*开头的错误码为设备相关错误码。

错误码	描述
iot.device.AddTopoRelationFailed	添加拓扑关系失败。 请确认入参信息正确，然后重试。
iot.device.AlreadyExistedDeviceName	设备名称已经存在。 设备名称需在产品维度唯一。
iot.device.ApplyManyDevicesFailed	申请批量创建设备失败。 请确认入参信息正确，然后重试。
iot.device.CreateDeviceFailed	创建设备失败。 请确认入参信息正确，然后重试。
iot.device.CreateDeviceTaskIsRunning	创建设备的申请任务还在执行中。
iot.device.DeviceApplyIsNotFound	申请设备的申请单不存在。 请确认输入的ApplyId值。其值需与您调用 BatchCheckDeviceNames 返回的ApplyId值一致。
iot.device.DeviceCountExceeded	批量申请的设备数量超过最大值。 单次调用，最多批量注册1,000 个设备。
iot.device.DeleteDeviceFailed	删除设备失败。 请确认入参信息正确，然后重试。
iot.device.DeleteDevicePropertyFailed	删除设备属性失败。 请确认入参信息正确，然后重试。
iot.device.DisableDeviceFailed	禁用设备失败。 请确认入参信息正确，然后重试。

错误码	描述
iot.device.EnableDeviceFailed	启用设备失败。 请确认入参信息正确，然后重试。
iot.device.InactiveDevice	设备未激活，即物理设备从未连接物联网平台。
iot.device.InvalidFormattedApplyId	创建设备的申请单（ApplyId）错误。 其值需与您调用 BatchCheckDeviceNames 返回的ApplyId值一致。
iot.device.IncorrentDeviceApplyInfo	设备申请信息错误。 请确认入参信息，如ApplyId等。
iot.device.InvalidFormattedDeviceName	设备名称格式错误。 设备名称长度为4-32个字符，可以包含英文字母、数字和特殊字符：连字符（-）、下划线（_）、at符号（@）、点号（.）、和英文冒号（:）。
iot.device.InvalidFormattedDevicePropertyKey	设备属性标识符格式错误。 请查看相关API文档中，关于入参属性格式的描述。
iot.device.InvalidFormattedDevicePropertiesString	入参设备属性格式错误。 请查看相关API文档中，关于入参属性格式的描述。
iot.device.InvalidIoTId	设备ID错误。 请调用 QueryDeviceDetail 或 QueryDevice 查看正确的IoTId值，或用ProductKey与DeviceName组合代替IoTId。
iot.device.InvalidTimeBucket	指定的时间区间不合法。 请根据API文档中描述正确设置参数。 · Asc为0倒序查询时，StartTime必须大于EndTime。 · Asc为1正序查询时，StartTime必须小于EndTime。

错误码	描述
iot.device.InvokeThingServiceFailed	调用设备服务失败。 请检查输入参数是否正确，如Args的参数格式和取值等。
iot.device.LongDevicePropertiesString	入参设备属性长度超过最大值。 请查看相关API文档的限制说明。
iot.device.NoneDeviceNameElement	设备名称列表为空。
iot.device.NoneDeviceProperties	没有有效的设备属性。 请核对传入的属性Identifier是否与TSL中定义的一致。
iot.device.NotExistedDevice	设备不存在。 传入的设备IotId、ProductKey或DeviceName值错误。请调用 QueryDeviceDetail 或 QueryDevice 查看正确值。
iot.device.NullApplyId	创建设备的申请ID（ApplyId）不能为空。
iot.device.NullDeviceName	设备名称不能为空。
iot.device.NullDevicePropertyKey	设备属性名称不能为空。
iot.device.NullDevicePropertiesString	入参设备属性不能为空。
iot.device.QueryDeviceApplyActionError	查询设备申请单信息出错。 请确认入参信息正确，然后重试。
iot.device.QueryDeviceAttrDataHistoryFailed	获取设备属性数据历史记录失败。 请确认入参信息正确，然后重试。
iot.device.QueryDeviceAttrStatusFailed	获取设备属性状态信息失败。 请确认入参信息正确，然后重试。
iot.device.QueryDeviceEventHistoryFailed	获取设备事件调用记录失败。 请确认入参信息正确，然后重试。

错误码	描述
iot.device.QueryDeviceListActionError	查询设备列表失败。 请确认入参信息正确，然后重试。
iot.device.QueryDeviceServiceHistoryFailed	获取设备服务调用记录失败。 请确认入参信息正确，然后重试。
iot.device.QueryDeviceStatisticsFailed	查询设备统计信息失败。 请确认入参信息正确，然后重试。
iot.device.QueryDeviceStatusFailed	查询设备状态信息失败。 请确认入参信息正确，然后重试。
iot.device.QueryTopoRelationFailed	查询拓扑关系失败。 请确认入参信息是否正确。如，传入的 PageSize 值大于限定值50会报此错误。
iot.device.RemoveTopoRelationFailed	移除拓扑关系失败。 请确认入参信息正确，然后重试。
iot.device.SaveOrUpdateDevicePropertiesFailed	新增或者修改设备属性失败。 请确认入参信息正确，然后重试。
iot.device.SetDevicePropertyFailed	设置设备属性失败。 请检查入参Items的参数值和格式是否正确，指定的属性是否是读写型。
iot.device.TooManyDevicePropertiesPerTime	传入的属性个数超过限定值。 请参见相关API文档限制说明。
iot.device.TopoRelationCountExceeded	拓扑关系数量过多。 请参见 使用限制 中网关与子设备数量限制。
iot.device.VerifyDeviceFailed	验证设备失败。 请确认入参信息正确，然后重试。

设备分组（Group）相关错误码

以iot.group开头的错误码为设备分组相关错误码。

错误码	描述
iot.group.NullGroupId	入参分组ID没有赋值。
iot.group.DeleteGroupFailed	删除分组失败。 请确认入参信息正确，然后重试。
iot.group.SubGroupNotNull	分组下有子分组。 当分组下有子分组时，不能删除分组，需先删除自分组。
iot.group.InvalidGroupName	分组名称不合法。 分组名称可包含中文汉字、英文字母、数字和下划线（_）。长度范围 4 - 30 字符（一个中文汉字占二个字符）。
iot.group.GroupNameExisted	分组名称已存在。
iot.group.QueryGroupInfoFailed	查询分组详情失败。 请确认入参信息正确，然后重试。
iot.group.NotExistedGroup	分组不存在。 请确认GroupId值。
iot.group.QueryGroupCountFailed	查询分组数量失败。 请确认入参信息正确，然后重试。
iot.group.QueryGroupListFailed	查询分组列表失败。 请确认入参信息正确，然后重试。
iot.group.BindGroupRelationFailed	绑定分组关系失败。 请确认入参信息正确，然后重试。
iot.group.UpdateGroupFailed	修改分组信息失败。 请确认入参信息正确，然后重试。
iot.group.QueryGroupTreeFailed	获取分组关系结构失败。 请确认入参信息正确，然后重试。
iot.group.CreateGroupFailed	创建分组失败。 请确认入参信息正确，然后重试。

错误码	描述
iot.group.InvalidFormattedTagString	标签格式不合法。 标签数据为JSON格式。由标签的tagKey和tagValue组成，tagKey和tagValue均不能为空。多个标签以英文逗号间隔。如， <code>[{"tagKey":"h1","tagValue":"rr"}, {"tagKey":"7h","tagValue":"rr"}]</code> 。
iot.group.TagCountExceedMax	标签数量超过最大值。 每个分组最多可有100个标签。
iot.group.GroupCountExceedMax	分组数量超过最大值。 · 一个分组最多可包含100个子分组。 · 一个设备最多可以被添加到10个分组中。
iot.group.SetGroupTagFailed	设置分组标签信息失败。 请确认入参信息正确，然后重试。
iot.group.QueryGroupTagFailed	查询分组标签信息失败。 请确认入参信息正确，然后重试。
iot.group.LongGroupDescError	分组描述字段过长。 分组描述长度限制为100字符（一个中文汉字占一个字符）。
iot.group.QueryGroupRelationFailed	查询分组关系失败。 请确认入参信息正确，然后重试。
iot.group.GroupLevelExceedingLimitError	分组层级超过限制。 分组只支持三级嵌套，即分组>子分组>子子分组。

消息相关错误码

以`iot.messagebroker`开头的错误码为消息相关错误码。此类错误码主要出现在调用消息通信相关API、设备影子相关API和规则引擎相关API失败时。（规则引擎相关API调用失败错误码，请见本文档下一章节。）

错误码	描述
iot.messagebroker.CreateTopicRouteFailed	创建Topic之间消息路由失败。 请确认入参信息正确，然后重试。
iot.messagebroker.CreateTopicTemplateException	创建Topic类过程发生异常。 请确认入参信息正确，然后重试。
iot.messagebroker.CreateTopicTemplateFailed	创建Topic类失败。 请确认入参信息正确，然后重试。
iot.messagebroker.DeleteTopicTemplateException	删除Topic类过程发生异常。 请确认入参信息正确，然后重试。
iot.messagebroker.DeleteTopicTemplateFailed	删除Topic类失败。 请确认入参信息正确，然后重试。
iot.messagebroker.DestTopicNameArraySizeIsLarge	同一消息源Topic配置的路由目标Topic数量超过最大限制数。 一个源Topic最多可对应100个目标Topic。
iot.messagebroker.DeleteTopicRouteFailed	删除指定Topic间的路由失败。 请确认入参信息正确，然后重试。
iot.messagebroker.DesireInfoInShadowMessageIsNotJson	设备影子中的desire信息不是JSON格式。
iot.messagebroker.DesireValueIsNullInShadowMessage	设备影子中的desire信息值为空。
iot.messagebroker.ElementKeyOrValueIsNullInDesire	desire信息包含有空的属性标识符或者属性值。
iot.messagebroker.ElementKeyOrValueIsNullInReport	report信息包含有空的属性标识符或者属性值。
iot.messagebroker.HALFCONN	由于设备为半连接状态导致失败。
iot.messagebroker.InvalidFormattedSrcTopicName	消息源Topic名称格式错误。 可在控制台设备详情页的Topic列表下查看设备的Topic。

错误码	描述
iot.messagebroker.InvalidFormattedTopicName	Topic格式错误。 可在控制台设备详情页的Topic列表下查看设备的Topic。
iot.messagebroker.InvalidFormattedTopicTemplateId	Topic类ID格式错误。 可调用 QueryProductTopic 查看TopicId。
iot.messagebroker.InvalidTimeoutValue	超时时间参数设置有误。 请参见相关API文档查看时间设置方法。
iot.messagebroker.InvalidTopicTemplateOperationValue	Topic类的操作权限值错误。操作权限取值： SUB：订阅。 PUB：发布。 ALL：发布和订阅。
iot.messagebroker.InvalidVersionValueInShadowMessage	设备影子中的version值错误。
iot.messagebroker.MethodValueIsNotUpdate	设备影子中的method信息值不是update。
iot.messagebroker.MessageContentIsNotBase64Encode	消息内容没有经过base64编码。
iot.messagebroker.NoneElementInDesire	desire信息中没有属性。
iot.messagebroker.NoneElementInReport	report信息中没有属性。
iot.messagebroker.NoneElementDestTopicNameInArray	目标Topic列表中没有元素。
iot.messagebroker.NotFoundDesireInShadowMessage	设备影子的state信息中没有desire信息。
iot.messagebroker.NotFoundMethodInShadowMessage	设备影子没有method信息。
iot.messagebroker.NotFoundReportInShadowMessage	设备影子中没有report信息。
iot.messagebroker.NotFoundStateInShadowMessage	设备影子中没有state信息。

错误码	描述
iot.messagebroker.NotFoundVersionOrNullVersionValue	缺少version信息或者version值为空。
iot.messagebroker.NotMatchedProductKeyWithSrcTopicOwner	消息源Topic对应的产品ID不属于当前用户。
iot.messagebroker.NullMessageContent	消息内容不能为空。
iot.messagebroker.NullShadowMessage	设备影子内容不能为空。
iot.messagebroker.NullSrcTopicName	消息源Topic名称不能为空。
iot.messagebroker.NullTopicName	Topic不能为空。
iot.messagebroker.NullTopicTemplateId	Topic类ID不能为空。
iot.messagebroker.NullTopicTemplateOperation	Topic类的操作权限不能为空。
iot.messagebroker.OFFLINE	由于设备离线导致失败。
iot.messagebroker.PublishMessageException	发送消息过程出现异常。 请确认入参信息正确，然后重试。
iot.messagebroker.PublishMessageFailed	发送消息失败。 请确认入参信息正确，然后重试。
iot.messagebroker.QueryDeviceShadowActionError	查询设备影子失败。 请确认入参信息正确，然后重试。
iot.messagebroker.QueryProductTopicListActionError	获取Topic类列表失败。 请确认入参信息正确，然后重试。
iot.messageborker.QueryTopicReverseRouteTableListActionError	获取消息反向路由列表（即消息源Topic列表）失败。 请确认入参信息正确，然后重试。
iot.messageborker.QueryTopicRouteTableListActionError	获取消息路由列表失败。 请确认入参信息正确，然后重试。
iot.messagebroker.QueryTopicTemplateActionError	查询Topic类失败。 请确认入参信息正确，然后重试。
iot.messagebroker.QueryTopicTemplateException	获取Topic类过程发生异常。 请确认入参信息正确，然后重试。

错误码	描述
iot.messagebroker.RateLimit	由于限流导致失败。 请参见 使用限制 。
iot.messagebroker.ReportInShadowMessageIsNotJson	设备影子中的state信息中的report信息不是JSON格式。
iot.messagebroker.RrpcException	RRPC发送消息过程出现异常。 请确认入参信息正确，然后重试。
iot.messagebroker.RrpcFailed	RRPC发送消息失败。 请确认入参信息正确，然后重试。
iot.messagebroker.ShadowMessageIsNotJson	设备影子不是JSON格式。
iot.messagebroker.ShadowMessageLengthIsLarge	设备影子的长度超过最大限制。 设备影子文档的大小限制16 KB。
iot.messagebroker.TIMEOUT	由于超时导致失败。
iot.messagebroker.TooManyElementInDesire	desire信息中包含的属性总数超过限定数。 设备影子JSON文档的属性数量限制为128。
iot.messagebroker.TooManyElementInReport	report信息包含的属性总数超过限定最大数。 设备影子JSON文档的属性数量限制为128。
iot.messagebroker.TopicAlreadyFound	同一产品下Topic类名称重复。
iot.messagebroker.TopicTemplateCountExceedMax	产品的Topic类数量超过最大值。 一个产品最多可以定义50个Topic类。
iot.messagebroker.TopicTemplateIsNotFound	Topic类不存在。 可调用 QueryProductTopic 查看产品的Topic类。
iot.messagebroker.UpdateDeviceShadowMessageFailed	更新设备影子失败。 请确认入参信息正确，然后重试。
iot.messagebroker.UpdateTopicTemplateException	更新Topic类过程发生异常。 请确认入参信息正确，然后重试。

错误码	描述
iot.messagebroker.UpdateTopicTemplateFailed	更新Topic类失败。 请确认入参信息正确，然后重试。

规则相关错误码

以iot.rule和iot.ruleng开头的错误码，及少量iot.messagebroker开头的错误码，是规则引擎相关错误码。

提示出现异常或失败时，请确认入参信息正确，然后重试。

错误码	描述
iot.rule.CreateRuleException	创建规则过程发生异常。 请确认入参信息正确，然后重试。
iot.rule.DeleteRuleFailed	删除规则失败。 请确认入参信息正确，然后重试。
iot.rule.IncorrentRuleActionId	规则动作ID错误。 可调用 ListRuleActions 查看规则动作ID。
iot.rule.IncorrentRuleActionType	规则动作类型错误。 规则动作类型参数Type支持可选值： - DATAHUB：DataHub - ONS：消息队列（RocketMQ） - MNS：消息服务 - FC：函数计算 - OTS：表格存储  说明： 数据格式为二进制的规则（即规则的DataType参数是BINARY）不支持转发数据至OTS（表格存储）。 - REUBLISH：另一个物联网平台Topic。
iot.rule.IncorrentRuleId	规则ID错误。

错误码	描述
iot.rule.NullForwardDestForRule	转发数据目的地不能为空。 Configuration中的具体配置方法，请参见 CreateRuleAction 。
iot.rule.NullSqlForRule	规则的SQL语句不能为空。
iot.rule.NotFoundRule	规则不存在。 请输入正确的规则ID（RuleId）。可调用 ListRule 查看账号下所有规则的RuleId。
iot.rule.NotFoundRuleAction	规则动作不存在。 请输入正确的规则动作ID（ActionId）。 可调用 ListRuleActions 查看某个规则下的所有ActionId。
iot.rule.ParseRuleActionConfigError	无法正常解析规则动作的配置。 请确认入参信息正确，然后重试。
iot.rule.QueryRuleActionListError	查询规则动作列表失败。 请确认入参信息正确，然后重试。
iot.rule.QueryRulePageActionError	分页获取规则列表失败。 请确认入参信息正确，然后重试。
iot.rule.RuleActionIsAlreadyCreated	已存在相同的规则动作。
iot.rule.RuleCountExceedMax	规则总数超过最大限制数。 单账号最多可以设置1000条规则。
iot.rule.RuleNameIsAlreadyExisted	规则名称已经存在。
iot.rule.StartRuleFailed	启动规则失败。 请确认入参信息正确，然后重试。
iot.rule.StopRuleFailed	停止规则失败。 请确认入参信息正确，然后重试。
iot.rule.TooManyRuleAction	规则动作数量超过最大限制。 一条规则中转发数据的操作不能超过10个。

错误码	描述
iot.rule.UpdateRuleFailed	更新规则失败。 请确认入参信息正确，然后重试。
iot.ruleng.CreateRuleActionFailed	创建规则动作失败。 请确认入参信息正确，然后重试。
iot.ruleng.DeleteRuleActionFailed	删除规则动作失败。 请确认入参信息正确，然后重试。
iot.ruleng.IncorrectType	应用规则的Topic类型错误。 TopicType支持的可选值： · 0：系统Topic · 1：自定义Topic · 2：设备状态消息Topic
iot.ruleng.IncorrectSysTopic	错误的系统Topic。 可在控制台设备详情页的Topic列表页签下查看正确的Topic。
iot.ruleng.InvalidRamRole	非法的RAM角色。 请登录RAM控制台查看角色信息。
iot.ruleng.QueryRuleActionFailed	获取规则动作失败。 请确认入参信息正确，然后重试。
iot.ruleng.RuleActionConfigurationIsNotJson	规则动作配置不是JSON格式。 参数Configuration的值必须是正确的JSON格式。具体请参见 CreateRuleAction 。
iot.ruleng.RuleAlreadyIsStarted	规则是已启动状态。
iot.ruleng.NullRamRoleArn	roleArn不能为空。
iot.ruleng.NullRamRoleName	roleName不能为空。
iot.ruleng.NullRuleActionConfig	规则动作配置（参数Configuration）不能为空。
iot.ruleng.NullRuleActionType	规则动作类型（参数Type）不能为空。

错误码	描述
iot.messagebroker.IncorrectRuleSql	规则的SQL配置错误。 请根据 CreateRule 说明配置SQL。
iot.messagebroker.QueryRuleConfigActionException	获取规则配置信息过程出现异常。 请确认入参信息正确，然后重试。

以下表格分别列举消息转发目标设置失败的特有错误码。

表 2-1: 目标为REPUBLISH（另一个IoT Topic）的错误码

错误码	描述
iot.messagebroker.InvalidFormattedTopicName	Topic格式错误。 可在控制台设备详情页的Topic列表页签下查看正确的Topic格式。
iot.prod.NotExistedProduct	产品不存在。 请确认输入的ProductKey正确，并该产品属于当前阿里云账号。
iot.common.QueryProductActionError	查询产品失败。 请确认入参信息正确，然后重试。
iot.ruleng.IncorrectSysTopic	系统Topic错误。 可在控制台设备详情页的Topic列表页签下查看正确的Topic。
iot.messagebroker.NullTopicName	Topic名称不能为空。

表 2-2: 目标为DATAHUB（DataHub）的错误码

错误码	描述
iot.ruleng.IncorrectRegionName	regionName值错误。
iot.ruleng.NullProjectOfDatahub	DataHub的projectName不能为空。
iot.ruleng.NullTopicInDatahubProject	DataHub产品下的project中topicName不能为空。

错误码	描述
iot.ruleng.EmptySchemaNameOfTopic	目标DataHub Topic的Schema的名称name值不能为空。
iot.ruleng.EmptySchemaTypeOfTopic	目标DataHub Topic的Schema的类型type值不能为空。
iot.ruleng.EmptySchemaValueOfTopic	目标DataHub Topic的Schema值value不能为空。
iot.ruleng.NullOrEmptySchemaOfTopic	目标DataHub Topic的Schema不能为空。
iot.ruleng.NotFoundProjectInDataHub	DataHub中不存在此项目（project）。 请在DataHub中确认项目名称是否正确。
iot.ruleng.IncorrectSchemaValueOfTopic	目标DataHub Topic的Schema值错误。

表 2-3: 目标为OTS（表格存储）的错误码

错误码	描述
iot.ruleng.NullOtsInstanceName	表格存储的实例名称不能为空。
iot.ruleng.NullTableNameInOtsInstance	表格存储中实例的表名不能为空。
iot.ruleng.NullPrimaryKeyInOtsTable	表格存储中表的主键不能为空。
iot.ruleng.NullPrimaryKeyNameInOts	主键的名称不能为空。
iot.ruleng.NullPrimaryKeyValueInOts	主键的值不能为空。
iot.ruleng.IncorrectPrimaryKeyValueInOtsTable	表格存储中主键值错误。 请在表格存储中，查看您创建数据表时定义的主键。

表 2-4: 目标为MNS（消息服务）的错误码

错误码	描述
iot.ruleng.NullTopicNameInMns	消息服务中的主题不能为空。
iot.ruleng.NotFoundTopicInMns	消息服务中不存在此主题。 请在消息服务中，确认主题（Topic）名称。
iot.ruleng.QueryMnsTopicListActionError	获取消息服务主题列表失败。 请确认入参信息正确，然后重试。

表 2-5: 目标为FC（函数计算）的错误码

错误码	描述
iot.ruleng.NullServiceNameInFc	函数计算服务名称为空。
iot.ruleng.NullFunctionNameInFc	函数计算函数名称为空。
iot.ruleng.NotFoundServiceInFc	函数计算服务不存在。 请在函数计算中，确认正确的服务名称。

表 2-6: 目标为ONS（消息队列）的错误码

错误码	描述
iot.messagebroker.NullTopicName	消息队列中接收消息的Topic不能为空。

2.7 产品管理

2.7.1 CreateProduct

调用该接口新建产品。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：CreateProduct。
ProductName	String	是	为新建产品命名。产品名应满足以下限制：由4-30位中文、英文字母、数字和下划线（_）组成（一个中文字符占两位）。  说明： 产品名在当前账号下应保持唯一。

名称	类型	是否必需	描述
NodeType	Integer	是	产品的节点类型，取值： 0：设备。设备不能挂载子设备。可以直连物联网平台，也可以作为网关的子设备连接物联网平台。 1：网关。网关可以挂载子设备，具有子设备管理模块，维持子设备的拓扑关系，并将拓扑关系同步到物联网平台。
AliyunCommodityCode	String	否	产品版本类型。 · <code>iothub_senior</code>：高级版。 · <code>iothub</code>：基础版。 若不传入此参数默，则默认为基础版。
DataFormat	Integer	是	产品版本类型选择为<code>iothub_senior</code>的产品数据格式。 此参数为高级版产品的特有参数，并且是创建高级版产品的必需参数。 可选值： · 0：透传/自定义格式（<code>CUSTOM_FORMAT</code>）。 · 1：Alink协议（<code>ALINK_FORMAT</code>）。
Description	String	否	为新建产品添加描述信息。描述信息应在100字符以内。
Id2	Boolean	否	是否使用ID²认证。 可选值： · <code>true</code>：开通ID²认证。 · <code>false</code>：不开通ID²认证。 不传入此参数，则默认为不开通。

名称	类型	是否必需	描述
ProtocolType	String	否	设备接入网关的协议类型。 此参数为创建高级版产品，且产品节点类型为要接入网关的设备时的特有参数。 可选值： • modbus: Modbus协议。 • opc-ua: OPC UA协议。 • customize: 自定义协议。 • ble: BLE协议。 • zigbee: ZigBee协议。
NetType	String	否	联网方式。 此参数为创建高级版产品，产品节点类型为网关或不接入网关的设备时的特有参数。 可选值： • WIFI: WiFi • CELLULAR: 蜂窝网 • ETHERNET: 以太网 • LORA: LoRaWAN • OTHER: 其他 若不传入此参数，则默认为WiFi。
JoinPermissionId	String	否	LoRaWAN入网凭证ID。连网方式NetType选择为LORA时，该参数必需。 请调用QueryLoRaJoinPermissions查询您账号下的LoRaWAN入网凭证的JoinPermissionId。 如果您还没有LoRaWAN入网凭证，请访问物联网络管理平台创建。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见本文错误码章节。
ProductKey	String	产品的Key。
Data	Data	调用成功时返回的新建产品信息。详情参见下表ProductInfo。

表 2-7: ProductInfo

名称	类型	描述
ProductName	String	产品的名称。
ProductKey	String	物联网平台为新建产品颁发的产品Key，作为该产品 的全局唯一标识。  说明： 请妥善保管新建产品的ProductKey。在其他操作中 会用到该信息。
Description	String	产品描述信息。
DataFormat	Integer	产品类型数据格式。 · 0：透传/自定义格式（CUSTOM_FORMAT）。 · 1：Alink协议（ALINK_FORMAT）。 此参数为高级版产品的特有参数。
AliyunCommodityCode	String	产品类型。 · iothub_senior：高级版。 · iothub：基础版。
ProtocolType	String	设备接入网关协议类型。 此参数为高级版产品的特有参数。

名称	类型	描述
NodeType	Integer	产品的节点类型，取值： 0：设备。设备不能挂载子设备。可以直连物联网平台，也可以作为网关的子设备连接物联网平台。 1：网关。网关可以挂载子设备，具有子设备管理模块，维持子设备的拓扑关系，和将拓扑关系同步到物联网平台。 此参数为高级版产品的特有参数。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=CreateProduct
&AliyunCommodityCode=iothub_senior
&DataFormat=1
&Description=Product test
&NodeType=0
&ProductName=Test
&ProtocolType=modbus
&公共请求参数
```

返回示例

· JSON格式

```
{
  "Data": {
    "Description": "Product test",
    "DataFormat": 1,
    "ProtocolType": "modbus",
    "ProductKey": "a1FlqIQ****",
    "NodeType": 0,
    "ProductName": "Test",
    "AliyunCommodityCode": "iothub_senior"
  },
  "ProductKey": "a1FlqIQ****",
  "RequestId": "E55E50B7-40EE-4B6B-8BBE-D3ED55CCF565",
  "Success": true
}
```

```
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<CreateProductResponse>
  <Data>
    <Description>Product test</Description>
    <DataFormat>1</DataFormat>
    <ProtocolType>modbus</ProtocolType>
    <ProductKey>a1FlqIQ****</ProductKey>
    <NodeType>0</NodeType>
    <ProductName>Test</ProductName>
    <AliyunCommodityCode>iothub_senior</AliyunCommodityCode>
  </Data>
  <ProductKey>a1FlqIQ****</ProductKey>
  <RequestId>E55E50B7-40EE-4B6B-8BBE-D3ED55CCF565</RequestId>
  <Success>true</Success>
</CreateProductResponse>
```

2.7.2 QueryProduct

调用该接口查询指定产品的详细信息。

限制说明

该接口的调用有限流，不得超过50次/秒。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryProduct。
ProductKey	String	是	要查询的产品的ProductKey。ProductKey是物联网平台为新建产品颁发的产品Key，作为其全局唯一标识符。您可以在创建产品的返回结果中查看该信息。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

名称	类型	描述
Data	Data	调用成功时，返回的数据。详情见表格Data。

表 2-8: Data

名称	类型	描述
GmtCreate	Long	产品创建时间。
DataFormat	Integer	高级版产品的数据类型，指设备与云端之间的数据通信协议类型。取值： 0：透传模式。使用自定义的串口数据格式。该模式下，设备可以上报原始数据（如二进制数据流）。阿里云IoT平台会运行您配置在云端的数据解析脚本，将原始数据转换成Alink JSON标准数据格式。 1：Alink JSON。阿里云IoT平台定义的设备与云端的数据交换协议，采用JSON格式。  说明： 此参数为高级版产品特有参数。
Description	String	产品的描述信息。
DeviceCount	Integer	该产品下的设备数量。
NodeType	Integer	高级版产品的节点类型。取值： 0：设备。设备不能挂载子设备，可以直连IoT Hub，也可以作为网关的子设备连接IoT Hub。 1：网关。网关可以挂载子设备，具有子设备管理模块，维持子设备的拓扑关系，并且可以将拓扑关系同步到云端。
ProductKey	String	产品Key。新建产品时，IoT平台为该产品颁发的全局唯一标识。
ProductName	String	产品名称。
ProductSecret	String	产品密钥。
CategoryName	String	高级版产品的设备类型。取值为您在创建高级版产品时，所选择的设备类型。

名称	类型	描述
CategoryKey	String	高级版产品的设备类型的英文标识符。
AliyunCommodityCode	String	取值： • iothub：物联网平台基础版。 • iothub_senior：物联网平台高级版。
Id2	Boolean	该产品是否使用ID ² 认证。取值： • true：使用ID²认证。 • false：不使用ID²认证。
ProductStatus	String	产品状态。 • DEVELOPMENT_STATUS：开发中。 • RELEASE_STATUS：产品已发布。
NetType	Integer	联网方式。取值： • 3：WiFi • 6：Cellular (2G/3G/4G)蜂窝网 • 7：Ethernet 以太网。 • 8：其他

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryProduct
&ProductKey=a1*****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "Data": {
    "Owner": true,
    "DataFormat": 1,
    "CategoryName": "",
    "ProductStatus": "DEVELOPMENT_STATUS",
    "ProductKey": "a1*****",
    "GmtCreate": 1530117156000,
    "NodeType": 0,
    "ProductSecret": "i6YZucqlp6*****",
    "ProductName": "protest",
    "NetType": 3,
    "DeviceCount": 1,
    "Id2": false,
    "AliyunCommodityCode": "iothub_senior"
  },
}
```

```
"RequestId": "E4F94B97-1D64-4080-BFD2-67461667AA43",
"Success": true
}
```

- XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryProductResponse>
  <Data>
    <Owner>true</Owner>
    <DataFormat>1</DataFormat>
    <CategoryName></CategoryName>
    <ProductStatus>DEVELOPMENT_STATUS</ProductStatus>
    <ProductKey>a1*****</ProductKey>
    <GmtCreate>1530117156000</GmtCreate>
    <NodeType>0</NodeType>
    <ProductSecret>i6YZucqlp6*****</ProductSecret>
    <ProductName>protest</ProductName>
    <NetType>3</NetType>
    <DeviceCount>1</DeviceCount>
    <Id2>false</Id2>
    <AliyunCommodityCode>iotHub_senior</AliyunCommodityCode>
  </Data>
  <RequestId>E4F94B97-1D64-4080-BFD2-67461667AA43</RequestId>
  <Success>true</Success>
</QueryProductResponse>
```

2.7.3 QueryProductList

调用该接口查看所有产品列表。

接口说明

该接口可用于查看当前账号下所有的产品，也可以指定只查询基础版或高级版产品。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryProductList。
CurrentPage	Integer	是	指定显示返回结果中的第几页。
PageSize	Integer	是	指定返回结果中每页显示的产品数量，最大值是200。

名称	类型	是否必需	描述
AliyunCommodityCode	String	否	指定要查看的产品类型，取值： iothub_senior：物联网平台高级版 iothub：物联网平台基础版  说明： 如果不传入该参数，则返回所有产品的列表。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的产品列表。详情参见下表Data。

表 2-9: Data

名称	类型	描述
PageSize	Integer	每页显示的产品数。
PageCount	Integer	总页数。
CurrentPage	Integer	当前页号。
Total	Integer	当前账号下的产品总数。
List	List	产品信息列表。请参见下表ProductInfo。  说明： 返回的产品信息按照产品创建时间倒序排列。

表 2-10: ProductInfo

名称	类型	描述
DataFormat	Integer	数据格式，指设备与云端之间的数据通信协议类型，取值： 0：透传模式。使用自定义的串口数据格式。该模式下，设备可以上报原始数据（如二进制数据流），阿里云IoT平台会运行您配置在云端的数据解析脚本，将原始数据转换成Alink JSON标准数据格式。 1：Alink JSON。阿里云物联网平台定义的设备与云端的数据交换协议，采用JSON格式。  说明： 此参数为高级版产品特有参数。
ProductKey	String	产品的Key。新建产品时，物联网平台为该产品颁发的全局唯一标识。
NodeType	Integer	产品的节点类型，取值： 0：设备。设备不能挂载子设备，可以直连物联网平台，也可以作为网关的子设备连接物联网平台。 1：网关。网关可以挂载子设备，具有子设备管理模块，维持子设备的拓扑关系，并且可以将拓扑关系同步到云端。
ProductName	String	产品名称。
DeviceCount	Integer	产品下的设备数量。
GmtCreate	Long	产品的创建时间。
Description	String	产品的描述信息。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryProductList
&CurrentPage=1
&PageSize=10
```

&公共请求参数

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "PageCount": 1,
    "PageSize": 10,
    "List": {
      "ProductInfo": [
        {
          "DataFormat": 0,
          "ProductKey": "a1*****",
          "NodeType": 0,
          "ProductName": "test123123123",
          "DeviceCount": 4,
          "GmtCreate": 1517479077000,
          "Description": "描述信息"
        },
        {
          "DataFormat": 1,
          "ProductKey": "a1*****",
          "NodeType": 0,
          "ProductName": "测试事件",
          "DeviceCount": 0,
          "GmtCreate": 1517466635000,
          "Description": "描述信息"
        }
      ]
    },
    "CurrentPage": 1,
    "Total": 2
  }
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<QueryProductListResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <PageCount>1</PageCount>
    <PageSize>10</PageSize>
    <List>
      <ProductInfo>
        <DataFormat>0</DataFormat>
        <ProductKey>a1*****</ProductKey>
        <NodeType>0</NodeType>
        <ProductName>test123123123</ProductName>
        <DeviceCount>4</DeviceCount>
        <GmtCreate>1517479077000</GmtCreate>
        <Description>描述信息</Description>
      </ProductInfo>
      <ProductInfo>
        <DataFormat>1</DataFormat>
        <ProductKey>a1*****</ProductKey>
        <NodeType>0</NodeType>
```

```
<ProductName>测试事件</ProductName>
<DeviceCount>0</DeviceCount>
<GmtCreate>1517466635000</GmtCreate>
<Description>描述信息</Description>
</ProductInfo>
</List>
<CurrentPage>1</CurrentPage>
<Total>2</Total>
</Data>
</QueryProductListResponse>
```

2.7.4 UpdateProduct

调用该接口修改指定产品的信息。

限制说明

该接口的调用有限流，不得超过50次/秒。

请求参数

参数	类型	是否必需	描述
Action	String	是	要执行的操作，取值：UpdateProduct。
ProductKey	String	是	要修改信息的产品的Key。
ProductName	String	是	修改后的产品名称。产品名应满足以下限制：长度为4-30字符，可以包含中文、英文字母、数字和下划线（一个中文字符占两位）。  说明： 产品名在当前账号下应具有唯一性。
Description	String	否	指定修改后的产品描述信息。描述信息应在100字符以内。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=UpdateProduct
&ProductKey=a1*****
&ProductName=TestProductNew
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "C4FDA54C-4201-487F-92E9-022F42387458",
  "Success": true,
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateProductResponse>
  <RequestId>C4FDA54C-4201-487F-92E9-022F42387458</RequestId>
  <Success>true</Success>
</UpdateProductResponse>
```

2.7.5 DeleteProduct

调用该接口删除指定产品。

接口说明

如果您不再需要某个产品，您可以将其删除。产品删除后，产品Key（ProductKey）将失效，与产品关联的其他信息也一并删除，您将无法执行与该产品关联的任何操作。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteProduct。
ProductKey	String	是	要删除的产品Key。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。

名称	类型	描述
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteProduct
&ProductKey=a1QDKAU****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<DeleteDeviceResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</DeleteDeviceResponse>
```

2.7.6 CreateProductTags

调用该接口为指定产品创建标签。

限制说明

- 单次调用该接口最多能为指定产品创建10个标签。
- 单个产品的标签总数不超过100个。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值CreateProductTags。
ProductKey	String	是	产品Key，物联网平台为产品颁发的唯一标识。

名称	类型	是否必需	描述
ProductTags	List<String>	是	要创建的标签。标签包括TagKey和TagValue，分别对应标签的key和value。请参见下表 ProductTag 。  说明： 新增标签的TagKey不能重复，也不能和已有标签的key重复。
公共参数	-	是	请参见 公共参数 。

表 2-11: ProductTag

名称	类型	是否必需	描述
TagKey	String	是	产品标签键(key)。可包含英文大小写字母，数字和点号(.)，长度不可超过30个字符。
TagValue	String	是	产品标签值(value)。可包含中文、英文字母、数字、下划线(_)和连字符(-)。长度不可超过128字符。一个中文汉字算2字符。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
InvalidProductTags	List<ProductTag>	调用失败时，返回不合法的产品标签列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=CreateProductTags
&ProductKey=a1h7knJdld1
&ProductTag.1.TagKey=first
&ProductTag.1.TagValue=value1
&ProductTag.2.TagKey=second
&ProductTag.2.TagValue=value2
```

&公共请求参数

返回示例

· JSON格式

```
{
  "RequestId": "354A4F9B-6B01-4498-8084-867F59720BA5",
  "Success": true
}
```

· XML格式

```
<CreateProductTagsResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</CreateProductTagsResponse>
```

2.7.7 UpdateProductTags

调用该接口更新产品标签。

限制说明

单次调用该接口最多可更新10个标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值UpdateProductTags。
ProductKey	String	是	产品Key，物联网平台为产品颁发的唯一标识。
ProductTags	List<String>	是	要更新的标签。标签包括TagKey和TagValue，分别对应标签的key和value。请参见下表 ProductTag 。
公共请求参数	-	是	请参见 公共参数 。

表 2-12: ProductTag

名称	类型	是否必需	描述
TagKey	String	是	产品标签键(key)。可包含英文大小写字母，数字和点号(.)，长度不能超过30个字符。  说明： - 传入标签的TagKey不能重复。 - 传入标签的TagKey必须是已存在的标签key。

名称	类型	是否必需	描述
TagValue	String	是	产品标签值(value)。可包含中文、英文字母、数字、下划线(_)和连字符(-)。长度不可超过128字符。一个中文汉字算2字符。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
InvalidProductTags	List<String>	调用失败时，返回不合法的产品标签列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=UpdateProductTags
&ProductKey=a1h7knJdld1
&ProductTag.1.TagKey=first
&ProductTag.1.TagValue=value1
&ProductTag.2.TagKey=second
&ProductTag.2.TagValue=value2
&公共请求参数
```

返回示例

JSON格式

```
{
  "RequestId": "92586B4B-FF78-494A-A22C-368E4293FBB7",
  "Success": true
}
```

XML格式

```
<UpdateProductTagsResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
```



```
</UpdateProductTagsResponse>
```

2.7.8 DeleteProductTags

调用该接口删除产品标签。

限制说明

单次调用该接口最多可删除10个标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值DeleteProductTags。
ProductKey	String	是	产品Key，物联网平台为产品颁发的唯一标识。
ProductTag Keys	List<String>	是	产品标签键列表。
公共参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=DeleteProductTags
&ProductKey=a1h7knJdld1
&ProductTagKey.1=first
&ProductTagKey.2=second
&公共请求参数
```

返回示例

JSON格式

```
{
  "RequestId": "E7E8456E-EDD7-41D3-83B1-62FF4F5ED6BD",
  "Success": true
}
```

```
}
```

XML格式

```
<DeleteProductTagsResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</DeleteProductTagsResponse>
```

2.7.9 ListProductTags

调用该接口查询指定产品的所有标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值ListProductTags。
ProductKey	String	是	产品Key，物联网平台为产品颁发的唯一标识。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	List<String>	调用成功时，返回产品标签信息列表。详情见下表 ProductTag 。

表 2-13: ProductTag

名称	类型	是否必需	描述
TagKey	String	是	产品标签键(key)。
TagValue	String	是	产品标签值(value)。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=ListProductTags
```

```
&ProductKey=a1h7knJdld1  
&公共请求参数
```

返回示例

JSON格式

```
{  
  "Data": {  
    "ProductTag": [  
      {  
        "TagValue": "alterTable",  
        "TagKey": "binary"  
      },  
      {  
        "TagValue": "json2",  
        "TagKey": "extt"  
      },  
      {  
        "TagValue": "1234",  
        "TagKey": "Lock"  
      },  
      {  
        "TagValue": "support",  
        "TagKey": "Lockk"  
      },  
      {  
        "TagValue": "reen",  
        "TagKey": "Reen"  
      },  
      {  
        "TagValue": "try",  
        "TagKey": "Reenn"  
      },  
      {  
        "TagValue": "DropTable",  
        "TagKey": "roc"  
      }  
    ]  
  },  
  "RequestId": "7FBE60F8-4AB5-4A8C-AFCB-F4F38851F01F",  
  "Success": true  
}
```

XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>  
<ListProductTagsResponse>  
  <Data>  
    <ProductTag>  
      <TagValue>alterTable</TagValue>  
      <TagKey>binary</TagKey>  
    </ProductTag>  
    <ProductTag>  
      <TagValue>json2</TagValue>  
      <TagKey>extt</TagKey>  
    </ProductTag>  
    <ProductTag>  
      <TagValue>1234</TagValue>  
      <TagKey>Lock</TagKey>  
    </ProductTag>
```

```

<ProductTag>
  <TagValue>support</TagValue>
  <TagKey>Lockk</TagKey>
</ProductTag>
<ProductTag>
  <TagValue>reen</TagValue>
  <TagKey>Reen</TagKey>
</ProductTag>
<ProductTag>
  <TagValue>try</TagValue>
  <TagKey>Reenn</TagKey>
</ProductTag>
<ProductTag>
  <TagValue>DropTable</TagValue>
  <TagKey>roc</TagKey>
</ProductTag>
</Data>
  <RequestId>7FBE60F8-4AB5-4A8C-AFCB-F4F38851F01F</RequestId>
  <Success>true</Success>
</ListProductTagsResponse>

```

2.7.10 ListProductByTags

调用该接口根据标签分页查询产品列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值ListProductByTags。
ProductTags	List<String>	是	产品标签。ProductTag包括TagKey和TagValue，分别对应标签的key和value。请参见下表ProductTag。 - 支持按照TagKey和TagValue组合来搜索。 - 支持只按照TagKey来搜索。 - 传入多个ProductTag是或的关系。
PageSize	Integer	否	指定返回结果中每页显示的记录数量。最大值是50。默认值是10。
CurrentPage	Integer	否	指定显示返回结果中的第几页。默认值为1。
公共请求参数	-	是	请参见 公共参数 。

表 2-14: ProductTag

名称	类型	是否必需	描述
TagKey	String	是	产品标签键(key)。
TagValue	String	否	产品标签值(value)。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
ProductInfos	List<String>	调用成功时，返回产品信息列表。详情见下表 ProductInfo  说明： 返回的产品信息按照产品创建时间倒序排列。

表 2-15: ProductInfo

名称	类型	描述
ProductKey	String	产品Key。
ProductName	String	产品名称。
NodeType	Integer	节点类型。
CreateTime	Long	产品创建时间。
Description	String	产品描述。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=ListProductByTags
&CurrentPage=1
&PageSize=10
&ProductTag.1.TagKey=Reen
&ProductTag.1.TagValue=reem
&ProductTag.2.TagKey=Lock
&ProductTag.2.TagValue=1234
&公共请求参数
```

返回示例

JSON格式

```
{
  "ProductInfos": {
    "ProductInfo": [
      {
```

```
        "Description": "Bulbs in the rooms",
        "ProductKey": "a1h7knJdld1",
        "NodeType": 0,
        "CreateTime": 1545355537000,
        "ProductB": "Bulbs"
    }
  ],
  "RequestId": "2E410BE3-C688-487B-9BF1-F04B33632CCC",
  "Success": true
}
```

XML格式

```
<ListProductByTagsResponse>
  <ProductInfos>
    <ProductInfo>
      <Description>Bulbs in the rooms</Description>
      <ProductKey>a1h7knJdld1</ProductKey>
      <NodeType>0</NodeType>
      <CreateTime>1545355537000</CreateTime>
      <ProductName>Bulbs</ProductName>
    </ProductInfo>
  </ProductInfos>
  <RequestId>2E410BE3-C688-487B-9BF1-F04B33632CCC</RequestId>
  <Success>true</Success>
</ListProductByTagsResponse>
```

2.8 设备管理

2.8.1 RegisterDevice

调用该接口在指定产品下注册设备。

接口说明

注册设备指在物联网平台产品下添加设备。在指定产品下成功注册设备后，阿里云物联网平台为设备颁发全局唯一的设备ID（`IotId`），用来标识该设备。在进行与设备相关的操作时，您可能需要提供目标设备的`IotId`。



说明:

除了`IotId`，您也可以使用`ProductKey`和`DeviceName`组合来标识一个设备。其中`ProductKey`是新建产品时物联网平台为产品颁发的产品Key，`DeviceName`是注册设备时由您指定或由系统随机生成的设备名称。`IotId`的优先级高于`ProductKey`和`DeviceName`组合。

- 如果您希望在一个产品下批量注册多个设备，并且随机生成设备名，您可以调用[BatchRegisterDevice](#)接口。

- 如果您希望在一个产品下批量注册多个设备，并且为每个设备单独命名，您需要先调用[BatchCheckDeviceNames](#)接口为每个设备命名，并生成申请批次ID（ApplyId），再调用[BatchRegisterDeviceWithApplyId](#)接口批量注册设备。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：RegisterDevice。
ProductKey	String	是	待注册设备所隶属的产品的Key，即产品的唯一标识符。
DeviceName	String	否	为待注册的设备命名。设备名称长度为4-32个字符，可以包含英文字母、数字和特殊字符：连字符（-）、下划线（_）、at符号（@）、点号（.）、和英文冒号（:）。 DeviceName通常与ProductKey组合使用，用作标识具体的设备。  说明： 如果不传入该参数，则由系统随机生成设备名称。
Nickname	String	否	为待注册的设备设置备注名称。备注名称长度为4-32个字符，可包含中文汉字、英文字母、数字和下划线（_）。一个中文汉字算2字符。  说明： 如果不传入该参数，系统不会为设备生成备注名称
DevEui	String	否	LoRaWAN设备的DevEUI。 创建LoRaWAN设备时，该参数必传。
PinCode	String	否	LoRaWAN设备的PIN Code，用于校验DevEUI的合法性。 创建LoRaWAN设备时，该参数必传。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见本文错误码章节。
Data	Data	调用成功时，返回注册的设备信息。详情参见下表DeviceInfo。

表 2-16: DeviceInfo

名称	类型	描述
ProductKey	String	设备隶属的产品Key。
DeviceName	String	设备名称。  说明： 请妥善保管该信息。
DeviceSecret	String	设备密钥。  说明： 请妥善保管该信息。
IotId	String	物联网平台为该设备颁发的设备ID，作为该设备的唯一标识符。  说明： 请妥善保管该信息。
Nickname	String	设备的备注名称。 若您没有为该设备设置备注名称，则该参数返回为空。
DevEui	String	LoRaWAN设备的DevEUI。仅LoRaWAN设备才会返回该参数。
JoinEui	String	LoRaWAN设备的入网凭证 JoinEUI。仅LoRaWAN设备才会返回该参数。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=RegisterDevice
&ProductKey=a1rYuVF****
&DeviceName=device1
&Nickname=detectors_in_beijing
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "DeviceName": "device1",
    "ProductKey": "a1rYuVF****",
    "DeviceSecret": "tXHf4ezGEHcwdyMwoCDHGBmk9avi****",
    "IotId": "CqXL5h5ysRTA4NxjABjj0010fa****",
    "Nickname": "detectors_in_beijing"
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<RegisterDeviceResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <DeviceName>device1</DeviceName>
    <ProductKey>a1rYuVF****</ProductKey>
    <DeviceSecret>tXHf4ezGEHcwdyMwoCDHGBmk9avi****</DeviceSecret>
  >
    <IotId>CqXL5h5ysRTA4NxjABjj0010fa****</IotId>
    <Nickname>detectors_in_beijing</Nickname>
  </Data>
</RegisterDeviceResponse>
```

2.8.2 QueryDeviceDetail

调用该接口查询指定设备的详细信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceDetail。 。

名称	类型	是否必需	描述
IotId	String	否	要查询的设备ID。  说明: 如果传入该参数, 则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符, 和ProductKey与DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合, 则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明: 如果传入该参数, 需同时传入DeviceName。
DeviceName	String	否	指定要查询的设备的名称。  说明: 如果传入该参数, 需同时传入ProductKey。
公共请求参数	-	是	公共请求参数, 请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功, false表示调用失败。
ErrorMessage	String	调用失败时, 返回的出错信息。
Code	String	调用失败时, 返回的错误码。错误码详情, 请参见 错误码 。
Data	Data	调用成功时, 返回设备的详细信息。详情请参见下表Data。

表 2-17: Data

名称	类型	描述
ProductKey	String	设备隶属的产品Key。
ProductName	String	设备隶属的产品名称。
DeviceName	String	设备名称。
Nickname	String	设备的备注名称。
DeviceSecret	String	设备密钥。
IotId	String	IoT平台为该设备颁发的ID，作为该设备的唯一标识符。
UtcCreate	String	设备的创建时间（UTC）。
GmtCreate	String	设备的创建时间（GMT）。
UtcActive	String	设备的激活时间（UTC）。
GmtActive	String	设备的激活时间（GMT）。
UtcOnline	String	设备最近一次上线的时间（UTC）。
GmtOnline	String	设备最近一次上线的时间（GMT）。
Status	String	设备状态。取值： ONLINE：设备在线。 OFFLINE：设备离线。 UNACTIVE：设备未激活。 DISABLE：设备已禁用。
FirmwareVersion	String	设备的固件版本号。
IpAddress	String	设备的IP地址。
NodeType	Integer	节点类型，取值： 0：设备。设备不能挂载子设备，可以直连IoT Hub，也可以作为网关的子设备连接IoT Hub。 1：网关。网关可以挂载子设备，具有子设备管理模块，维持子设备的拓扑关系，并且可以将拓扑关系同步到云端。

名称	类型	描述
Region	String	设备所在地区（与控制台上的物联网地平台服务地域对应）。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceDetail
&ProductKey=a1rYuVF****
&DeviceName=device1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "DeviceName": "device1",
    "GmtActive": "2018-08-06 10:48:41",
    "UtcActive": "2018-08-06T02:48:41.000Z",
    "ProductKey": "a1rYuVF****",
    "DeviceSecret": "CPwUjMUgzdvaZv56TMy6773V3v3****",
    "Nickname": "detectors_in_beijing",
    "GmtCreate": "2018-08-06 10:47:50",
    "UtcCreate": "2018-08-06T02:47:50.000Z",
    "IotId": "SR8FiTu1R9tlUR2V1bmi00105****",
    "Status": "ONLINE",
    "Region": "cn-shanghai",
    "NodeType": 0,
    "GmtOnline": "2018-08-06 13:43:12",
    "UtcOnline": "2018-08-06T05:43:12.000Z",
    "ProductName": "test",
    "IpAddress": "10.0.0.1",
    "FirmwareVersion": "V1.0"
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<QueryDeviceDetailResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <DeviceName>device1</DeviceName>
    <GmtActive>2018-08-06 10:48:41</GmtActive>
    <UtcActive>2018-08-06T02:48:41.000Z</UtcActive>
    <ProductKey>a1rYuVF****</ProductKey>
    <DeviceSecret>CPwUjMUgzdvaZv56TMy6773V3v3****</DeviceSecret>
    <Nickname>detectors_in_beijing</Nickname>
    <GmtCreate>2018-08-06 10:47:50</GmCreate>
    <UtcCreate>2018-08-06T02:47:50.000Z</UtcCreate>
    <IotId>SR8FiTu1R9tlUR2V1bmi0010a5****</IotId>
    <Status>ONLINE</Status>
  </Data>
</QueryDeviceDetailResponse>
```

```

    <Region>cn-shanghai</Region>
    <NodeType>0</NodeType>
    <GmtOnline>2018-08-06 13:43:2</GmtOnline>
    <UtcOnline>2018-08-06T05:43:12.000Z</UtcOnline>
    <ProductName>test</ProductName>
    <IpAddress>10.0.0.1</IpAddress>
    <FirmwareVersion>V1.0</FirmwareVersion>
  </Data>
</QueryDeviceDetailResponse>

```

2.8.3 QueryDevice

调用该接口查询指定产品下的所有设备列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDevice。
ProductKey	String	是	要查询的设备所隶属的产品Key。
PageSize	Integer	否	指定返回结果中每页显示的记录数量，最大值是50。默认值是10。
CurrentPage	Integer	否	指定显示返回结果中的第几页的内容。默认值是1。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
PageCount	Integer	总页数。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页面号。
Total	Integer	总记录数。

名称	类型	描述
Data	Data	调用成功时，返回设备信息列表。详情参见下表DeviceInfo。  说明： 返回的设备信息按照设备创建时间倒序排列。

表 2-18: DeviceInfo

名称	类型	描述
DeviceId	String	设备ID（旧版参数）。  说明： 该参数是旧版本遗留参数，已无实际作用，不能用来标识设备。目前，有效的设备标识符为IotId和ProductKey 与DeviceName组合。
DeviceName	String	设备名称。
ProductKey	String	设备所属产品Key，即物联网平台为该产品颁发的唯一标识。
DeviceSecret	String	设备密钥。
GmtCreate	String	设备创建GMT时间。
GmtModified	String	设备信息修改GMT时间。
UtcCreate	String	设备创建UTC时间。
UtcModified	String	设备信息修改UTC时间。
DeviceStatus	String	设备状态。
IotId	String	设备ID，物联网平台为设备颁发的唯一标识。
Nickname	String	设备的备注名称。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDevice
&ProductKey=aldafD****
&PageSize=10
&CurrentPage=1
&公共请求参数数
```

返回示例

· JSON格式

```
{
  "PageCount": 1,
  "Data": {
    "DeviceInfo": [
      {
        "DeviceId": "Av8NGHGtwPrH9BYG****",
        "DeviceName": "Av8NGHGtwPrH9BYGLMBi",
        "ProductKey": "aldafDE****",
        "DeviceSecret": "d7GYhf5hfcPHDe1bXSd3n9Mj01G3****",
        "UtcModified": "2019-02-20T02:16:09.000Z",
        "GmtCreate": "Wed, 20-Feb-2019 02:16:09 GMT",
        "UtcCreate": "2019-02-20T02:16:09.000Z",
        "GmtModified": "Wed, 20-Feb-2019 02:16:09 GMT",
        "IotId": "Av8NGHGtwPrH9BYGLMBi00****",
        "DeviceStatus": "UNACTIVE",
        "Nickname": "SensorInShanghai"
      },
      {
        "DeviceId": "zNlcSmWQ9BPJlkmkj****",
        "DeviceName": "zNlcSmWQ9BPJlkmkj3H1",
        "ProductKey": "aldafDE****",
        "DeviceSecret": "C27XXmC18yLIEDXvUj6FSlvg07ag****",
        "UtcModified": "2019-02-20T02:16:09.000Z",
        "GmtCreate": "Wed, 20-Feb-2019 02:16:09 GMT",
        "UtcCreate": "2019-02-20T02:16:09.000Z",
        "GmtModified": "Wed, 20-Feb-2019 02:16:09 GMT",
        "IotId": "zNlcSmWQ9BPJlkmkj3H100****",
        "DeviceStatus": "UNACTIVE",
        "Nickname": "DriverInShanghai"
      }
    ]
  },
  "Page": 1,
  "PageSize": 10,
  "RequestId": "CD9E5F99-A095-4A05-9256-D924EA3075E8",
  "Success": true,
  "Total": 2
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryDeviceResponse>
  <PageCount>1</PageCount>
  <Data>
    <DeviceInfo>
      <DeviceId>VIjUsKaJsdWqt6nl****</DeviceId>
      <DeviceName>led_device</DeviceName>
      <ProductKey>aldafDE****</ProductKey>
      <DeviceSecret>qQP6xARs7ky1HuM2reh7y6M4BTdfe****</DeviceSecret>
      <UtcModified>2019-02-19T10:20:14.000Z</UtcModified>
      <GmtCreate>Wed, 30-Jan-2019 11:32:39 GMT</GmtCreate>
      <UtcCreate>2019-01-30T11:32:39.000Z</UtcCreate>
      <GmtModified>Tue, 19-Feb-2019 10:20:14 GMT</GmtModified>
      <DeviceStatus>OFFLINE</DeviceStatus>
      <IotId>nadRfljdEndlfadgadgdfse****</IotId>
      <Nickname>SensorInShanghai</Nickname>
    </DeviceInfo>
  </Data>
</QueryDeviceResponse>
```

```

        <DeviceId>QwTxRbhFwcaJBad****</DeviceId>
        <DeviceName>led_driver</DeviceName>
        <ProductKey>a1dafDE****</ProductKey>
        <DeviceSecret>6ahkx2c0d0kVPsjnVmxYaGEhMfdf****</
DeviceSecret>
        <UtcModified>2019-01-30T11:10:10.000Z</UtcModified>
        <GmtCreate>Wed, 30-Jan-2019 11:10:10 GMT</GmtCreate>
        <UtcCreate>2019-01-30T11:10:10.000Z</UtcCreate>
        <GmtModified>Wed, 30-Jan-2019 11:10:10 GMT</GmtModified>
        <DeviceStatus>UNACTIVE</DeviceStatus>
        <IotId>nadddfRddfaEEadfed****</IotId>
        <Nickname>DriverInShanghai</Nickname>
    </DeviceInfo>
</Data>
<PageSize>10</PageSize>
<RequestId>A4F5820D-E2B3-48D4-AAF8-53FF6A319E27</RequestId>
<Success>true</Success>
<Total>2</Total>
</QueryDeviceResponse>

```

2.8.4 DeleteDevice

调用该接口删除指定设备。

接口说明

如果您不再需要某台设备，您可以将其删除。设备删除后，设备ID（IotId）将失效，与设备关联的其他信息也一并删除，您将无法执行与该设备关联的任何操作。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteDevice。
IotId	String	否	要删除的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey&DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要删除的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName

名称	类型	是否必需	描述
DeviceName	String	否	指定要删除的设备的名称。  说明: 如果传入该参数, 需同时传入ProductKey。 。
公共请求参数	-	是	公共请求参数, 请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功, false表示调用失败。
ErrorMessage	String	调用失败时, 返回的出错信息。
Code	String	调用失败时, 返回的错误码。错误码详情, 请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteDevice
&IotId=MpEKNuEUJzIORNANAWJX0010929900*****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
}
```

· XML示例

```
<?xml version="1.0" encoding="UTF-8"?>
<DeleteDeviceResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</DeleteDeviceResponse>
```

</DeleteDeviceResponse>

2.8.5 GetDeviceStatus

调用该接口查看指定设备的运行状态。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetDeviceStatus。
IotId	String	否	要查看运行状态的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey&DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查看运行状态的设备隶属的产品的Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	指定要查看运行状态的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。

名称	类型	描述
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回设备状态信息。详情参见下表 DeviceStatusInfo 。

表 2-19: DeviceStatusInfo

名称	类型	描述
Status	String	设备状态。取值： ONLINE：设备在线。 OFFLINE：设备离线。 UNACTIVE：设备未激活。 DISABLE：设备已禁用。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetDeviceStatus
&IotId=MpEKNuEUJzIORNANAWJX0010929900*****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
  "Data": {
    "Status": "UNACTIVE"
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<GetDeviceStatusResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <Status>UNACTIVE</Status>
  </Data>
```

</GetDeviceStatusResponse>

2.8.6 BatchGetDeviceState

调用该接口批量查看同一产品下指定设备的运行状态。

限制说明

该接口用于查看一个产品下多个设备的运行状态，单次最多可查询50个设备。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：BatchGetDeviceState。
ProductKey	String	否	要查看运行状态的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	List<String>	否	要查看运行状态的设备的名称列表。  说明： · 如果传入该参数，需同时传入ProductKey。 · 单次查询最多50个设备。
IotIds	List	否	要查看运行状态的设备ID列表。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
DeviceStatusList	List	调用成功时，返回设备状态信息列表。详情参见下表 DeviceStatus。

表 2-20: DeviceStatus

名称	类型	描述
DeviceName	String	设备名称。
Status	String	设备状态。取值： ONLINE：设备在线。 OFFLINE：设备离线。 UNACTIVE：设备未激活。 DISABLE：设备已禁用。
DeviceId	String	设备ID（旧版参数）。  说明： 该参数是旧版本遗留参数，已无实际作用，不能用来标识设备。目前，有效的设备标识符为IotId和ProductKey 与DeviceName组合。
IotId	String	设备ID，物联网平台为设备颁发的唯一标识。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=BatchGetDeviceState
&productKey=al*****
&DeviceName.1=device1
&DeviceName.2=device2
```

&公共请求参数

返回示例

· JSON格式

```
{
  "DeviceStatusList": {
    "DeviceStatus": [
      {
        "Status": "UNACTIVE",
        "DeviceId": "xHkR9exGvarIjZ*****",
        "DeviceName": "device1",
        "IotId": "xHkR9exGvarIjZzpZ0H*****"
      },
      {
        "Status": "UNACTIVE",
        "DeviceId": "YXm137sPvABrJy*****",
        "DeviceName": "device2",
        "IotId": "YXm137sPvABrJyqoKP*****"
      }
    ]
  },
  "RequestId": "D9E3DADE-5B9D-4594-AE0E-8090C16B485D",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<BatchGetDeviceStateResponse>
  <DeviceStatusList>
    <DeviceStatus>
      <Status>UNACTIVE</Status>
      <DeviceId>xHkR9exGvarIjZz*****</DeviceId>
      <DeviceName>device1</DeviceName>
      <IotId>xHkR9exGvarIjZzpZ0HW*****</IotId>
    </DeviceStatus>
    <DeviceStatus>
      <Status>UNACTIVE</Status>
      <DeviceId>YXm137sPvABrJy*****</DeviceId>
      <DeviceName>device2</DeviceName>
      <IotId>YXm137sPvABrJyqoKP2o*****</IotId>
    </DeviceStatus>
  </DeviceStatusList>
  <RequestId>D9E3DADE-5B9D-4594-AE0E-8090C16B485D</RequestId>
  <Success>true</Success>
</BatchGetDeviceStateResponse>
```

2.8.7 DisableThing

调用该接口禁用指定设备。

接口说明

使用该接口可以禁用一个设备。设备被禁用后将不能接入物联网平台，您将无法执行与设备有关的操作，但与设备关联的信息依然保留。您可以调用EnableThing接口重新接入被禁用的设备。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DisableThing。
IotId	String	否	要禁用的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要禁用的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	指定禁用的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DisableThing
```

```
&IotId=MpEKNuEUJzIORNANAWJX0010929900*****  
&公共请求参数
```

返回示例

· JSON格式

```
{  
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",  
  "Success": true  
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>  
<DisableThingResponse>  
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>  
  <Success>true</Success>  
</DisableThingResponse>
```

2.8.8 EnableThing

调用该接口解除指定设备的禁用状态，即启用被禁用的设备。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：EnableThing。
IotId	String	否	要禁用的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要禁用的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	指定禁用的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。

名称	类型	是否必需	描述
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=EnableThing
&IotId=SR8FiTu1R9tlUR2V1bmi0010a5****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<EnableThingResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</EnableThingResponse>
```

2.8.9 BatchCheckDeviceNames

调用该接口在指定产品下批量自定义设备名称。IoT平台将检查名称的合法性。

使用说明

- 单次调用，最多能传入1,000 个设备名称。

- 该接口和BatchRegisterDeviceWithApplyId接口结合使用，在一个产品下批量注册多个设备，并且为每个设备单独命名。

批量注册设备流程：

- 调用本接口，传入要批量注册的设备的名称，物联网平台返回申请批次ID（ApplyId）。返回成功结果，表示批量校验设备名称的申请已经提交成功。实际的校验是异步执行的，会有一个过程。
- 调用[QueryBatchRegisterDeviceStatus](#)查看名称设置结果。
- 调用[BatchRegisterDeviceWithApplyId](#)批量注册设备。
- （可选）调用[QueryBatchRegisterDeviceStatus](#)查看设备注册结果。
- 调用[QueryPageByApplyId](#)查看批量注册的设备信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：BatchCheckDeviceNames。
ProductKey	String	是	要批量注册的设备所隶属的产品Key。
DeviceName	List<String>	是	要批量注册的设备的名称列表。每个设备名称应包含4-32个字符，可以包含英文字母、数字和特殊字符（连字符、下划线、@符号、点号、和英文冒号）。  说明： 单次调用，最多能传入1,000个设备名称。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

名称	类型	描述
Data	Data	调用成功时，返回的数据。详情见下表Data。

表 2-21: Data

名称	类型	描述
ApplyId	Long	调用成功时，系统返回的申请批次ID。使用该ApplyId，调用BatchRegisterDeviceWithApplyId接口来批量创建设备。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=BatchCheckDeviceNames
&productKey=a1*****
&DeviceName.1=device1
&DeviceName.3=device2
&DeviceName.2=device3
&DeviceName.4=device4
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "ApplyId": 4399
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<BatchCheckDeviceNamesResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <ApplyId>4401</ApplyId>
  </Data>
```

```
</BatchCheckDeviceNamesResponse>
```

2.8.10 BatchRegisterDeviceWithApplyId

调用该接口根据申请批次ID（ApplyId）批量注册设备。

使用说明

该接口需要和[BatchCheckDeviceNames](#)接口结合使用，实现在一个产品下批量注册（即新建）多个设备，并且为每个设备单独命名。

建议按照以下流程批量注册设备：

1. 调用[BatchCheckDeviceNames](#)接口，传入要批量注册的设备的名称。物联网平台检查您提交的设备名称符合要求后，返回申请批次ID（ApplyId）。
2. 调用[QueryBatchRegisterDeviceStatus](#)查看设备名称设置结果。
3. 调用本接口批量注册设备。本接口调用返回的成功结果，仅表示批量注册的申请已经提交成功。实际的注册会有一个过程。
4. 调用[QueryBatchRegisterDeviceStatus](#)查看设备注册结果（这一步不是必须步骤）。
5. 调用[QueryPageByApplyId](#)查看批量注册的设备信息。



说明：

如果您想在一个产品下批量注册多个设备，且不指定设备名称，设备名称由系统随机生成，您可以调用[BatchRegisterDevice](#)接口。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：BatchRegisterDeviceWithApplyId。
ProductKey	String	是	要批量注册的设备所隶属的产品Key。
ApplyId	Long	是	要批量注册的设备的申请批次ID。申请批次ID由调用BatchCheckDeviceNames接口返回。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。

名称	类型	描述
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的数据。详情见下表Data。

表 2-22: Data

名称	类型	描述
ApplyId	Long	传入的申请批次ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=BatchRegisterDeviceWithApplyId
&ProductKey=a1*****
&ApplyId=4416
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "ApplyId": 4416
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<BatchRegisterDeviceWithApplyId>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <ApplyId>4416</ApplyId>
  </Data>
</BatchRegisterDeviceWithApplyId>
```

```
</BatchRegisterDeviceWithApplyId>
```

2.8.11 BatchRegisterDevice

调用该接口在指定产品下批量注册多个设备（随机生成设备名）。

限制说明

- 调用本接口，在一个产品下批量注册多个设备，并且随机生成设备名。单次调用，最多可创建1,000个设备。

如果您想在一个产品下单独注册一个设备，您可以调用RegisterDevice接口。

如果您想在一个产品下批量注册多个设备，并且为每个设备单独命名，您需要分步调用BatchCheckDeviceNames和BatchRegisterDeviceWithApplyId接口。

- 返回成功结果，表示批量注册的申请已经提交成功。实际的注册是异步执行的，会有一个过程。请调用QueryBatchRegisterDeviceStatus接口查看详细结果。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：BatchRegisterDevice。
ProductKey	String	是	要批量注册的设备所隶属的产品Key。
Count	Integer	是	要注册的设备数量，取值不能大于1,000。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的数据。详情见下表Data。

表 2-23: Data

名称	类型	描述
ApplyId	Long	调用成功时，系统返回的申请批次ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=BatchRegisterDevice
&ProductKey=a1*****
&Count=10
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "ApplyId": 4415
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<BatchRegisterDeviceResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <ApplyId>4415</ApplyId>
  </Data>
</BatchRegisterDeviceResponse>
```

2.8.12 BatchUpdateDeviceNickname

调用该接口批量修改设备备注名称。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：BatchUpdateDeviceNickname。

名称	类型	是否必需	描述
DeviceNicknameInfo	String	是	包含设备标识参数（ProductKey和DeviceName组合或IotId）和备注名称参数（Nickname）。  说明: - 设备标识信息为必传参数，用于指定设备。 - Nickname为非必填参数。若不传入，则删除该设备的原有备注名称。 详情请见下表DeviceNicknameInfo。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

表 2-24: DeviceNicknameInfo

名称	类型	是否必需	描述
IotId	String	否	要修改别名的设备ID。  说明: 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，和ProductKey与DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明: 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	指定要查询的设备的名称。  说明: 如果传入该参数，需同时传入ProductKey。

名称	类型	是否必需	描述
Nickname	String	否	新的设备备注名称。备注名称长度为4-32个字符，可包含中文汉字、英文字母、数字和下划线。一个中文汉字算2字符。  说明： 若不传入该参数，则删除该设备的原有备注名称。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=BatchUpdateDeviceNickname
&DeviceNicknameInfo.1.ProductKey=a1rYuVF****
&DeviceNicknameInfo.1.DeviceName=SR8FiTu1R9tlUR2V1bmi
&DeviceNicknameInfo.1.Nickname=airconditioning_type1
&DeviceNicknameInfo.2.ProductKey=a1yrZMH****
&DeviceNicknameInfo.2.DeviceName=RkQ8CFtNpDok4BEunymt
&DeviceNicknameInfo.2.Nickname=airconditioning_type2
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<BatchUpdateDeviceNicknameResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</BatchUpdateDeviceNicknameResponse>
```

</BatchUpdateDeviceNicknameResponse>

2.8.13 QueryBatchRegisterDeviceStatus

调用该接口查询批量注册设备申请的处理状态和结果。

请求参数

名称	类型	是否必须	描述
Action	String	是	要执行的操作，取值：QueryBatchRegisterDeviceStatus。
ProductKey	String	是	要查询的设备所隶属的产品Key。
ApplyId	Long	是	要查询的申请批次ID。申请批次ID在成功调用BatchRegisterDeviceWithApplyId或BatchRegisterDeviceStatus接口的返回结果中。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的状态信息。详情参见 DeviceApplyStatusInfo 。

表 2-25: DeviceApplyStatusInfo

名称	类型	描述
Status	String	申请单的处理状态和结果，取值： CHECK：校验。 CHECK_SUCCESS：校验成功。 CHECK_FAILED：校验失败。 CREATE：创建。 CREATE_SUCCESS：创建成功。 CREATE_FAILED：创建失败。
ValidList	List	- 当返回Status参数值为CHECK_FAILED或CREATE_FAILED时，返回创建成功的设备名称集合。 - 当返回Status参数值为CHECK_SUCCESS或CREATE_SUCCESS时，该参数为空数组。
InvalidList	List	- 当返回Status参数值为CHECK_FAILED或CREATE_FAILED时，创建失败的设备名称集合。 - 当返回Status参数值为CHECK_SUCCESS或CREATE_SUCCESS时，该参数为空数组。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryBatchRegisterDeviceStatus
&ProductKey=a1*****
&ApplyId=4416
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "Status": "CREATE_SUCCESS",
    "ValidList": {
      "Name": [
```

```

    ]
  },
  "InvalidList": {
    "Name": [
    ]
  }
}
}
}

```

· XML格式

```

<?xml version='1.0' encoding='utf-8'?>
<QueryBatchCheckDeviceNamesStatusResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <Status>CREATE_SUCCESS</Status>
    <ValidList/>
    <InvalidList/>
  </Data>
</QueryBatchCheckDeviceNamesStatusResponse>

```

2.8.14 QueryPageByApplyId

调用该接口查询批量注册的设备信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryPageByApplyId。
ApplyId	Long	是	要查询的申请批次ID。申请批次ID可在 BatchRegisterDeviceWithApplyId 和 BatchRegisterDevice 接口返回结果中查看。
PageSize	Integer	否	指定返回结果中每页显示的记录数量。数量限制：每页最多可显示50条。默认值是10。
CurrentPage	Integer	否	指定从返回结果中的第几页开始显示。默认值是1。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。

名称	类型	描述
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
PageCount	Integer	总页数。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页面号。
Total	Integer	总记录数。
ApplyDeviceList	List	调用成功时生成的已注册的设备列表。详情参见下表 ApplyDeviceInfo。

表 2-26: ApplyDeviceInfo

名称	类型	描述
DeviceId	String	设备ID（旧版）。  说明： 该参数是旧版本遗留参数，已无实际作用，不能用来标识设备。目前，有效的设备标识符为IotId和ProductKey 与DeviceName组合。
DeviceName	String	设备名称。
DeviceSecret	String	设备密钥。  说明： 请妥善保管该信息。
IotId	String	设备ID，物联网平台为该设备颁发的唯一标识符。  说明： 请妥善保管该信息。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryPageByApplyId
&ApplyId=4416
&PageSize=10
&CurrentPage=1
```

&公共请求参数

返回示例

· JSON格式

```
{
  "PageCount": 1,
  "ApplyDeviceList": {
    "ApplyDeviceInfo": [
      {
        "DeviceId": "gQG2GJ2y10m6hIk87jFm",
        "DeviceName": "test1",
        "DeviceSecret": "SkfeXXKrTgp1DbDxYr74mfJ5cnu*****",
        "IotId": "nadRfljdEndlfadgadfaj*****"
      },
      {
        "DeviceId": "STGEURQaU1o794X7Kh4C",
        "DeviceName": "test2",
        "DeviceSecret": "QJxYeGXrif7nfxjyWPA1Qss5CFj0*****",
        "IotId": "nddRfljdFndlfsdfaHfaj*****"
      }
    ]
  },
  "Page": 1,
  "PageSize": 10,
  "RequestId": "6C5DCB26-47A5-4CE0-BDC4-AD74782BC3A9",
  "Success": true,
  "Total": 2
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryPageByApplyIdResponse>
  <PageCount>1</PageCount>
  <ApplyDeviceList>
    <ApplyDeviceInfo>
      <DeviceId>gQG2GJ2y10m6hIk87jFm</DeviceId>
      <DeviceName>test1</DeviceName>
      <DeviceSecret>SkfeXXKrTgp1DbDxYr74mfJ5cnu*****</DeviceSecret>
      <IotId>nadRfljdEndlfadgadfaj*****</IotId>
    </ApplyDeviceInfo>
    <ApplyDeviceInfo>
      <DeviceId>STGEURQaU1o794X7Kh4C</DeviceId>
      <DeviceName>test2</DeviceName>
      <DeviceSecret>QJxYeGXrif7nfxjyWPA1Qss5CFj0*****</DeviceSecret>
      <IotId>nddRfljdFndlfsdfaHfaj*****</IotId>
    </ApplyDeviceInfo>
  </ApplyDeviceList>
  <Page>1</Page>
  <PageSize>10</PageSize>
  <RequestId>6C5DCB26-47A5-4CE0-BDC4-AD74782BC3A9</RequestId>
  <Success>true</Success>
  <Total>2</Total>
</QueryPageByApplyIdResponse>
```

</QueryPageByApplyIdResponse>

2.8.15 QueryDeviceEventData

调用该接口查询指定设备的事件记录。

限制说明

仅能查询最近30天内的事件记录数据。



说明:

数据存储时间从事件生成当日起开始计算。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceEventData。
IotId	String	否	要查询的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。

名称	类型	是否必需	描述
EventType	String	否	要查询的事件类型。取值： info：信息。 alert：告警。 error：故障。
Identifier	String	是	要查询的事件标识符。高级版设备的事件 Identifier，可在控制台中设备所属的高级版产品的功能定义中查看。
StartTime	Long	是	要查询的事件记录的开始时间。  说明： 只能查询最近30天内的数据。
EndTime	Long	是	要查询的事件记录的结束时间。
PageSize	Integer	否	返回结果中每页显示的记录数。数量限制：每页最多可显示50条。
Asc	Integer	否	返回结果中事件记录的排序方式，取值： 0：倒序。 1：正序。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的设备事件记录。详情参见 Data 。

表 2-27: Data

名称	类型	描述
List	List<EventInfo>	事件集合。每个元素代表一个事件。元素的结构参见 EventInfo 。
NextValid	Boolean	表示下一页是否可用。true表示可用，false表示不可用。
NextTime	Long	下一页中的事件记录的起始时间。

表 2-28: EventInfo

名称	类型	描述
Name	String	事件名称。
Time	Long	事件发生时间。
OutputData	String	事件的输出参数，map格式的字符串。具体结构参见 OutputData 。
EventType	String	事件类型。
Identifier	String	事件标识符。

表 2-29: OutputData

名称	类型	描述
Key	String	参数标识符。
Value	Long	参数值。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceEventData
&IotId=abcdef1234567*****
&ProductKey=al*****
&DeviceName=device1
&EventType=info
&Identifier=lightLevel
&StartTime=1516538300303L
&EndTime=1516541900303L
&PageSize=10
&Asc=1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "NextValid": false,
    "NextTime": 1516449221253,
    "List": {
      "EventInfo": [
        {
          "Name": "testEventInfoName",
          "Time": "1516517974638",
          "OutputData": "{\"structArgs\":{\"structchildFLOATf71c20e\":1.23,\"structchildINT6b6b626\":3,\"structchildDATE663436a\":\"1516517966152\",\"structchildDOUBLE08d0f74\":1.23,\"structchildTEXTdc764f9\":\"07b68264b0ba42c18e5f\",\"structchildB00Ld260729\":0,\"structchildENUMbe62590\":1},\"enumArgs\":0,\"boolArgs\":0,\"floatArgs\":2.3,\"dateArgs\":\"1516517966152\",\"intArgs\":1,\"doubleArgs\":2.3,\"textArgs\":\"dV56zbkzjBjw1Ti1dA52\"}",
          "EventType": "info",
          "Identifier": "testEventInfo"
        },
        {
          "Name": "testEventInfoName",
          "Time": "1516449221254",
          "OutputData": "{\"structArgs\":{\"structchildFLOATf71c20e\":1.23,\"structchildINT6b6b626\":3,\"structchildDATE663436a\":\"1516449212507\",\"structchildDOUBLE08d0f74\":1.23,\"structchildTEXTdc764f9\":\"a1f3583dde3944289639\",\"structchildB00Ld260729\":0,\"structchildENUMbe62590\":1},\"enumArgs\":0,\"boolArgs\":0,\"floatArgs\":2.3,\"dateArgs\":\"1516449212507\",\"intArgs\":1,\"doubleArgs\":2.3,\"textArgs\":\"1z4XNBvvA7eZw8XViaJp\"}",
          "EventType": "info",
          "Identifier": "testEventInfo"
        }
      ]
    }
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<QueryDeviceEventDataResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <NextValid>false</NextValid>
    <NextTime>1516449221253</NextTime>
    <List>
      <EventInfo>
        <Name>testEventInfoName</Name>
        <Time>1516517974638</Time>
        <OutputData>{"structArgs":{"structchildFLOATf71c20e":1.23,"structchildINT6b6b626":3,"structchildDATE663436a":"1516517966152","structchildDOUBLE08d0f74":1.23,"structchildTEXTdc764f9":"07b68264b0ba42c18e5f","structchildB00Ld260729":0,"structchildENUMbe62590":1,"enumArgs":0,"boolArgs":0,"floatArgs":2.3,"dateArgs":"1516517966152","intArgs":1,"doubleArgs":2.3,"textArgs":"dV56zbkzjBjw1Ti1dA52"}</OutputData>
        <EventType>info</EventType>
      </EventInfo>
    </List>
  </Data>
</QueryDeviceEventDataResponse>
```

```
<Identifier>testEventInfo</Identifier>
</EventInfo>
<EventInfo>
  <Name>testEventInfoName</Name>
  <Time>1516449221254</Time>
  <OutputData>{"structArgs":{"structchildFLOATf71c
20e":1.23,"structchildINT6b6b626":3,"structchildDATE663436a":
1516449212507","structchildDOUBLE08d0f74":1.23,"structchildTEXTdc764
f9":"a1f3583dde3944289639","structchildB00Ld260729":0,"structchil
dENUMbe62590":1},"enumArgs":0,"boolArgs":0,"floatArgs":2.3,"
dateArgs":"1516449212507","intArgs":1,"doubleArgs":2.3,"textArgs":"
1z4XNBvvA7eZw8XViaJp"}</OutputData>
  <EventType>info</EventType>
  <Identifier>testEventInfo</Identifier>
</EventInfo>
</List>
</Data>
</QueryDeviceEventDataResponse>
```

2.8.16 QueryDevicePropertyData

调用该接口查询指定设备的属性记录。

限制说明

仅能查询最近30天内的属性数据。



说明:

数据存储时间从属性时间戳表示的时间当日开始计算。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDevicePropertyData。
IotId	String	否	要查询的设备ID，设备的唯一标识符。  说明: 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。

名称	类型	是否必需	描述
ProductKey	String	否	指定要查询的设备隶属的产品Key。  说明: 如果传入该参数, 需同时传入DeviceName
DeviceName	String	否	指定要查询的设备的名称。  说明: 如果传入该参数, 需同时传入ProductKey
Identifier	String	是	要查询的属性标识符。 设备的属性Identifier, 可在控制台中设备所属的产品的功能定义中查看。若设备有多个属性, 您可以多次调用该接口进行查询, 一次输入一个Identifier。
StartTime	Long	是	要查询的属性记录的开始时间。取值为毫秒值时间戳。  说明: 只能查询最近30天内的属性数据记录。
EndTime	Long	是	要查询的属性记录的结束时间。取值为毫秒值时间戳。
PageSize	Integer	是	返回结果中每页显示的记录数。数量限制: 每页最多可显示50条。
Asc	Integer	是	返回结果中属性记录的排序方式, 取值: 0: 倒序。 1: 正序。
公共请求参数	-	是	请参见公共参数。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见本文错误码章节。
Data	Data	调用成功时，返回的设备属性记录。详情参见下表Data。

表 2-30: Data

名称	类型	描述
List	List<PropertyList>	属性集合。每个元素代表一个属性。元素的结构参见下表PropertyInfo。  说明： 返回的属性信息按照属性生成时间倒序排列。
NextValid	Boolean	是否有下一页属性记录。true表示有，false表示没有。 返回NextValid为true时，可以将NextTime的值作为下次查询的StartTime，查询下一页数据。
NextTime	Long	下一页面中的属性记录的起始时间。 调用本接口查询下一页属性记录时，该值可作为入参StartTime的值。

表 2-31: PropertyInfo

名称	类型	描述
Value	String	属性值。
Time	Long	属性修改时间。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDevicePropertyData
```

```
&IotId=SR8FiTu1R9tlUR2V1bmi0010*****  
&ProductKey=a1*****  
&DeviceName=device1  
&Identifier=lightLevel  
&StartTime=1516538300303  
&EndTime=1516541900303  
&PageSize=10  
&Asc=1  
&公共请求参数
```

返回示例

· JSON格式

```
{  
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",  
  "Success": true,  
  "Data": {  
    "NextValid": true,  
    "NextTime": 1516541821599,  
    "List": {  
      "PropertyInfo": [  
        {  
          "Value": "32",  
          "Time": 1516541894876  
        },  
        {  
          "Value": "2",  
          "Time": 1516541885630  
        },  
        {  
          "Value": "95",  
          "Time": 1516541875947  
        },  
        {  
          "Value": "14",  
          "Time": 1516541830905  
        },  
        {  
          "Value": "78",  
          "Time": 1516541821600  
        }  
      ]  
    }  
  }  
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>  
<QueryDevicePropertyDataResponse>  
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>  
  <Success>true</Success>  
  <Data>  
    <NextValid>true</NextValid>  
    <NextTime>1516541821599</NextTime>  
    <List>  
      <PropertyInfo>  
        <Value>32</Value>  
        <Time>1516541894876</Time>  
      </PropertyInfo>  
      <PropertyInfo>
```

```
<Value>2</Value>
<Time>1516541885630</Time>
</PropertyInfo>
<PropertyInfo>
  <Value>95</Value>
  <Time>1516541875947</Time>
</PropertyInfo>
<PropertyInfo>
  <Value>14</Value>
  <Time>1516541830905</Time>
</PropertyInfo>
<PropertyInfo>
  <Value>78</Value>
  <Time>1516541821600</Time>
</PropertyInfo>
</List>
</Data>
</QueryDevicePropertyDataResponse>
```

2.8.17 QueryDevicePropertiesData

调用该接口批量查询指定设备的属性上报数据。

说明

- 此接口为产品特有接口。
- 一次调用最多可查询10个属性的历史数据。
- 每个属性最多返回100条数据。
- 仅能查询最近30天内的属性数据。



说明:

数据存储时间从属性生成当日开始计算。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QueryDevicePropertiesData。
ProductKey	String	是	设备所属的产品Key。
DeviceName	String	是	设备名称。
Identifiers	List	是	属性的标识符列表。设备的属性Identifier信息，可在控制台中，设备所属的产品的功能定义中查看。 不可输入重复的属性Identifier。

名称	类型	是否必需	描述
StartTime	Long	是	属性记录的开始时间。取值为13位毫秒值时间戳。  说明： 只能查询最近30天内的数据。
EndTime	Long	是	属性记录的结束时间。取值为13位毫秒值时间戳。
PageSize	Integer	是	单个属性可返回的数据记录数量。最大值为100。 任意一个属性返回的数据记录数量不超过该值。
Asc	Integer	是	返回结果中，属性记录按时间排序的方式。取值： · 0：倒序。倒序查询时，StartTime必须大于EndTime。 · 1：正序。正序查询时，StartTime必须小于EndTime。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
NextValid	Boolean	是否有下一页属性记录。true表示有，false表示没有。 返回NextValid为true时，可以将NextTime的值作为下次查询的StartTime，继续查询本次查询不显示的数据。
NextTime	Long	下一页属性记录的起始时间。 可以将NextTime的值作为下次查询的StartTime，继续查询本次查询不显示的数据。
Code	String	调用失败时，返回的错误码。请参见 错误码 。

名称	类型	描述
PropertyDataInfos	List	调用成功时，返回的属性信息列表。具体信息，请参见下表PropertyDataInfo。

表 2-32: PropertyDataInfo

名称	类型	描述
Identifier	String	属性的标识符。
List	List	属性数据列表。具体信息，请参见下表PropertyInfo。

表 2-33: PropertyInfo

名称	类型	描述
Time	Long	属性上报时间。
Value	String	属性值。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDevicePropertiesData
&Asc=0
&DeviceName=water
&EndTime=1540115948152
&Identifier.1=Weight
&Identifier.2=Circle
&PageSize=100
&ProductKey=a1bd0xkDbGC
&StartTime=1540116010723
&公共请求参数
```

返回示例

· JSON格式

```
{
  "NextValid": true,
  "NextTime": 1540115949818,
  "RequestId": "75649FF8-36CD-421B-96C6-35B725FE823B",
  "Success": true
  "PropertyDataInfos": {
    "PropertyDataInfo": [{
      "List": {
        "PropertyInfo": [{
          "Value": "166.0",
          "Time": 1540116010518
        }],
        {
          "Value": "166.0",
          "Time": 1540116009906
        }
      ]
    }
  ]
}
```

```

    },
    {
      "Value": "134.0",
      "Time": 1540115951051
    },
    {
      "Value": "133.0",
      "Time": 1540115950431
    },
    {
      "Value": "133.0",
      "Time": 1540115949819
    }
  ]
},
{
  "Identifier": "Circle"
},
{
  "List": {
    "PropertyInfo": [{
      "Value": "99.0",
      "Time": 1540116010314
    },
    {
      "Value": "99.0",
      "Time": 1540116009702
    },
    {
      "Value": "98.0",
      "Time": 1540116009090
    },
    {
      "Value": "51.0",
      "Time": 1540115950844
    },
    {
      "Value": "50.0",
      "Time": 154011595228
    }
  ]
},
{
  "Identifier": "Weight"
}
],
},
}

```

· XML格式

```

<?xml version="1.0" encoding="UTF-8" ?>
<QueryDevicePropertiesData>
  <NextValid>true</NextValid>
  <NextTime>1540115949818</NextTime>
  <RequestId>75649FF8-36CD-421B-96C6-35B725FE823B</RequestId>
  <PropertyDataInfos>
    <PropertyDataInfo>
      <List>
        <PropertyInfo>
          <Value>166.0</Value>
          <Time>1540116010518</Time>
        </PropertyInfo>
        <PropertyInfo>

```

```
<Value>166.0</Value>
<Time>1540116009906</Time>
</PropertyInfo>
<PropertyInfo>
  <Value>134.0</Value>
  <Time>1540115951051</Time>
</PropertyInfo>
<PropertyInfo>
  <Value>133.0</Value>
  <Time>1540115950431</Time>
</PropertyInfo>
<PropertyInfo>
  <Value>133.0</Value>
  <Time>1540115949819</Time>
</PropertyInfo>
</List>
<Identifier>Circle</Identifier>
</PropertyDataInfo>
<PropertyDataInfo>
  <List>
    <PropertyInfo>
      <Value>99.0</Value>
      <Time>1540116010314</Time>
    </PropertyInfo>
    <PropertyInfo>
      <Value>99.0</Value>
      <Time>1540116009702</Time>
    </PropertyInfo>
    <PropertyInfo>
      <Value>98.0</Value>
      <Time>1540116009090</Time>
    </PropertyInfo>
    <PropertyInfo>
      <Value>51.0</Value>
      <Time>1540115950844</Time>
    </PropertyInfo>
    <PropertyInfo>
      <Value>50.0</Value>
      <Time>1540115950228</Time>
    </PropertyInfo>
  </List>
  <Identifier>Weight</Identifier>
</PropertyDataInfo>
</PropertyDataInfos>
<Success>true</Success>
</QueryDevicePropertiesData>
```

2.8.18 QueryDeviceServiceData

调用该接口查询指定设备的服务记录。

限制说明

仅能查询最近30天内的服务数据。



说明:

数据存储时间从服务生成当日开始计算。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceServiceData。
IotId	String	否	要查询的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey&DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
Identifier	String	是	要查询的服务标识符。高级版设备的服务Identifier。可在控制台中设备所属的高级版产品的功能定义中查看。
StartTime	Long	是	要查询的服务记录的开始时间。取值为毫秒值时间戳。  说明： 只能查询最近30天内的数据。
EndTime	Long	是	要查询的服务记录的结束时间。取值为毫秒值时间戳。
PageSize	Integer	否	返回结果中每页显示的记录数。数量限制：每页最多可显示50条。

名称	类型	是否必需	描述
Asc	Integer	否	返回结果中服务记录的排序方式，取值： 0：倒序。 1：正序。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的设备服务记录。详情参见 Data 。

表 2-34: Data

名称	类型	描述
List	List<ServiceInfo>	服务记录集合。每个元素代表一个服务记录。元素的结构参见 ServiceInfo 。
NextValid	Boolean	表示下一页面是否可用。true表示可用，false表示不可用。
NextTime	Long	下一页面中的服务记录的起始时间。

表 2-35: ServiceInfo

名称	类型	描述
Name	String	服务名称。
Time	Long	服务执行的时间。
OutputData	String	服务的输出参数，map格式的字符串，结构为 key:value。

名称	类型	描述
InputData	String	服务的输入参数，map格式的字符串，结构为 key:value。
Identifier	String	服务标识符。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceServiceData
&IotId=SR8FiTu1R9tlUR2V1bmi0010*****
&ProductKey=a1*****
&DeviceName=device1
&Identifier=set
&StartTime=1516538300303L
&EndTime=1516541900303L
&PageSize=10
&Asc=1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "NextValid": true,
    "NextTime": 1517315865197,
    "List": {
      "ServiceInfo": [
        {
          "Name": "set",
          "Time": 1517315865198,
          "OutputData": "{\"code\":200,\"data\":{},\"id\":\"100686\", \"message\":\"success\", \"version\":\"1.0\"}",
          "InputData": "{\"LightAdjustLevel\":123}",
          "Identifier": "set"
        }
      ]
    }
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<QueryDeviceServiceDataResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <NextValid>true</NextValid>
    <NextTime>1517315865197</NextTime>
    <List>
      <ServiceInfo>
        <Name>set</Name>
```

```

        <Time>1517315865198</Time>
        <OutputData>{"code":200,"data":{},"id":"100686","
message":"success","version":"1.0"}</OutputData>
        <InputData>{"LightAdjustLevel":123}</InputData>
        <Identifier>set</Identifier>
    </ServiceInfo>
</List>
</Data>
</QueryDeviceServiceDataResponse>

```

2.8.19 InvokeThingService

调用该接口在一个设备上调用指定服务。

限制说明

同步调用服务，最大超时时间为5秒。若5秒内服务器未收到回复，则返回超时错误。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：InvokeThingService。
IotId	String	否	要调用服务的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，和ProductKey与DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。不能传入空参数。
ProductKey	String	否	要调用服务的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要调用服务的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。

名称	类型	是否必需	描述
Identifier	String	是	服务的标识符。设备的服务Identifier，可在控制台中设备所属的产品的功能定义中查看。
Args	String	是	要启用服务的入参信息，数据格式为JSON String，如，Args={"param1":1}。 若此参数为空时，需传入 Args={}。 Args详情参见下表Args。
公共请求参数	-	是	请参见 公共参数 。

表 2-36: Args

名称	类型	描述
key	String	输入参数的标识符。您在创建该服务时，设置的输入参数的标识符。 您可以在控制台设备所属的产品的功能定义页面，从该产品的物模型中查看，或单击该服务对应的编辑按钮，然后查看您设置的输入参数的信息。
value	Object	指定参数值。该值须在您设置的输入参数的取值范围内。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情,请参见 错误码 。
Data	Data	调用成功时，返回的数据。详情请见下表Data。

表 2-37: Data

名称	类型	描述
Result	String	同步调用服务，返回的调用结果。 异步调用服务，不返回此参数。
MessageId	String	云端向设备下发服务调用的消息ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=InvokeThingService
&ProductKey=a1*****
&DeviceName=device1
&Identifier=service
&Args=%7B%22param1%22%3A1%7D
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "Result": "...",
    "MessageId": "abcabc123"
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<InvokeThingServiceResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <Result>...</Result>
    <MessageId>abcabc123</MessageId>
  </Data>
```

```
</InvokeThingServiceResponse>
```

2.8.20 InvokeThingsService

调用该接口批量调用设备服务。

使用限制

- 单账号的每秒请求数最大限制为50 QPS。



说明:

子账号共享主账号配额。

- 目前只支持异步调用该接口。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：InvokeThingsService。
ProductKey	String	是	要调用服务的设备所隶属的产品Key。
DeviceNames	List<String>	是	要调用服务的设备的名称列表。最多支持500个设备。
Identifier	String	是	服务的标识符。 设备的服务Identifier，可在控制台中设备所属产品的功能定义中查看。
Args	String	是	要启用服务的入参信息，数据格式为JSON String，如，Args={"param1":1}。 若此参数为空时，需传入 Args={}。 Args详情参见下表Args。
公共请求参数	-	是	请参见 公共参数 。

表 2-38: Args

名称	类型	描述
key	String	输入参数的标识符。您在创建该服务时，设置的输入参数的标识符。 您可以在控制台设备所属产品的功能定义页面，从该产品的物模型中查看，或单击该服务对应的编辑按钮，然后查看您设置的输入参数的信息。
value	Object	指定参数值。该值须在您设置的输入参数的取值范围内。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=InvokeThingsService
&Args=%7B%20%20%20%20%22walk%22%3A%22a~z%22%2C%20%20%20%20%20%
22city%22%3A%22shanghai%22%20%7D
&DeviceName.1=1102andriod02
&DeviceName.2=1102android01
&Identifier=TimeReset
&ProductKey=a1hwjHDWUbF
&公共请求参数
```

返回示例

JSON 格式

```
{
  "RequestId": "059C3274-6197-4BEC-95E4-49A076330E57",
  "Success": true
}
```

```
}
```

XML 格式

```
<?xml version='1.0' encoding='utf-8'?>
<InvokeThingsServiceResponse>
  <RequestId>"059C3274-6197-4BEC-95E4-49A076330E57"</RequestId>
  <Success>>true</Success>
</InvokeThingsServiceResponse>
```

2.8.21 GetGatewayBySubDevice

调用该接口，根据挂载的子设备信息，查询对应的网关设备信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetGatewayBySubDevice。
IotId	String	否	子设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey与DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	子设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	子设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的网关设备的详细信息。详情请参见下表 DeviceDetailInfo 。

表 2-39: DeviceDetailInfo

名称	类型	描述
ProductKey	String	网关设备隶属的产品Key。
ProductName	String	网关设备隶属的产品名称。
DeviceName	String	网关设备名称。
DeviceSecret	String	网关设备密钥。
IotId	String	IoT平台为该网关设备颁发的ID，作为该设备的唯一标识符。
GmtCreate	String	网关设备的创建时间，GMT时间。
GmtActive	String	网关设备的激活时间，GMT时间。
GmtOnline	String	网关设备最近一次上线的时间，GMT时间。
UtcCreate	String	网关设备的创建时间，UTC时间。
UtcActive	String	网关设备的激活时间，UTC时间。
UtcOnline	String	网关设备最近一次上线的时间，UTC时间。
Status	String	网关设备状态。取值： ONLINE：设备在线。 OFFLINE：设备离线。 UNACTIVE：设备未激活。 DISABLE：设备已禁用。

名称	类型	描述
FirmwareVersion	String	网关设备的固件版本号。
IpAddress	String	网关设备的IP地址。
NodeType	Integer	节点类型，取值为1，表示该设备为网关设备。
Region	String	网关设备所在地域（与控制台上的地域对应）。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetGatewayBySubDevice
&ProductKey=a1*****S
&DeviceName=XTzosqEOgxFXKPRgd8zl
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "F227A41E-8A0F-4829-A1B1-727619DB58A3",
  "Data": {
    "Status": "UNACTIVE",
    "ProductName": "TEST",
    "DeviceSecret": "nIC0JkFJnG*****TWnvXHydEjX",
    "UtcOnline": "",
    "IotId": "9a1MTdk9brqQ2bhdG40Y001094fd00",
    "GmtCreate": "2018-08-07 15:54:15",
    "UtcCreate": "2018-08-07T07:54:15.000Z",
    "UtcActive": "",
    "GmtActive": "",
    "NodeType": 1,
    "Region": "cn-shanghai",
    "GmtOnline": "",
    "ProductKey": "a15tzTmkWZ2",
    "DeviceName": "d896e0ff000105bb",
    "IpAddress": "10.0.0.1",
    "FirmwareVersion": "1.2.3"
  },
  "Success": true
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<GetGatewayBySubDeviceResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <Status>UNACTIVE</Status>
    <ProductName>TEST</ProductName>
```

```

    <DeviceSecret>nIC0JkFJnG*****TWnvXHydEjX</DeviceSecret>
  >
    <UtcOnline></UtcOnline>
    <IotId>SR8FiTu1R9tlUR2V1bmi0010*****</IotId>
    <GmtCreate>2018-08-07 15:54:15</GmtCreate>
    <UtcCreate>2018-08-07T07:54:15.000Z</GmtOnline>
    <UtcActive></UtcActive>
    <GmtActive></GmtActive>
    <NodeType>1</NodeType>
    <Region>cn-shanghai</Region>
    <GmtOnline></GmtOnline>
    <ProductKey>a15tzTmkWZ2</ProductKey>
    <DeviceName>TEST</DeviceName>
    <IpAddress>10.0.0.1</IpAddress>
    <FirmwareVersion>1.2.3</FirmwareVersion>
  </Data>
</GetGatewayBySubDeviceResponse>

```

2.8.22 QueryDevicePropertyStatus

调用该接口查询指定设备的属性快照。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDevicePropertyStatus。
IotId	String	否	要查询的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey & DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。

名称	类型	是否必需	描述
DeviceName	String	否	要查询的设备的名称。  说明: 如果传入该参数, 需同时传入ProductKey。
公共请求参数		是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功, false表示调用失败。
ErrorMessage	String	调用失败时, 返回的出错信息。
Code	String	调用失败时, 返回的错误码。错误码详情, 请参见 错误码 。
Data	Data	调用成功时, 返回的数据。详情参见下表。

表 2-40: Data

名称	类型	描述
List	List	返回的属性集合信息。详情参见下表PropertyStatusInfo。

表 2-41: PropertyStatusInfo

名称	类型	描述
Identifier	String	属性标识符。
Name	String	属性名称。

名称	类型	描述
DataType	String	属性格式类型，取值： int：整型。 float：单精度浮点型。 double：双精度浮点型。 enum：枚举型。 bool：布尔型。 text：字符型。 date：时间型（String类型的UTC时间戳，单位是毫秒）。 array：数组型。 struct：结构体类型。
Time	String	属性修改的时间，单位是毫秒。
Value	String	属性值。
Unit	String	属性单位。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDevicePropertyStatus
&IotId=SR8FiTu1R9tlUR2V1bmi0010*****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "List": {
      "PropertyStatusInfo": [
        {
          "Name": "doublePropertyName",
          "Value": "50.0",
          "Time": "1517553572362",
          "DataType": "double",
          "Identifier": "doubleProperty",
```

```
        "Unit": "C"
      }
    ]
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<QueryDevicePropertyStatusResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <List>
      <PropertyStatusInfo>
        <Time>1517553572362</Time>
        <Name>doublePropertyName</Name>
        <DataType>double</DataType>
        <Identifier>doubleProperty</Identifier>
        <Value>50.0</Value>
        <Unit>C</Unit>
      </PropertyStatusInfo>
    </List>
  </Data>
</QueryDevicePropertyStatusResponse>
```

2.8.23 SetDeviceProperty

调用该接口为指定设备设置属性值。

说明

因为云端下发属性设置命令和设备收到并执行该命令是异步的，所以调用该接口时，返回的成功结果只表示云端下发属性设置的请求成功，不能保证设备端收到并执行了该请求。需设备端SDK成功响应云端设置设备属性值的请求，设备属性值才能真正设置成功。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：SetDeviceProperty。
IotId	String	否	要设置属性值的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。

名称	类型	是否必需	描述
ProductKey	String	否	要设置属性值的设备所隶属的产品Key。  说明: 如果传入该参数, 需同时传入DeviceName
DeviceName	String	否	要设置属性值的设备的名称。  说明: 如果传入该参数, 需同时传入ProductKey
Items	String	是	要设置的属性信息, 组成为key:value, 数据格式为 JSON String。 若此参数为空时, 需传入 Items={}。 详情参见下表Items。
公共请求参数	-	是	请参见 公共参数 。

表 2-42: Items

名称	类型	描述
key	String	要设置的属性的标识符 (identifier)。设备的属性 Identifier, 可在控制台中设备所属的产品的功能定义中查看。  说明: 设置的属性必需是读写型。如果您指定了一个只读型的属性, 设置将会失败。
value	Obejct	属性值。取值需和您定义的属性的数据类型和取值范围保持一致。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。

名称	类型	描述
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的数据。详情请参见下表Data。

表 2-43: Data

名称	类型	描述
MessageId	String	云端给设备下发属性设置的消息ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=SetDeviceProperty
&ProductKey=a1*****
&DeviceName=device1
&Items=%7B%22LightAdjustLevel%22%3A1%7D
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "MessageId": "abcabc123"
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<SetDevicePropertyResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <MessageId>abcabc123</MessageId>
  </Data>
</SetDevicePropertyResponse>
```

```
</SetDevicePropertyResponse>
```

2.8.24 SetDevicesProperty

调用该接口批量设置设备属性值。

使用限制

单账号的每秒请求数最大限制为50 QPS。



说明:

子账号共享主账号配额。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：SetDevicesProperty。
ProductKey	String	是	要设置属性值的设备所隶属的产品Key。
DeviceNames	List<String>	是	要设置属性值的设备名称列表。目前，最多支持500个设备。
Items	String	是	要设置的属性信息，组成为key:value，数据格式为JSON String。 若此参数为空时，需传入Items={}。 Items数据详情，参见下表Items。
公共请求参数	-	是	请参见 公共参数 。

表 2-44: Items

名称	类型	描述
key	String	要设置的属性的标识符（identifier）。设备的属性Identifier，可在控制台中设备所属的产品的功能定义中查看。  说明: 设置的属性必需是读写型。如果您指定了一个只读型的属性，设置将会失败。

名称	类型	描述
value	Object	属性值。取值需和您定义的属性的数据类型和取值范围保持一致。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/&Action=SetDevicesProperty
&DeviceName.1=1102andriod02
&DeviceName.2=1102android01
&Items=%7B%20%20%20%20%20%20%22Data%22%3A%221372060916%22%2C%20%20%20%20%
20%22Status%22%3A1%20%7D
&ProductKey=a1hWjHDWUbF
&公共请求参数
```

返回示例

JSON 格式

```
{
  "RequestId": "2E19BDAF-0FD0-4608-9F41-82D230CFEE38",
  "Success": true
}
```

XML 格式

```
<?xml version='1.0' encoding='utf-8'?>
<SetDevicesPropertyResponse>
  <RequestId>"2E19BDAF-0FD0-4608-9F41-82D230CFEE38"</RequestId>
  <Success>true</Success>
</SetDevicesPropertyResponse>
```

</SetDevicesPropertyResponse>

2.8.25 QueryDeviceProp

调用该接口查询指定设备的标签列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceProp。
IotId	String	否	物联网平台为该设备颁发的设备ID，作为该设备的唯一标识符。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。

名称	类型	描述
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Props	String	调用成功时，返回的设备标签信息列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceProp
&ProductKey=al*****
&DeviceName=device1
&公共请求参数
```

返回示例

· JSON 格式

```
{
  "RequestId": "D8CEF5E7-1484-4164-A242-C06BA3A54E0F",
  "Props": "{\"coordinate\":\"104.07086:30.549169\"}",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryDevicePropResponse>
  <RequestId>D8CEF5E7-1484-4164-A242-C06BA3A54E0F</RequestId>
  <Props>
    <coordinate>104.07086:30.549169</coordinate>
  </Props>
  <Success>true</Success>
</QueryDevicePropResponse>
```

2.8.26 SaveDeviceProp

调用该接口为指定设备设置标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：SaveDeviceProp。
ProductKey	String	否	要创建标签的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。

名称	类型	是否必需	描述
DeviceName	String	否	要创建标签的设备名称。  说明: 如果传入该参数, 需同时传入ProductKey。
IotId	String	否	设备ID, 物联网平台为设备颁发的唯一标识。  说明: 如果传入该参数, 则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符, 与ProductKey & DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合, 则以IotId为准。
Props	String	是	要设置的设备标签, 每个标签的具体结构参见Prop。  说明: · 设备标签是JSON格式, Props是对应的String格式。Prop由标签key和value组成, key和value均不能为空。如, Props = {"color": "red"}。 · Props的总大小不超过5K。 · 单个设备的设备标签总数不超过100个。 · 单次修改或新增的设备标签数不超过100个。
公共请求参数	-	是	请参见 公共参数 。

表 2-45: Prop

名称	类型	是否必需	描述
key	String	是	设置标签键值。标签键值可包含英文字母，数字和点号（.），长度在2-30字符之间。 如果设置的键值已存在，则将覆盖之前的标签值。 如果设置的键值不存在，则新增标签。
value	Object	是	标签值。可包含中文、英文字母、数字、下划线（_）、连字符（-）和点号（.）。长度不可超过128字符。一个中文汉字算2字符。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=SaveDeviceProp
&ProductKey=a1*****
&DeviceName=device1
&Props=%7B%22color%22%3A%22red%22%7D
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
}
```

```
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<SaveDevicePropResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</SaveDevicePropResponse>
```

2.8.27 DeleteDeviceProp

调用该接口删除设备下的指定标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteDeviceProp。
IotId	String	否	要删除标签的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey & DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要删除标签的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要删除标签的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。

名称	类型	是否必需	描述
PropKey	String	是	要删除的设备标签键值（Key）。  说明： 物联网平台在目标设备的标签中检索您提供的Key值，并删除与之对应的标签。如果目标设备的标签中没有与您提供的Key值对应的记录，则不执行任何操作。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteDeviceProp
&ProductKey=al*****
&DeviceName=device1
&PropKey=temperature
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<DeleteDevicePropResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
</DeleteDevicePropResponse>
```

```
</DeleteDevicePropResponse>
```

2.8.28 QueryDeviceByTags

调用该接口通过标签查询设备。

限制说明

一次调用，最多可输入10个标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值 QueryDeviceByTags。
PageSize	Integer	否	指定返回结果中每页显示的记录数量，最大值是50。默认值是10。
CurrentPage	Integer	否	指定从返回结果中的第几页开始显示。默认值是1。
Tag	List<Tag>	是	设备标签。数量限制：可输入最多10个标签。 Tag 包括TagKey和TagValue，分别对应标签的key和value。 支持只传入TagKey进行查询。 详情请参见下表 Tag。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

表 2-46: Tag

名称	类型	是否必需	描述
TagKey	String	是	设备标签的key。
TagValue	String	否	设备标签的value。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。

名称	类型	描述
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
PageCount	Integer	总页数。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页面号。
Total	Integer	总记录数。
Data	Data	调用成功时，返回的设备信息列表。详情参见下表 SimpleDeviceInfo。

表 2-47: SimpleDeviceInfo

名称	类型	描述
ProductName	String	产品名称。
ProductKey	String	产品Key。
DeviceName	String	设备名称。
IotId	String	物联网平台为该设备颁发的ID，作为该设备的唯一标识符。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceByTags
&CurrentPage=1
&PageSize=10
&Tag.1.TagKey=dfdfd
&Tag.1.TagValue=dfdfd
&Tag.2.TagKey=dddfd
&Tag.2.TagValue=dfdfd
&公共请求参数
```

返回示例

· JSON格式

```
{
  "PageCount": 1,
  "Data": {
    "SimpleDeviceInfo": [
      {
        "DeviceName": "1102jichu02",
        "ProductKey": "a1SM5S1shy1",
        "IotId": "GookTiUcwqRbHosp9Ta10017d3a00",
        "ProductName": "TEST"
```

```
    }  
  ]  
},  
"PageSize": 10,  
"Page": 1,  
"RequestId": "2B5091E4-32D5-4884-A5B2-2E8E713D84AF",  
"Success": true,  
"Total": 1  
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>  
<QueryDeviceByTags>  
  <PageCount>1</PageCount>  
  <Data>  
    <SimpleDeviceInfo>  
      <DeviceName>1102jichu02</DeviceName>  
      <ProductKey>a1SM5S1shy1</ProductKey>  
      <IotId>GookTiUcwqRbHosp9Ta10017d3a00</IotId>  
      <ProductName>TEST</ProductName>  
    </SimpleDeviceInfo>  
  </Data>  
  <PageSize>10</PageSize>  
  <Page>1</Page>  
  <RequestId>2B5091E4-32D5-4884-A5B2-2E8E713D84AF</RequestId>  
  <Success>true</Success>  
  <Total>1</Total>  
</QueryDeviceByTags>
```

2.8.29 GetThingTopo

调用该接口查询指定设备的拓扑关系。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetThingTopo。
IotId	String	否	要查询的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。

名称	类型	是否必需	描述
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey
PageSize	Integer	是	返回结果中每页显示的记录数量。数量限制：每页最多可显示50条记录。
PageNo	Integer	是	从返回结果中的第几页开始显示。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的设备信息集合。详情参见 DeviceDataInfo 。

表 2-48: DeviceDataInfo

名称	类型	描述
PageCount	Integer	总页数。
PageSize	Integer	每页显示的记录数。
List	List	设备信息集合。每个元素的结构参见 DeviceInfo 。
CurrentPage	Integer	当前页面号。

名称	类型	描述
Total	Integer	总记录数。

表 2-49: DeviceInfo

名称	类型	描述
IotId	String	设备的ID。
ProductKey	String	设备所隶属的产品Key。
DeviceName	String	设备名称。
DeviceSecret	String	设备密钥。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetThingTopo
&ProductKey=al*****
&DeviceName=device1
&PageSize=10
&PageNo=1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true
  "Data": {
    "PageCount": 1,
    "PageSize": 10,
    "List": {
      "deviceInfo": [
        {
          "DeviceName": "device1",
          "ProductKey": "al*****",
          "DeviceSecret": "abc*****",
          "IotId": "SR8FiTu1R9tlUR2V1bmi0010*****"
        }
      ]
    },
    "CurrentPage": 1,
    "Total": 1
  },
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<GetThingTopoResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <PageCount>1</PageCount>
    <PageSize>10</PageSize>
    <List>
      <DeviceInfo>
        <DeviceName>device1</DeviceName>
        <ProductKey>al*****</ProductKey>
        <DeviceSecret>abc*****</DeviceSecret>
        <IotId>SR8FiTu1R9tLUR2V1bmi0010*****</IotId>
      </DeviceInfo>
    </List>
    <CurrentPage>1</CurrentPage>
    <Total>1</Total>
  </Data>
</GetThingTopoResponse>
```

2.8.30 NotifyAddThingTopo

调用该接口通知网关设备增加拓扑关系。

限制说明

返回的成功结果只表示添加拓扑关系的指令成功下发给网关，但并不表示网关成功添加拓扑关系。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：NotifyAddThingTopo。
GwIotId	String	否	要通知的网关设备ID，即网关类型设备的IotId。  说明： 如果传入该参数，则无需传入GwProductKey和GwDeviceName。GwIotId作为设备唯一标识符，与GwProductKey和GwDeviceName组合是一一对应的关系。如果您同时传入GwIotId和GwProductKey与GwDeviceName组合，则以GwIotId为准。
GwProductKey	String	否	要通知的网关设备所隶属的产品Key，即网关类型产品的ProductKey。  说明： 如果传入该参数，需同时传入GwDeviceName。

名称	类型	是否必需	描述
GwDeviceName	String	否	要通知的网关设备的名称，即网关类型设备的DeviceName。  说明： 如果传入该参数，需同时传入GwProductKey。
DeviceListStr	List	是	要挂载在目标网关设备上的子设备数组，为JSON字符串形式。具体结构参见下表DeviceList。
公共请求参数	-	是	请参见 公共参数 。

表 2-50: DeviceList

名称	类型	是否必需	描述
ProductKey	String	否	子设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	子设备名称。  说明： 如果传入该参数，需同时传入ProductKey。
IotId	String	否	子设备ID，物联网平台颁发给设备的唯一标识。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey & DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的数据。详情请见下表Data。

表 2-51: Data

名称	类型	描述
MessageId	String	云端向网关设备下发增加拓扑关系的消息ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=NotifyAddThingTopo
&GwProductKey=aldnfald7a
&GwDeviceName=gateway
&DeviceListStr=[{"productKey":"alabcabcab","deviceName":"device1"}, {"
IotId":"edAjkIeBSsdfadjjllja***"}]
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "419A3FC1-B517-4958-9414-5546765FA51F",
  "Success": true,
  "Data": {
    "MessageId": "abccabc123"
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
<NotifyAddThingTopoResponse>
  <RequestId>419A3FC1-B517-4958-9414-5546765FA51F</RequestId>
  <Success>true</Success>
  <Data>
    <MessageId>abccabc123</MessageId>
  </Data>
```

</NotifyAddThingTopoResponse>

2.8.31 RemoveThingTopo

调用该接口移除网关与子设备的拓扑关系。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：RemoveThingTopo。
IotId	String	否	要删除拓扑关系的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要删除拓扑关系的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要删除拓扑关系的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。

名称	类型	描述
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Boolean	删除拓扑关系的结果。true表示删除成功，false表示删除失败。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=RemoveThingTopo
&IotId=CqXL5h5ysRTA4NxjABjj0010fa****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "Data": true,
  "RequestId": "098BEEF4-58F4-4181-A891-5D37DB6C4C3B",
  "Success": true
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
<RemoveThingTopoResponse>
  <RequestId>098BEEF4-58F4-4181-A891-5D37DB6C4C</RequestId>
  <Success>true</Success>
  <Data>true</Data>
</RemoveThingTopoResponse>
```

2.8.32 QueryDeviceStatistics

调用该接口查询设备统计数据。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceStatistics。
ProductKey	String	否	要查询的设备所隶属的产品Key。 - 传入产品Key，将返回该产品下的设备统计数据。 - 不传入该参数，则返回账号下所有设备统计数据。

名称	类型	是否必需	描述
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的设备统计信息。详情参见 StatisticsInfo 。

表 2-52: StatisticsInfo

名称	类型	描述
deviceCount	Integer	设备总数。
onlineCount	Integer	在线的设备数量。
activeCount	Integer	已激活的设备数量。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceStatistics
&ProductKey=a1*****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "deviceCount": 100,
    "onlineCount": 10,
    "activeCount": 50
  }
}
```

```
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<QueryDeviceStatisticsResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <deviceCount>100</deviceCount>
    <onlineCount>10</onlineCount>
    <activeCount>50</activeCount>
  </Data>
</QueryDeviceStatisticsResponse>
```

2.8.33 SetDeviceDesiredProperty

调用该接口为指定设备批量设置期望属性值。

限制说明

- 只读属性不支持设置期望属性值。
- 一次调用最多可设置10个期望属性值。
- 设备创建后，期望属性值的版本（Version）为0。首次设置期望属性值时，如果指定Version参数，则需指定Version值为0。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：SetDeviceDesiredProperty。
IotId	String	否	要设置期望属性值的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey和DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要设置期望属性值的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。

名称	类型	是否必需	描述
DeviceName	String	否	要设置期望属性值的设备的名称。  说明: 如果传入该参数, 需同时传入ProductKey。
Items	String	是	要设置的期望属性值, 组成为key:value, 数据格式为 JSON String, 如{"Power":1}。 详情参见下表Items。  说明: 最多可输入10个期望属性值。
Versions	String	否	当前期望属性值版本, 组成为key:value, 数据格式为 JSON String, 如{"Power":2}。 详情参见下表Versions。  说明: 如果传入的版本号与当前版本不符, 服务器将拒绝此次请求。若您不确定当前期望值的版本号, 可以不传入版本号, 但仍需传入有效的JSON, 即传入{}。
公共请求参数	-	是	请参见 公共参数 。

表 2-53: Items

名称	类型	描述
key	String	取值为属性的标识符 (identifier)。可在控制台中, 设备所属产品的功能定义中查看属性的identifier。  说明: - 指定属性必须是读写型。如果您指定了一个只读型的属性, 设置将会失败。 - 一次调用中, key的取值 (即属性的identifier) 不能重复。

名称	类型	描述
value	Object	要设置的期望属性值。取值需符合您为该属性定义的数据类型和取值范围。  说明: 若属性值设置为null, 则表示清空期望属性值。

表 2-54: Versions

名称	类型	描述
key	String	取值为属性的标识符（identifier）。可在控制台中，设备所属产品的功能定义中查看属性的identifier。  说明: 一次调用中，key的取值（即属性的identifier）不能重复。
value	Long	当前期望属性值的版本号。 首次设置期望属性值时，指定该参数值为0。首次设置期望属性值后，期望值版本号为1。以后每次设置期望值后，物联网平台自动将期望值版本加1（即第二次设置期望属性值时，指定该参数值为1。设置成功后，版本号自动变为2；第三次设置时，指定该参数值为2。设置成功后，版本号自动变为3；以此类推）。  说明: 若传入的版本号与当前版本号不符，请求将失败。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

名称	类型	描述
Data	Data	调用成功时，返回的数据。详情请参见下表Data。

表 2-55: Data

名称	类型	描述
MessageId	String	云端下发给设备的设置期望属性值的消息ID。
Version	String	本次设置期望属性值后，期望属性值的当前版本号。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=SetDeviceDesiredProperty
&ProductKey=a1*****
&DeviceName=device1
&Items=%7B%22LightAdjustLevel%22%3A1%7D
&Versions=%7B%22LightAdjustLevel%22%3A10%7D
&公共请求参数
```

返回示例

· JSON格式

```
{
  "Data": {
    "MessageId": "300511751",
    "Versions": "{\"Power\":2}"
  },
  "RequestId": "AADE79D2-B328-4FC6-A3E0-34BB23BCA440",
  "Success": true
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<SetDeviceDesiredPropertyResponse>
  <RequestId>AADE79D2-B328-4FC6-A3E0-34BB23BCA440</RequestId>
  <Success>true</Success>
  <Data>
    <MessageId>300511751</MessageId>
    <Versions>{"Power":2}</Versions>
  </Data>
</SetDeviceDesiredPropertyResponse>
```

2.8.34 QueryDeviceDesiredProperty

调用该接口查询指定设备的期望属性值。

限制说明

- 只读属性不支持查询期望属性值。

- 一次调用最多能查询10个属性的期望值。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceDesiredProperty。
IotId	String	否	要查询的设备ID。  说明： 如果传入该参数，则无需传入ProductKey和DeviceName。IotId作为设备唯一标识符，与ProductKey & DeviceName组合是一一对应的关系。如果您同时传入IotId和ProductKey与DeviceName组合，则以IotId为准。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入DeviceName。
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入ProductKey。
Identifiers	List	否	要查询期望值的属性的标识符 (identifier) 列表。 设备的属性identifier，可在控制台中，设备所属产品的功能定义中查看。  说明： · 单次调用，最多能传入10个identifier。 · 不可输入重复的属性identifier。 · 若不传入此参数，将返回该设备所有属性的期望值。

名称	类型	是否必需	描述
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的数据。详情参见下表。

表 2-56: Data

名称	类型	描述
List	List<DesiredPropertyInfo>	返回的属性信息。详情参见下表DesiredPropertyInfo。

表 2-57: DesiredPropertyInfo

名称	类型	描述
Identifier	String	属性标识符。
Name	String	属性名称。
DataType	String	属性数据类型。
Time	String	期望属性值的修改时间，单位是毫秒。
Value	String	期望属性值。
Unit	String	属性单位。
Version	Long	当前期望属性值的版本号。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceDesiredProperty
&IotId=SR8FiTu1R9tlUR2V1bmi0010*****
&Identifier.1=Weight
&Identifier.2=Circle
&公共请求参数
```

返回示例

· JSON 格式

```
{
  "RequestId": "57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true,
  "Data": {
    "List": [
      {
        "DesiredPropertyInfo": [
          {
            "Name": "WeightName",
            "Value": "50.0",
            "Time": "1517553572362",
            "DataType": "double",
            "Identifier": "Weight",
            "Unit": "C",
            "Version": 10
          },
          {
            "Name": "CircleName",
            "Value": "50.0",
            "Time": "1517553572362",
            "DataType": "double",
            "Identifier": "Circle",
            "Unit": "C",
            "Version": 10
          }
        ]
      }
    ]
  }
}
```

· XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<QueryDeviceDesiredPropertyResponse>
  <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
  <Success>true</Success>
  <Data>
    <List>
      <DesiredPropertyInfo>
        <Time>1517553572362</Time>
        <Name>doublePropertyName</Name>
        <DataType>double</DataType>
        <Identifier>doubleProperty</Identifier>
        <Value>50.0</Value>
        <Unit>C</Unit>
        <Version>10</Version>
      </DesiredPropertyInfo>
    </List>
  </Data>
</QueryDeviceDesiredPropertyResponse>
```

```
<DesiredPropertyInfo>
  <Time>1517553572362</Time>
  <Name>CircleName</Name>
  <DataType>double</DataType>
  <Identifier>Circle</Identifier>
  <Value>50.0</Value>
  <Unit>C</Unit>
  <Version>10</Version>
</DesiredPropertyInfo>
</List>
</Data>
</QueryDeviceDesiredPropertyResponse>
```

2.8.35 QueryLoRaJoinPermissions

调用该接口查询LoRaWAN入网凭证列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryLoRaJoinPermissions。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。
JoinPermissions	List<JoinPermission>	调用成功时，返回的入网凭证数据。详情见下表JoinPermission。

表 2-58: JoinPermission

名称	类型	描述
JoinPermissionId	String	入网凭证ID，入网凭证的唯一标识。
OwnerAliyunPk	String	入网凭证创建者的阿里云账号ID。
JoinPermissionName	String	入网凭证名称。

名称	类型	描述
FreqBandPlanGroupId	String	入网凭证采用的频谱计划ID。取值： · CN470：同频 · CN470：异频 · AS923：同频
ClassMode	String	入网凭证采用的通信模式。取值： · A：终端设备允许双向通信。 · B：终端设备会在预设时间中开放接收窗口。 · C：终端设备持续开放接收窗口，只在传输时关闭。
JoinPermissionType	String	入网凭证的类型。取值： · LOCAL：专用凭证。 · ROAMING：漫游凭证。
Enabled	Boolean	入网凭证的启停用状态。取值： · true：启用 · false：停用

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryLoRaJoinPermissions
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "A6B85BE6-D199-4634-AC24-AD08CD289D9B",
  "Success": true,
  "JoinPermissions": {
    "JoinPermission": [
      {
        "Enabled": true,
        "JoinPermissionType": "LOCAL",
        "JoinPermissionId": "8***",
        "OwnerAliyunPk": "1375364789****",
        "ClassMode": "B",
        "JoinPermissionName": "ForTest"
      }
    ]
  }
}
```


}

- XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryLoRaJoinPermissions>
  <RequestId>A6B85BE6-D199-4634-AC24-AD08CD289D9B</RequestId>
  <Success>true</Success>
  <JoinPermissions>
    <JoinPermission>
      <Enabled>true</Enabled>
      <JoinPermissionType>LOCAL</JoinPermissionType>
      <JoinPermissionId>8***</JoinPermissionId>
      <OwnerAliyunPk>1375364789****</OwnerAliyunPk>
      <ClassMode>B</ClassMode>
      <JoinPermissionName>ForTest</JoinPermissionName>
    </JoinPermission>
  </JoinPermissions>
</QueryLoRaJoinPermissions>
```

2.8.36 CreateLoRaNodesTask

调用该接口生成批量注册LoRaWAN设备的任务。

使用限制

单次调用最多可添加500个设备。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：CreateLoRaNodesTask。
ProductKey	String	是	设备所隶属的产品Key。
DeviceInfos	List<DeviceInfo>	是	LoRaWAN设备信息列表。详情参见下表DeviceInfo。  说明： 单次调用最多可添加500个设备的信息。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

表 2-59: DeviceInfo

名称	类型	是否必需	描述
DevEui	String	是	LoRaWAN设备的DevEUI，其全球唯一标识。
PinCode	String	是	LoRaWAN设备的PIN Code，用于校验DevEUI的合法性。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。
TaskId	String	调用成功时，返回的任务ID。  说明： 请妥善保管该ID。查询设备创建的状态时需使用该ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=CreateLoRaNodesTask
&DeviceInfo.1.DevEui=d896e0fffffet****
&DeviceInfo.1.PinCode=562959
&DeviceInfo.2.DevEui=d896e0fffffer****
&DeviceInfo.2.PinCode=573091
&ProductKey=a1HfDI4****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "38D5FDA5-19B9-445D-8713-213B743266DE",
  "TaskId": "62146",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<CreateLoRaNodesTask>
  <RequestId>38D5FDA5-19B9-445D-8713-213B743266DE</RequestId>
  <TaskId>62146</TaskId>
  <Success>true</Success>
</CreateLoRaNodesTask>
```

</CreateLoRaNodesTask>

2.8.37 GetLoraNodesTask

调用该接口查询批量注册LoRaWAN设备任务的状态。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetLoraNodesTask。
TaskId	String	是	注册LoRaWAN设备任务的ID，即调用 CreateLoRaNodesTask 创建任务后，返回的TaskId。
公共请求参数	-	是	公共请求参数，请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。
TaskId	String	调用成功时，返回的任务ID。
TotalCount	Long	任务中待注册设备的总数。
SuccessCount	Long	注册成功的数量。
FailedCount	Long	注册失败的数量。
SuccessDevEuis	List<String>	注册成功的设备的DevEUI列表。
TaskState	String	任务执行状态。 · RUNNING：任务正在执行中。 · FINISH：任务已执行完毕。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetLoraNodesTask
&TaskId=62146
```

&公共请求参数

返回示例

· JSON格式

```
{
  "TotalCount": 2,
  "SuccessCount": 2,
  "RequestId": "D5C7AC10-97A4-42EA-8F92-CDBCCC02DDAC",
  "SuccessDevEuis": {
    "SuccessDevEui": [
      "d896e0ffff01****",
      "d896e0ffff01****"
    ]
  },
  "TaskId": "62146",
  "TaskState": "FINISH",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<GetLoraNodesTask>
  <TotalCount>2</TotalCount>
  <SuccessCount>2</SuccessCount>
  <RequestId>D5C7AC10-97A4-42EA-8F92-CDBCCC02DDAC</RequestId>
  <SuccessDevEuis>
    <SuccessDevEui>d896e0ffff01****</SuccessDevEui>
    <SuccessDevEui>d896e0ffff01****</SuccessDevEui>
  </SuccessDevEuis>
  <TaskId>62146</TaskId>
  <TaskState>FINISH</TaskState>
  <Success>true</Success>
</GetLoraNodesTask>
```

2.8.38 QueryDeviceFileList

调用该接口查询指定设备上传到物联网平台的所有文件列表。

说明

调用该接口返回的文件信息中，不包括文件下载地址。如需获取文件下载地址，请调用[QueryDeviceFile](#)查询。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceFileList。

名称	类型	是否必需	描述
IotId	String	否	要查询的设备ID，设备唯一标识符。  说明： 如果传入该参数，则无需传入 ProductKey和 DeviceName。IotId作为设备唯一标识符，和 ProductKey与 DeviceName组合是一一对应的关系。如果您同时传入 IotId和 ProductKey与 DeviceName组合，则以 IotId为准。不能传入空参数。
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入 DeviceName。
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入 ProductKey。
PageSize	Integer	否	返回结果中每页显示的文件记录数量。最大取值200，默认值是10。
CurrentPage	Integer	否	显示返回结果中的第几页。最小取值1，默认值1。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	List<FileStoreSummary>	调用成功时，返回的文件信息列表。详情参见下表 FileStoreSummary 。

表 2-60: FileStoreSummary

名称	类型	描述
FileId	String	文件标识符。
Name	String	文件名。
Size	String	文件大小。
UtcCreatedOn	String	文件创建时间。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceFileList
&ProductKey=a1*****
&DeviceName=deviceName1
&PageSize=10
&CurrentPage=1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "PageCount": 1,
  "Data": {
    "FileStoreSummary": [
      {
        "Name": "testFile2.txt",
        "FileId": "xL0G67MBLBDtkR7GCfT*****",
        "UtcCreatedOn": "2019-03-21T08:45:42.000Z",
        "Size": "102400"
      },
      {
        "Name": "testFile3.txt",
        "FileId": "6UCo1SqbqnQEoh9aKqD*****",
        "UtcCreatedOn": "2019-03-21T08:45:42.000Z",
        "Size": "102400"
      },
      {
        "Name": "testFile1.txt",
        "FileId": "IhXXww3Eeu6uzS0SCyu*****",
        "UtcCreatedOn": "2019-03-21T08:45:40.000Z",
        "Size": "102400"
      }
    ]
  },
  "PageSize": 10,
  "RequestId": "7C7BA526-826D-46AA-A45E-55D21E6D1583",
  "CurrentPage": 1,
  "Success": true,
  "Total": 3
}
```

}

· XML格式

```
<?xml version="1.0" encoding="utf-8"?>
<QueryDeviceFileListResponse>
  <PageCount>1</PageCount>
  <Data>
    <FileStoreSummary>
      <Name>testFile2.txt</Name>
      <FileId>xL0G67MBLBDtkR7GCfT*****</FileId>
      <UtcCreatedOn>2019-03-21T08:45:42.000Z</UtcCreatedOn>
      <Size>102400</Size>
    </FileStoreSummary>
    <FileStoreSummary>
      <Name>testFile3.txt</Name>
      <FileId>6UCo1SqbnQEoh9aKqD*****</FileId>
      <UtcCreatedOn>2019-03-21T08:45:42.000Z</UtcCreatedOn>
      <Size>102400</Size>
    </FileStoreSummary>
    <FileStoreSummary>
      <Name>testFile1.txt</Name>
      <FileId>IhXXww3Eeu6uzS0SCyu*****</FileId>
      <UtcCreatedOn>2019-03-21T08:45:40.000Z</UtcCreatedOn>
      <Size>102400</Size>
    </FileStoreSummary>
  </Data>
  <PageSize>10</PageSize>
  <RequestId>BF06F7FD-B199-4B90-9128-8416F975135E</RequestId>
  <CurrentPage>1</CurrentPage>
  <Success>true</Success>
  <Total>3</Total>
</QueryDeviceFileListResponse>
```

2.8.39 QueryDeviceFile

调用该接口查询指定设备上传到物联网平台的指定文件信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceFile。
IotId	String	否	要查询的设备ID，设备唯一标识符。  说明： 如果传入该参数，则无需传入 ProductKey 和 DeviceName。IotId 作为设备唯一标识符，和 ProductKey 与 DeviceName 组合是一一对应的关系。如果您同时传入 IotId 和 ProductKey 与 DeviceName 组合，则以 IotId 为准。不能传入空参数。

名称	类型	是否必需	描述
ProductKey	String	否	要查询的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入 DeviceName。
DeviceName	String	否	要查询的设备的名称。  说明： 如果传入该参数，需同时传入 ProductKey。
FileId	String	是	文件标识符。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	Data	调用成功时，返回的文件信息。详情参见下表 Data 。

表 2-61: Data

名称	类型	描述
FileId	String	文件标识符。
Name	String	文件名。
Size	String	文件大小。
UtcCreatedOn	String	文件创建时间。
DownloadUrl	String	文件下载URL。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceFile
&ProductKey=a1*****
&DeviceName=deviceName1
&FileId=6UCo1SqbqnQEoh9aKqD*****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "93C5276D-5C8A-40D9-BFD6-4BD5B8C1A08F",
  "Data": {
    "Name": "testFile3.txt",
    "DownloadUrl": "http://iotx-file-store.oss-cn-shanghai.aliyuncs
.com/device_file/A849E4E5CFF64804A18D9384AC9D****/aGEKIpp5NA
GxdP2oo90000****/testFile3.txt?Expires=1553162075&OSSAccessKeyId=
LTAIYLScbHiV****&Signature=%2F88xdEFPukJ****%2F8****%2Bdv3io%3D",
    "FileId": "6UCo1SqbqnQEoh9aKqDQ01****",
    "UtcCreatedOn": "2019-03-21T08:45:42.000Z",
    "Size": "102400"
  },
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<RequestId>93C5276D-5C8A-40D9-BFD6-4BD5B8C1A08F</RequestId>
<Data>
  <Name>testFile3.txt</Name>
  <DownloadUrl>http://iotx-file-store.oss-cn-shanghai.aliyuncs
.com/device_file/A849E4E5CFF64804A18D9384AC9D****/aGEKIpp5NA
GxdP2oo90000****/testFile3.txt?Expires=1553162075&OSSAccessKeyId
=LTAIYLScbHiV****&Signature=%2F88xdEFPukJ****%2F8****%2Bdv3io%3D
</DownloadUrl>
  <FileId>6UCo1SqbqnQEoh9aKqDQ01****</FileId>
  <UtcCreatedOn>2019-03-21T08:45:42.000Z</UtcCreatedOn>
  <Size>102400</Size>
</Data>
<Success>true</Success>
```

2.8.40 DeleteDeviceFile

调用该接口删除指定设备上传到物联网平台的指定文件。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteDeviceFile。

名称	类型	是否必需	描述
IotId	String	否	文件所属的设备ID，设备唯一标识符。  说明： 如果传入该参数，则无需传入 ProductKey和 DeviceName。IotId作为设备唯一标识符，和 ProductKey与 DeviceName组合是一一对应的关系。如果您同时传入 IotId和 ProductKey与 DeviceName组合，则以 IotId为准。不能传入空参数。
ProductKey	String	否	文件所属的设备所隶属的产品Key。  说明： 如果传入该参数，需同时传入 DeviceName。
DeviceName	String	否	文件所属的设备的名称。  说明： 如果传入该参数，需同时传入 ProductKey。
FileId	String	是	文件标识符。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteDeviceFile
&ProductKey=al*****
&DeviceName=deviceName1
&FileId=6UCo1SqbqnQEoh9aK*****
```

&公共请求参数

返回示例

· JSON格式

```
{
  "RequestId": "C25255EA-9F6E-4E04-85FE-AC1E9C136EBB",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="utf-8"?>
<DeleteDeviceFileResponse>
  <RequestId>C25255EA-9F6E-4E04-85FE-AC1E9C136EBB</RequestId>
  <Success>true</Success>
</DeleteDeviceFileResponse>
```

2.9 分组管理

2.9.1 CreateDeviceGroup

调用该接口新建分组。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：CreateDeviceGroup。
SuperGroupId	String	否	父组ID。 若要创建的是一级分组，则不传入此参数。
GroupName	String	是	分组名称。名称可包含中文汉字、英文字母、数字和下划线（_）。长度范围 4 - 30 字符（一个中文汉字占二个字符）。
GroupDesc	String	否	分组描述。长度限制为100字符（一个中文汉字占一个字符）。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。

名称	类型	描述
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。
Data	Data	请参考下表Group。

表 2-62: Group

名称	类型	描述
GroupName	String	分组名称。
UtcCreate	Date	创建时间。
GroupDesc	String	分组描述。
GroupId	String	分组ID，系统为分组生成的全局唯一标识符。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=CreateDeviceGroup
&GroupDesc=Group test
&GroupName=groupptest
&公共请求参数
```

返回示例

```
{
  "Data":{
    "GroupDesc":"Group test",
    "GroupName":"groupptest",
    "UtcCreate":"2018-10-17T11:19:31.000Z",
    "GroupId":"HtMLECKbdJQLIcbY"
  },
  "RequestId":"4D6D7F71-1C94-4160-8511-EFF4B8F0634D",
  "Success":true
}
```

2.9.2 UpdateDeviceGroup

调用该接口修改分组信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：UpdateDeviceGroup。
GroupId	String	是	分组ID。分组的全局唯一标识符。

名称	类型	是否必需	描述
GroupDesc	String	否	修改后的分组描述。长度限制为100字符（一个中文汉字占一个字符）。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=UpdateDeviceGroup
&GroupId=W16X8TvdosecZu91
&GroupDesc=test2
&公共请求参数
```

返回示例

```
{
  "RequestId": "B78B4FD1-AE89-417B-AD55-367EBB0C6759",
  "Success": true
}
```

2.9.3 DeleteDeviceGroup

调用该接口删除指定分组。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：DeleteDeviceGroup。
GroupId	String	是	分组ID，分组的全局唯一标识符。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=DeleteDeviceGroup
&GroupId=tDQvBJqbUyHskDse
&公共请求参数
```

返回示例

```
{
  "RequestId": "B78B4FD1-AE89-417B-AD55-367EBB0C6759",
  "Success": true
}
```

2.9.4 BatchAddDeviceGroupRelations

调用该接口添加设备到某一分组（可批量添加设备）。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：BatchAddDeviceGroupRelations。
GroupId	String	是	分组ID，分组的全局唯一标识符。
Devices	List<Device>	是	要添加到分组的设备信息，请参见下表Device。
公共请求参数	-	是	请参见 公共参数 。

表 2-63: Device

名称	类型	是否必需	描述
ProductKey	String	是	要添加到分组的设备所属的产品Key。
DeviceName	String	是	要添加到分组的设备名称。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示调用是否成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	返回的结果信息。
Code	String	返回错误码，请参见 错误码 。
ValidDeviceCount	Integer	请求参数中合法的设备数量。
SuccessAddedDeviceCount	Integer	成功添加到分组的设备数量。
ExceedMaxGroupDeviceCount	Integer	请求参数中，已经添加到10个或者10个以上分组的设备数量（一个设备最多添加到10个分组）。
AlreadyRelatedGroupDeviceCount	Integer	原已经添加到此分组的设备数量。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=BatchAddDeviceGroupRelations
&Device.1.DeviceName=ZHuPo6sZzv7p0zYhv31y
&Device.1.ProductKey=a1kORrKiwQj
&Device.2.DeviceName=rB4V9PDW2FCPmwuf2N3Y
&Device.2.ProductKey=a1kORrKiwQj
&GroupId=6VfheblG5iUerXep
&公共请求参数
```

返回示例

```
{
  "SuccessAddedDeviceCount":2,
  "ExceedTenGroupDeviceCount":0,
  "ErrorMessage":"2 devices have been added, and 0 devices failed to be added.",
  "ValidDeviceCount":2,
  "AlreadyRelatedGroupDeviceCount":0,
  "RequestId":"671D0F8F-FDC7-4B12-93FA-336C079C965A",
  "Success":true
}
```

```
}

```

2.9.5 BatchDeleteDeviceGroupRelations

调用该接口批量删除指定分组中的设备。（只删除设备与分组的关联关系，不会删除设备本身。）

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：BatchDeleteDeviceGroupRelations。
GroupId	String	是	分组ID，分组的全局唯一标识符。
Devices	List<Device>	是	要从分组中删除的设备信息，请参见下表Device。
公共请求参数	-	是	请参见 公共参数 。

表 2-64: Device

名称	类型	是否必需	描述
ProductKey	String	是	要从分组中删除的设备所属的产品Key。
DeviceName	String	是	要从分组中删除的设备名称。

返回参数

名称	类型	描述
ValidDeviceCount	Integer	请求参数中要删除的设备中，有效的设备数量（即可删除的设备数量）。
SuccessDeviceCount	Integer	成功从分组中删除的设备数量。
AlreadyRelatedGroupDeviceCount	Integer	删除前，已添加到此分组的设备数量。
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=BatchDeleteDeviceGroupRelations
&Device.1.DeviceName=ZHUPo6sZzv7p0zYhv31y
&Device.1.ProductKey=a1k0RrKiwQj
&Device.2.DeviceName=rB4V9PDW2FCPmwuf2N3Y
&Device.2.ProductKey=a1k0RrKiwQj
&GroupId=W16X8TvdosecZu91
&公共请求参数
```

返回示例

```
{
  "SuccessDeviceCount":2,
  "ValidDeviceCount":2
  "AlreadyRelatedGroupDeviceCount":2,
  "RequestId":"8739385E-143F-4389-B900-B7DF9174CE0D",
  "Success":true
}
```

2.9.6 QueryDeviceGroupInfo

调用该接口查询分组详情。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QueryDeviceGroupInfo。
GroupId	String	是	分组ID，分组的全局唯一标识符。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。
Data	Data	调用成功时，返回的分组详细信息数据。请参考下表GroupInfo。

表 2-65: GroupInfo

名称	类型	描述
GroupName	String	分组名称。
UtcCreate	String	创建时间。
DeviceOnline	Integer	在线设备数量。
DeviceActive	Integer	激活设备数量。
DeviceCount	Integer	设备总数。
GroupId	String	分组ID。
GroupDesc	String	分组描述。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceGroupInfo
&GroupId=tDQvBJqbUyHskDse
&公共请求参数
```

返回示例

```
{
  "Data":{
    "DeviceOnline":0,
    "DeviceActive":1,
    "GroupName":"yanglv",
    "DeviceCount":10,
    "UtcCreate":"2018-09-14T14:35:51.000Z",
    "GroupId":"tDQvBJqbUyHskDse"
  },
  "RequestId":"7411716B-A488-4EEB-9AA0-6DB05AD2491F",
  "Success":true
}
```

2.9.7 QueryDeviceGroupList

调用该接口分页查询分组列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QueryDeviceGroupList。

名称	类型	是否必需	描述
GroupName	String	否	分组名称。 · 传入分组名称，则根据名称进行查询。支持分组名称模糊查询。 · 若不传入此参数，则进行全量分组查询。
SuperGroupId	String	否	父组ID。查询某父组下的子分组列表时，需传入此参数。
PageSize	Integer	否	每页记录数。最大值是200。默认值是10。
CurrentPage	Integer	否	指定从返回结果中的第几页开始显示。默认值为1。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。
PageCount	Integer	总页数。
PageSize	Integer	每页记录数。
CurrentPage	Integer	当前页号。
Total	Integer	总记录数。
Data	Data	调用成功时，返回的分组信息。请参见下表GroupInfo。  说明： 返回的分组信息按照分组创建时间倒序排列。

表 2-66: GroupInfo

名称	类型	描述
GroupName	String	分组名称。
UtcCreate	Date	分组创建时间。
GroupDesc	String	分组描述。

名称	类型	描述
GroupId	String	分组ID。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceGroupList
&PageSize=10
&CurrentPage=1
&公共请求参数
```

返回示例

```
{
  "PageCount":3,
  "Data":{
    "GroupInfo":[
      {
        "GroupName":"test1",
        "UtcCreate":"2018-10-09T02:58:34.000Z",
        "GroupId":"Kzt9FD8wje8oW64y"
      },
      {
        "GroupName":"test2",
        "UtcCreate":"2018-10-09T02:56:40.000Z",
        "GroupId":"0ayrSQ3DSd7uXXXC"
      },
      {
        "GroupDesc":"Test",
        "GroupName":"test3",
        "UtcCreate":"2018-09-16T05:38:27.000Z",
        "GroupId":"oWXLlQeFZtzCNsAV"
      },
      {
        "GroupName":"ylv0915",
        "UtcCreate":"2018-09-15T04:51:56.000Z",
        "GroupId":"SfEiVapLPUjBVvSq"
      },
      {
        "GroupName":"ydlv",
        "UtcCreate":"2018-09-14T14:35:51.000Z",
        "GroupId":"z2S2h9NsDTZmH8MR"
      },
      {
        "GroupName":"ldh_group_3",
        "UtcCreate":"2018-09-14T12:26:20.000Z",
        "GroupId":"chn5fkjinXGc7pGl"
      },
      {
        "GroupDesc":"ddd",
        "GroupName":"ylvgisbim",
        "UtcCreate":"2018-09-14T11:41:20.000Z",
        "GroupId":"ncUZ8DjWYaB9Rx10"
      },
      {
        "GroupName":"abc",
        "UtcCreate":"2018-09-14T09:14:30.000Z",
        "GroupId":"zpdvwxxBdt4FljYf"
      }
    ]
  }
}
```

```
    },
    {
      "GroupName": "test11",
      "UtcCreate": "2018-09-14T07:22:39.000Z",
      "GroupId": "BTaudF16X2xKQBNa"
    },
    {
      "GroupName": "testy",
      "UtcCreate": "2018-09-14T01:58:06.000Z",
      "GroupId": "PrTm3VOeggPwRUKQ"
    }
  ]
},
"PageSize": 10,
"RequestId": "BEFCA316-D6C7-470C-81ED-1FF4FFD4AA0D",
"CurrentPage": 1,
"Success": true,
"Total": 24
}
```

2.9.8 SetDeviceGroupTags

调用该接口添加、更新、或删除分组标签。

限制说明

单个分组最多可有100个标签。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：SetDeviceGroupTags
GroupId	String	是	分组ID，分组的全局唯一标识符。
TagString	String	是	JSON格式的标签数据。TagString由标签的tagKey和tagValue组成，tagKey和tagValue均不能为空。多个标签以英文逗号间隔。如， [{"tagKey": "h1", "tagValue": "rr"}, {"tagKey": "7h", "tagValue": "rr"}]。请参见下表 TagString 。 若更新已有标签，新的标签value值将覆盖原来的值。 若要删除某个标签，则不传入该标签的key和value即可。
公共请求参数	-	是	请参见 公共参数 。

表 2-67: TagString

名称	类型	是否必需	描述
tagKey	String	是	标签键。可包含英文大小写字母，数字和点号(.), 长度在2-30字符之间。
tagValue	String	是	标签值。可包含中文、英文字母、数字、下划线(_)和连字符(-)。长度不可超过128字符。一个中文汉字算2字符。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=SetDeviceGroupTags
&GroupId=W16X8Tvdosec****
&TagString=[{"tagKey":"h1","tagValue":"rr"}, {"tagKey":"7h","tagValue":"rr"}]
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "12CFDAF1-99D9-42E0-8C2F-F281DA5E8953",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<SetDeviceGroupTagsResponse>
  <RequestId>12CFDAF1-99D9-42E0-8C2F-F281DA5E8953</RequestId>
  <Success>true</Success>
```

```
</SetDeviceGroupTagsResponse>
```

2.9.9 QueryDeviceGroupTagList

调用该接口查询分组标签列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QueryDeviceGroupTagList。
GroupId	String	是	分组ID，分组的全局唯一标识符。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。
Data	Data	调用成功时，返回的标签信息。请参见下表GroupTagInfo。

表 2-68: GroupTagInfo

名称	类型	描述
TagKey	String	标签键。
TagValue	String	标签值。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceGroupTagList
&GroupId=W16X8TvdosecZu91
&公共请求参数
```

返回示例

```
{
  "Data":{
```

```

    "GroupTagInfo": [
      {
        "TagValue": "bulb",
        "TagKey": "room1"
      }
    ],
    "RequestId": "214154FF-9D47-4E3F-AAAD-F4CE67F41060",
    "Success": true
  }

```

2.9.10 QueryDeviceGroupByDevice

调用该接口查询某一设备所在的分组列表。

说明

目前一个设备最多能被添加到10个分组中。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryDeviceGroupByDevice。
ProductKey	String	是	产品Key。
DeviceName	String	是	设备名称。
公共请求参数	-	是	请参见 公共参数

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
Code	String	调用失败时，返回的错误码。请参见 错误码 。
ErrorMessage	String	调用失败时，返回的错误信息。
GroupInfos	List<GroupInfo>	调用成功时，返回的分组信息。详情请参见下表GroupInfo。

表 2-69: GroupInfo

名称	类型	描述
GroupId	String	分组ID。

名称	类型	描述
GroupName	String	分组名称。
UtcCreate	String	分组的创建时间。
GroupDesc	String	分组描述。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceGroupByDevice
&DeviceName=test456
&ProductKey=a1SKk9K****
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "7941C8CD-7764-4A94-8CD9-E2762D4A73AC",
  "GroupInfos": {
    "GroupInfo": [
      {
        "GroupDesc": "father desc",
        "GroupName": "father1543152336554",
        "UtcCreate": "2018-11-25T13:25:37.000Z",
        "GroupId": "6a3FF2XE2BKayWsM"
      }
    ]
  },
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryDeviceGroupByDevice>
  <RequestId>7941C8CD-7764-4A94-8CD9-E2762D4A73AC</RequestId>
  <GroupInfos>
    <GroupInfo>
      <GroupDesc>father desc</GroupDesc>
      <GroupName>father1543152336554</GroupName>
      <UtcCreate>2018-11-25T13:25:37.000Z</UtcCreate>
      <GroupId>6a3FF2XE2BKayWsM</GroupId>
    </GroupInfo>
  </GroupInfos>
  <Success>true</Success>
</QueryDeviceGroupByDevice>
```

</QueryDeviceGroupByDevice>

2.9.11 QuerySuperDeviceGroup

调用该接口根据子分组ID查询父分组信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QuerySuperDeviceGroup。
GroupId	String	是	子分组ID，分组的全局唯一标识符。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	返回错误码，请参见 错误码 。
Data	Data	调用成功时，返回的父分组信息数据。请参考下表GroupInfo。

表 2-70: GroupInfo

名称	类型	描述
GroupName	String	子分组所属的父分组名称。
GroupId	String	子分组所属的父分组ID。
GroupDesc	String	父分组描述。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QuerySuperDeviceGroup
&GroupId=DMoI2Kby5m62Sirz
&公共请求参数
```

返回示例

· JSON 格式

```
{
  "Data": {
    "GroupName": "IOTTEST",
    "GroupId": "tDQvBJqbUyHskDse",
    "GroupDesc": "A test."
  },
  "RequestId": "7411716B-A488-4EEB-9AA0-6DB05AD2491F",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QuerySuperDeviceGroupResponse>
  <Data>
    <GroupName>IOTTEST</GroupName>
    <GroupId>tDQvBJqbUyHskDse</GroupId>
    <GroupDesc>A test.</GroupDesc>
  </Data>
  <RequestId>7411716B-A488-4EEB-9AA0-6DB05AD2491F</RequestId>
  <Success>true</Success>
</QuerySuperDeviceGroupResponse>
```

2.9.12 QueryDeviceListByDeviceGroup

调用该接口查询分组中的设备列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QueryDeviceListByDeviceGroup。
GroupId	String	是	分组ID，分组的全局唯一标识符。
CurrentPage	Integer	否	指定显示查询结果中的第几页。默认值为1。
PageSize	Integer	否	指定返回结果中，每页显示的设备数量。默认值为10。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。

名称	类型	描述
Code	String	返回错误码，请参见 错误码 。
Page	Integer	当前页码。
PageSize	Integer	每页所显示的设备数量。
PageCount	Integer	总页数。
Total	Integer	设备总数。
Data	List	调用成功时，返回的设备列表信息数据。详情请参见下表 SimpleDeviceInfo。

表 2-71: SimpleDeviceInfo

名称	类型	描述
ProductName	String	设备所属的产品名称。
ProductKey	String	设备所属的产品Key。
DeviceName	String	设备名称。
IotId	String	物联网平台为该设备颁发的ID，作为该设备的唯一标识符。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceListByDeviceGroup
&GroupId=7DIgqIl1IjnhBsrA
&公共请求参数
```

返回示例

· JSON格式

```
{
  "PageCount": 1,
  "Data": {
    "SimpleDeviceInfo": [
      {
        "DeviceName": "ios_1207_08",
        "ProductKey": "a1hWjHDWUbF",
        "ProductName": "WIFIdevice",
        "IotId": "TfmUAeJjQQhCPH84UVNn0010c66100"
      },
      {
        "DeviceName": "ios_1207_07",
        "ProductKey": "a1hWjHDWUbF",
        "ProductName": "WIFIGateway",
        "IotId": "wVPeAksaboXBlRgvZNHQ0010310800"
      }
    ]
  }
}
```

```
    },
    {
      "DeviceName": "E1IPK25iL4CT0wnuI2yt",
      "ProductKey": "a1mV8bKjeP6",
      "ProductName": "yanlv",
      "IotId": "E1IPK25iL4CT0wnuI2yt0010598d00"
    }
  ]
},
"PageSize": 10,
"Page": 1,
"RequestId": "B1A921D9-1061-4D45-9F12-EA6B0FDEDE30",
"Success": true,
"Total": 3
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<QueryDeviceListByDeviceGroup>
  <PageCount>1</PageCount>
  <Data>
    <SimpleDeviceInfo>
      <DeviceName>ios_1207_08</DeviceName>
      <ProductKey>a1hWjHDWUbF</ProductKey>
      <ProductName>WIFIdevice</ProductName>
      <IotId>TfmUAeJjQQhCPH84UVNn0010c66100</IotId>
    </SimpleDeviceInfo>
    <SimpleDeviceInfo>
      <DeviceName>ios_1207_07</DeviceName>
      <ProductKey>a1hWjHDWUbF</ProductKey>
      <ProductName>WIFIGateway</ProductName>
      <IotId>wVPeAksaboXB1RgvZNHQ0010310800</IotId>
    </SimpleDeviceInfo>
    <SimpleDeviceInfo>
      <DeviceName>E1IPK25iL4CT0wnuI2yt</DeviceName>
      <ProductKey>a1mV8bKjeP6</ProductKey>
      <ProductName>yanlv</ProductName>
      <IotId>E1IPK25iL4CT0wnuI2yt0010598d00</IotId>
    </SimpleDeviceInfo>
  </Data>
  <PageSize>10</PageSize>
  <Page>1</Page>
  <RequestId>B1A921D9-1061-4D45-9F12-EA6B0FDEDE30</RequestId>
  <Success>true</Success>
  <Total>3</Total>
</QueryDeviceListByDeviceGroup>
```

2.9.13 QueryDeviceGroupByTags

调用该接口根据标签查询设备分组。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作。取值：QueryDeviceGroupByTags。

名称	类型	是否必需	描述
Tags	List	是	标签列表。Tag由TagKey和TagValue组成。具体请参见下表Tag。 · 支持根据TagKey和TagValue查询。 · 也支持只输入TagKey进行查询。
CurrentPage	Integer	否	指定显示查询结果的第几页。默认值是1。
PageSize	Integer	否	指定每页显示的记录数。默认值是10。
公共请求参数	-	是	请参见 公共参数 。

表 2-72: Tag

名称	类型	是否必需	描述
TagKey	String	是	分组标签键(key)。
TagValue	String	否	分组标签值(value)。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
PageCount	Integer	总页数。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页码。
Total	Integer	总记录数。
Data	Data	调用成功时，返回分组信息。详情见下表DeviceGroup。

表 2-73: DeviceGroup

名称	类型	描述
GroupName	String	分组名称。

名称	类型	描述
GroupID	String	分组ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryDeviceGroupByTags
&Tag.1.TagKey=group
&Tag.1.TagValue=tag
&CurrentPage=1
&PageSize=10
&公共请求参数
```

返回示例

· JSON格式

```
{
  "PageCount": 1,
  "Data": {
    "DeviceGroup": [
      {
        "GroupName": "test11",
        "GroupId": "Z0ElGF5aqc0thBtW"
      }
    ]
  },
  "Page": 1,
  "PageSize": 10,
  "RequestId": "9599EE98-1642-4FCD-BFC4-039E458A4693",
  "Success": true,
  "Total": 1
}
```

· XML格式

```
<?xml version="1.0" encoding="utf-8"?>
<QueryDeviceGroupByTags>
  <PageCount>1</PageCount>
  <Data>
    <DeviceGroup>
      <GroupName>test11</GroupName>
      <GroupId>Z0ElGF5aqc0thBtW</GroupId>
    </DeviceGroup>
  </Data>
  <Page>1</Page>
  <PageSize>10</PageSize>
  <RequestId>9599EE98-1642-4FCD-BFC4-039E458A4693</RequestId>
  <Success>true</Success>
  <Total>1</Total>
</QueryDeviceGroupByTags>
```

</QueryDeviceGroupByTags>

2.10 Topic管理

2.10.1 CreateProductTopic

调用该接口为指定产品创建产品Topic类。

限制说明

一个产品下最多允许创建50个Topic类。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：CreateProductTopic。
ProductKey	String	是	要为其创建Topic类的产品Key。
TopicShortName	String	是	设置Topic类的自定义类目名称。Topic类默认包含productKey和deviceName两级类目，类目间以正斜线 (/) 分隔，其格式为：productKey/deviceName/topicShortName。  说明： 每级类目的名称只能包含字母、数字和下划线 (_)，且不能为空。
Operation	String	是	设备对该Topic类的操作权限，取值： SUB：订阅。 PUB：发布。 ALL：发布和订阅。
Desc	String	否	Topic类的描述信息。长度限制为100字符（一个汉字占一个字符）。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
TopicId	Long	调用成功时，IoT Hub为新建的Topic类生成的Topic ID。  说明： 请妥善保管该信息。在调用与该Topic类相关的接口时，您可能需要提供对应的Topic ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=CreateProductTopic
&ProductKey=al*****
&TopicShortName=submit
&Operation=PUB
&Desc=submit a test topic
&公共请求参数
```

返回示例

```
{
  "RequestId": "FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success": true,
  "TopicId": 10000
}
```

2.10.2 UpdateProductTopic

调用该接口修改指定的产品Topic类。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：UpdateProductTopic。

名称	类型	是否必需	描述
TopicId	Long	是	要修改的Topic类的 ID。
TopicShort Name	String	否	设置Topic类的自定义类目名称。Topic类默认包含productKey和deviceName两级类目，类目间以正斜线 (/) 分隔，其格式为：productKey/deviceName/topicShortName。  说明： 每级类目的名称只能包含字母、数字和下划线 (_)，且不能为空。
Operation	String	否	设备对该Topic类的操作权限，取值： SUB：订阅。 PUB：发布。 ALL：发布和订阅。
Desc	String	否	Topic类的描述信息。长度限制为100字符（一个汉字占一个字符）。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=UpdateProductTopic
&ProductKey=a1*****
&TopicId=10000
```

```
&TopicShortName=resubmit
&Operation=PUB
&Desc=resubmit a test topic
&公共请求参数
```

返回示例

```
{
  "RequestId": "FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success": true
}
```

2.10.3 QueryProductTopic

调用该接口查询指定产品的Topic类。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryProductTopic。
ProductKey	String	是	要查询Topic类的产品Key。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
Data	ProductTopicInfo	调用成功时，返回的Topic类信息列表。详情参见 ProductTopicInfo 。

表 2-74: ProductTopicInfo

名称	类型	描述
ProductKey	String	产品Key。
Id	String	Topic类的 ID。

名称	类型	描述
TopicShort Name	String	Topic类中除 <code>productKey</code> 和 <code>deviceName</code> 以外的类目。
Operation	String	设备对该Topic类的操作权限，取值： 0：发布 1：订阅 2：发布和订阅
Desc	String	Topic类的描述信息。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryProductTopic
&ProductKey=HMyB*****
&公共请求参数
```

返回示例

```
{
  "Data": [{
    "Id": "10000",
    "Operation": "0",
    "ProductKey": "HMyB***",
    "TopicShortName": "/HMyB***/${deviceName}/update"
  },
  {
    "Id": "10001",
    "Operation": "2",
    "ProductKey": "HMyB***",
    "TopicShortName": "/HMyB***/${deviceName}/update/error"
  },
  {
    "Id": "10002",
    "Operation": "1",
    "ProductKey": "HMyB***",
    "TopicShortName": "/HMyB***/${deviceName}/get"
  }
  ],
  "RequestId": "B953EAFF-CFF6-4FF8-BC94-8B89F018E4DD",
  "Success": true
}
```

```
}
```

2.10.4 DeleteProductTopic

调用该接口删除指定的Topic类。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteProductTopic。
TopicId	Long	是	要删除的Topic类的 ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteProductTopic
&TopicId=10000
&公共请求参数
```

返回示例

```
{
  "RequestId": "FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success": true
}
```

2.10.5 CreateTopicRouteTable

调用该接口新建Topic间的消息路由关系。

限制说明

- 一个源Topic最多可对应100个目标Topic。

- 源Topic所属的设备必须为已激活设备。
- 源Topic和目标Topic均不能是以 sys 开头的系统 Topic。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：CreateTopicRouteTable。
SrcTopic	String	是	源Topic，即被订阅的Topic。如，SrcTopic=/x7aWKW9****/testDataToDataHub/update。
DstTopic	List	是	目标Topic列表，即从SrcTopic订阅消息的Topic列表。即使只有一个Topic，也使用数组格式。如，DstTopic.1=/x7aWKW9****/deviceNameTest1/add, DstTopic.2=/x7aWKW9****/deviceNameTest2/delete。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
FailureTopics	List	未能成功创建路由关系的Topic列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=CreateTopicRouteTable
&SrcTopic=%2Fx7aWKW9****%2FtestDataToDataHub%2Fupdate
&DstTopic.1=%2Fx7aWKW9****%2FdeviceNameTest1%2Fadd
&DstTopic.2=%2Fx7aWKW9****%2FdeviceNameTest2%2Fdelete
```

&公共请求参数

返回示例

```
{
  "RequestId":"FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success":true,
  "FailureTopics":[]
}
```

2.10.6 DeleteTopicRouteTable

调用该接口删除指定的Topic路由关系。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteTopicRouteTable。
SrcTopic	String	是	源Topic，即被订阅的Topic。如，SrcTopic.1=/x7aWKW9****/testDataToDataHub/update。
DstTopic	List	是	目标Topic列表，即从SrcTopic订阅消息的Topic列表。如，DstTopic.1=/x7aWKW9****/deviceNameTest1/add，DstTopic.2=/x7aWKW9****/deviceNameTest2/delete。
公共请求参数	-	是	请参见 签名机制 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
FailureTopics	List	未能成功删除路由关系的Topic列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteTopicRouteTable
&SrcTopic=%2F%2F7aWKW94bb8%2FtestDataToDataHub%2Fupdate
&DstTopic.1=%2F%2F7aWKW94bb8%2FdeviceNameTest1%2Fadd
&DstTopic.2=%2F%2F7aWKW94bb8%2FdeviceNameTest2%2Fdelete
&公共请求参数
```

返回示例

```
{
  "RequestId": "FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success": true,
  "FailureTopics": []
}
```

2.10.7 QueryTopicReverseRouteTable

调用该接口查询指定Topic订阅的源Topic，即反向路由表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryTopicReverseRouteTable。
Topic	String	是	要查询的目标Topic，即接收消息的Topic。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
SrcTopics	List	调用成功时，返回的源Topic列表，即被订阅的Topic列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryTopicReverseRouteTable
&Topic=%2F%7aWKW94bb8%2FtestDataToDataHub%2Fupdate
&公共请求参数
```

返回示例

```
{
  "RequestId": "FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success": true,
  "SrcTopics": ["/CXi4***/device1/get", "/CXi4***/device4/get"]
}
```

2.10.8 QueryTopicRouteTable

调用该接口查询向指定Topic订阅消息的目标Topic，即指定Topic的路由表。该接口只支持查询用户的Topic。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：QueryTopicRouteTable。
Topic	String	是	要查询的源Topic，即发布消息的Topic。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
DstTopics	List	调用成功时，返回的目标Topic列表，即向源Topic订阅消息的Topic列表。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryTopicRouteTable
&Topic=%2F%7aWKW94bb8%2FtestDataToDataHub%2Fupdate
&公共请求参数
```

返回示例

```
{
  "RequestId": "FCC27691-9151-4B93-9622-9C90F30542EC",
  "Success": true,
  "DstTopics": ["/CXi4***/device2/get", "/CXi4***/device3/get"]
}
```

2.11 规则引擎

2.11.1 CreateRule

调用该接口对指定Topic新建一个规则。

请求参数



说明:

如需启动规则，需包含ProductKey、ShortTopic、Select三个参数的信息。

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：CreateRule。
Name	String	是	规则名称。支持使用中英文字符、数字和下划线（_），长度应为4~30位（一个中文字符算2位）。
ProductKey	String	否	应用该规则的产品Key。

名称	类型	是否必需	描述
ShortTopic	String	否	应用该规则的具体Topic，格式为：\${deviceName}/topicShortName。其中，\${deviceName}指具体设备的名称，topicShortName是自定义类目。 调用QueryDevice接口，查看产品下的所有设备。返回结果中包含所有的DeviceName。 调用QueryProductTopic接口，查看产品下的所有Topic类。返回结果中包含所有的TopicShortName。 示例： · 系统Topic的ShortTopic，如：\${deviceName}/thing/event/property/post，其中\${deviceName}可以使用通配符+代替，表示产品下所有设备名称。 · 自定义Topic的ShortTopic，如：\${deviceName}/user/get。 指定自定义Topic时，可以使用通配符+和#。 - \${deviceName}可以使用通配符+代替，表示产品下所有设备； - 之后字段可以用/user/#，#表示/user层级之后的所有层级名称。 使用通配符，请参见Topic类中的通配符。 · 设备状态Topic的ShortTopic：\${deviceName}。

名称	类型	是否必需	描述
Select	String	否	要执行的SQL Select语句。具体内容参照SQL表达式。  说明: 此处传入的是Select下的内容。例如, 如果Select语句为Select a,b,c, 则此处传入a,b,c。
RuleDesc	String	否	规则的描述信息。长度限制为100字符（一个汉字占一个字符）。
DataType	String	否	规则处理的数据格式, 需与待处理的设备数据格式一致。取值: · JSON: JSON数据。 · BINARY: 二进制数据。  说明: 若选择为BINARY, TopicType不能选择为0 (系统Topic), 且不能将数据转发至表格存储、时序时空数据库和云数据库RDS版。 如果不传入该参数, 则默认使用JSON格式。
Where	String	否	规则的触发条件。具体内容参照SQL表达式。  说明: 此处传入的是Where中的内容。例如, 如果Where语句为Where a>10, 则此处传入a>10。

名称	类型	是否必需	描述
TopicType	Integer	是	0: 系统Topic, 包含: /thing/event/property/post 设备上报的属性消息。 /thing/event/\${tsl.event.identifier}/post, \${}中的是产品物模型中事件identifier。设备上报的事件消息 - thing/lifecycle 设备生命周期变更消息。 /thing/downlink/reply/message 设备响应云端指令的结果消息。 /thing/list/found网关上报发现子设备消息。 - thing/topo/lifecycle设备拓扑关系变更消息。 1: 自定义Topic。 2: 设备状态消息Topic: /as/mqtt/status/\${productKey}/\${deviceName}。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功, false表示调用失败。
ErrorMessage	String	调用失败时, 返回的出错信息。
Code	String	调用失败时, 返回的错误码。错误码详情, 请参见 错误码 。

名称	类型	描述
RuleId	Long	调用成功时，规则引擎为该规则生成的规则ID，作为该规则的标识符。  说明： 请妥善保管该信息。在调用和规则相关的接口时，您可能需要提供对应的规则ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=CreateRule
&Name=iot_test1
&ProductKey=a1*****
&ShortTopic=open_api_dev/get
&Select=a,b,c
&RuleDesc=rule test
&DataType=JSON
&Where=a>10
&TopicType=1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "E4C0FF92-2A86-41DB-92D3-73B60310D25E",
  "RuleId": 1000,
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<CreateRuleResponse>
  <RequestId>E4C0FF92-2A86-41DB-92D3-73B60310D25E</RequestId>
  <RuleId>1000</RuleId>
  <Success>true</Success>
</CreateRuleResponse>
```

2.11.2 CreateRuleAction

调用该接口在指定的规则下创建一个规则动作，定义将处理后的Topic数据转发至物联网平台的其他Topic或所支持的其他阿里云服务。

限制说明

- 服务地域不同，所支持的目标云产品有所不同。规则引擎支持的地域及目标云产品，请参见[地域与可用区](#)。

- 一个规则下面最多可创建10个规则动作。
- 您可以通过调用该API创建规则动作，定义将数据转发至物联网平台其他Topic和其他阿里云产品，包括DataHub、消息服务、函数计算、表格存储和消息队列RocketMQ。但是，如果您想将数据转发至时序时空数据库（TSDB）和云数据库RDS版，请在[物联网平台控制台](#)进行操作。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：CreateRuleAction。
RuleId	Long	是	要为其创建动作的规则ID。
Type	String	是	规则动作类型，取值： DATAHUB：将根据规则处理后的Topic数据转发至阿里云DataHub，进行流式数据处理。 ONS：将根据规则处理后的Topic数据转发至阿里云消息队列RocketMQ，进行消息分发。 MNS：将根据规则处理后的Topic数据发送至阿里云消息服务（Message Service）中，进行消息传输。 FC：将根据规则处理后的Topic数据发送至阿里云函数计算服务，进行事件计算。 OTS：将根据规则处理后的Topic数据发送至阿里云表格存储，进行NoSQL数据存储。  说明： 数据格式为二进制的规则（即规则的DataType参数是BINARY）不支持转发数据至OTS（表格存储）。 REUBLISH：将根据规则处理后的Topic数据转发至另一个物联网平台 Topic。
Configuration	String	是	该规则动作的配置信息。不同规则动作类型所需内容不同，具体要求见下文描述。
公共请求参数	-	是	请参见 公共参数 。

REUBLISH类型Configuration定义

表 2-75: REUBLISH Configuration

名称	描述
topic	转发至的目标Topic（系统Topic或自定义Topic）。 支持将数据转发至如下高级版设备的下行系统Topic： • /sys/\${YourProductKey}/\${YourDeviceName}/thing/service/property/set • /sys/\${YourProductKey}/\${YourDeviceName}/thing/service/\${tsl.service.identifier}, 变量\${tsl.service.identifier}的内容由该产品物模型中的服务决定。
topicType	Topic的类型。 • 0表示高级版产品下行系统Topic。 • 1表示用户自定义Topic。

REUBLISH 类型Configuration示例：

```
sys类型: {"topic":"/sys/a1TXXXXXWSN/xxx_cache001/thing/service/property/set","topicType":0}  
自定义类型: {"topic":"/a1TXXXXXWSN/xxx_cache001/user/update","topicType":1}
```

DATAHUB类型Configuration定义

表 2-76: DATAHUB Configuration

名称	描述
projectName	目标DataHub中用来接收信息的具体Project。
topicName	目标DataHub中用来接收信息的具体Topic。

名称	描述
regionName	目标DataHub所在地域。 具体RegionName写法请参考 地域和可用区 文档。
role	授权角色信息。通过授予IoT指定的系统服务角色，您可以授权物联网平台访问您的DataHub。授权角色信息格式如下： <pre>{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingdatahubrole", "roleName": "AliyunIOTAccessingDataHubRole" }</pre> 请将6541***替换成您的阿里云账号ID。您可以登录控制台，在 账号安全设置 页面查看您的账号ID。 AliyunIOTAccessingDataHubRole 是访问控制中定义的服务角色。用于授予物联网平台访问DataHub。关于角色的更多信息，请在 访问控制控制台 的角色管理页面进行角色管理。
schemaVals	目标DataHub中的Schema列表，详情参见 schemaVals 。

表 2-77: schemaVals

名称	描述
name	列名。
value	列值。
type	列类型，取值： BIGINT：大整数型 DOUBLE：双精度浮点型 BOOLEAN：布尔型 TIMESTAMP：时间戳型 STRING：字符串型 DECIMAL：小数型

DATAHUB类型Configuration示例：

```
{
  "schemaVals": [
    {
      "name": "devicename",
      "value": "${deviceName}",
      "type": "STRING"
    },
    {
      "name": "msgtime",
      "value": "${msgTime}",
      "type": "TIMESTAMP"
    }
  ],
  "role": {
    "roleArn": "acs:ram::6541***:role/aliyuniotaccessingdatahubrole",
    "roleName": "AliyunIOTAccessingDataHubRole"
  },
  "projectName": "iot_datahub_stream",
  "topicName": "device_message",
  "regionName": "cn-shanghai"
}
```

OTS类型Configuration定义**表 2-78: OTS Configuration**

名称	描述
instanceName	表格存储中用来接收信息的实例名称。
tableName	表格存储中用来接收信息的数据表名称。
regionName	目标实例所在地域。 具体RegionName写法请参考 地域和可用区 文档。

名称	描述
role	授权角色信息。通过授予物联网平台指定的系统服务角色，您可以授权物联网平台访问您的表格存储。授权角色信息如下： <pre>{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingotsrole", "roleName": "AliyunIOTAccessingOTSRole" }</pre> 请将6541***替换成您的阿里云账号ID。您可以登录控制台，在 账号安全设置 页面查看您的账号ID。 AliyunIOTAccessingOTSRole是访问控制中定义的服务角色。用于授予物联网平台访问表格存储。关于角色的更多信息，您可以登录访问控制控制台，在 角色管理 页面进行角色管理。
primaryKeys	目标表中的主键列表。详情参见 PrimaryKeys 。

表 2-79: PrimaryKeys

名称	描述
columnType	主键类型，取值： INTEGER：整型 STRING：字符串 BINARY：二进制
columnName	主键名称。
columnValue	主键值。
option	主键是否为自增列，取值：AUTO_INCREMENT或为空。当主键类型为INTEGER，且该字段为AUTO_INCREMENT时，主键为自增列。

OTS类型Configuration示例：

```
{
  "instanceName": "testaaa",
  "tableName": "tt",
  "primaryKeys": [
    {
      "columnType": "STRING",
      "columnName": "ttt",
      "columnValue": "${tt}",
      "option": ""
    }
  ]
}
```

```
    },
    {
      "columnType": "INTEGER",
      "columnName": "id",
      "columnValue": "",
      "option": "AUTO_INCREMENT"
    }
  ],
  "regionName": "cn-shanghai",
  "role": {
    "roleArn": "acs:ram::5645***:role/aliyuniotaccessingotsrole",
    "roleName": "AliyunIOTAccessingOTSRole"
  }
}
```

MNS类型Configuration定义

表 2-80: MNS Configuration

名称	描述
themeName	消息服务中用来接收信息的目标主题名称。
regionName	目标消息服务所在区域。 具体RegionName写法请参考 地域和可用区 文档。
role	授权角色信息。通过授予IoT指定的系统服务角色，您可以授权物联网平台访问您的消息服务。授权角色信息如下： <pre>{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingmnsrole", "roleName": "AliyunIOTAccessingMNSRole" }</pre> 请将6541***替换成您的阿里云账号ID。您可以登录控制台，在 账号安全设置 页面查看您的账号ID。 AliyunIOTAccessingMNSRole是访问控制中定义的服务角色。用于授予物联网平台访问消息服务。关于角色的更多信息，您可以登录访问控制控制台，在角色管理页面进行角色管理。

MNS类型Configuration示例：

```
{
  "themeName": "mns-test-topic1",
  "regionName": "cn-shanghai",
  "role": {
    "roleArn": "acs:ram::5645***:role/aliyuniotaccessingmnsrole",
    "roleName": "AliyunIOTAccessingMNSRole"
  }
}
```

```
    }  
  }  
}
```

FC类型Configuration定义

表 2-81: FC Configuration

名称	描述
functionName	函数服务中用来接收信息的目标函数名称。
serviceName	函数服务中用来接收信息的目标服务名称。
regionName	目标函数服务实例所在区域。 具体RegionName写法请参考 地域和可用区 文档。
role	授权角色信息。通过授予IoT指定的系统服务角色，您可以授权物联网平台访问您的函数计算服务。授权角色信息如下： <pre>{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingfc role", "roleName": "AliyunIOTAccessingFCRole" }</pre> 请将6541***替换成您的阿里云账号ID。您可以登录控制台，在 账号安全设置 页面查看您的账号ID。 AliyunIOTAccessingFCRole是访问控制中定义的服务角色。用于授予物联网平台访问函数计算。关于角色的更多信息，您可以登录访问控制控制台，在角色管理页面进行角色管理。

FC类型Configuration示例

```
{  
  "regionName": "cn-shanghai",  
  "role": {  
    "roleArn": "acs:ram::5645***:role/aliyuniotaccessingfcrole",  
    "roleName": "AliyunIOTAccessingFCRole"  
  },  
  "functionName": "weatherForecast",  
  "serviceName": "weather"  
}
```

ONS类型Configuration定义

 说明:

您需通过调用消息队列RocketMQ的SDK，或在消息队列RocketMQ控制台，授权物联网平台访问消息队列RocketMQ（至少要授予物联网平台发布权限），然后才能够成功创建将Topic数据转发至消息队列RocketMQ的规则动作。固定授权账号：1135850448829929。

表 2-82: ONS Configuration

名称	描述
instanceId	RocketMQ中用来接收消息的目标Topic所属的实例ID。
topic	RocketMQ中用来接收信息的目标Topic。
regionName	目标RocketMQ实例所在区域。 具体RegionName写法请参考地域和可用区文档。  说明： 公网和同区流转，使用普通版RocketMQ实例即可；如果您需要跨区流转，则RocketMQ实例必需是铂金版实例。
role	授权角色信息。通过授予IoT指定的系统服务角色，您可以授权物联网平台访问您的消息队列RocketMQ服务。授权角色信息如下： <pre>{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingonsrole", "roleName": "AliyunIOTAccessingONSRole" }</pre> 请将6541***替换成您的阿里云账号ID。您可以登录控制台，在 账号安全设置 页面查看您的账号ID。 AliyunIOTAccessingONSRole是访问控制中定义的服务角色。用于授予物联网平台访问消息队列RocketMQ。关于角色的更多信息，您可以登录访问控制控制台，在角色管理页面进行角色管理。

ONS类型Configuration示例

```
{
  "instanceId": "MQ_INST_1231579085529123_XXXXXX",
  "topic": "aliyun-iot-XXXXX",
  "regionName": "cn-hangzhou",
  "role": {
    "roleArn": "acs:ram::6541***:role/aliyuniotaccessingonsrole",
    "roleName": "AliyunIOTAccessingONSRole",
  }
}
```

```
}
```

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
ActionId	Long	调用成功时，规则引擎为该规则动作生成的规则动作ID，作为其标识符。  说明： 请妥善保管该信息。在调用与规则动作相关的接口时，您可能需要提供对应的规则动作ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=CreateRuleAction
&RuleId=1000
&Type=REUBLISH
&Configuration={"topic":"/a1*****/device1/get","topicType":1}
&公共请求参数
```

返回示例

```
{
  "RequestId": "21D327AF-A7DE-4E59-B5D1-ACAC8C024555",
  "ActionId": 10003,
  "Success": true
}
```

2.11.3 DeleteRule

调用该接口删除指定的规则。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteRule。
RuleId	Long	是	要删除的规则ID。

名称	类型	是否必需	描述
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteRule
&RuleId=1000
&公共请求参数
```

返回示例

```
{
  "RequestId": "A8F48485-44B9-40D8-A56D-F716F384F387",
  "Success": true
}
```

2.11.4 DeleteRuleAction

调用该接口删除指定的规则动作。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：DeleteRuleAction。
ActionId	Long	是	要删除的规则动作ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=DeleteRuleAction
&ActionId=10001
&公共请求参数
```

返回示例

```
{
  "RequestId": "8FC9E36B-E0DC-4802-84EE-184E255B4E95",
  "Success": true
}
```

2.11.5 GetRule

调用该接口查询指定规则的详细信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetRule。
RuleId	Long	是	要查询的规则ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。

名称	类型	描述
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
RuleInfo	RuleInfo	调用成功时，返回的规则详细信息。详情参见 RuleInfo 。

表 2-83: RuleInfo

名称	类型	描述
CreateUserId	Long	创建该规则的用户ID。
Created	String	该规则创建时的CST时间。
DataType	String	该规则的数据类型，取值：JSON或BINARY。
Id	Long	规则ID。
Modified	String	该规则最近一次被修改时的CST时间。
Name	String	规则名称。
ProductKey	String	应用该规则的产品Key。
RuleDesc	String	规则的描述信息。
Select	String	该规则SQL语句中的Select内容。
ShortTopic	String	该规则所处理消息来源的具体Topic（不包含ProductKey类目），格式为： <code>\${deviceName}/topicShortName</code> 。其中， <code>\${deviceName}</code> 指具体设备的名称， <code>topicShortName</code> 是该设备的自定义类目。
Status	String	该规则的运行状态。取值： RUNNING：运行中 STOP：停止
Topic	String	该规则所处理消息来源的完整Topic，格式为： <code>\${productKey}/\${deviceName}/topicShortName</code> 。
Where	String	该规则SQL语句中的Where查询条件。

名称	类型	描述
TopicType	Integer	若您设置了规则SQL语句，则返回： · 0：表示系统Topic。 · 1：表示自定义Topic。 · 2：表示设备状态Topic。 若未设置过规则SQL语句，则返回-1。
UtcCreated	String	规则创建时的UTC时间。
UtcModified	String	规则最近一次更新时的UTC时间。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetRule
&RuleId=100000
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "58D4CEC0-3E95-4DBE-AFC1-809D1400E52F",
  "Success": true,
  "RuleInfo": {
    "DataType": "JSON",
    "ShortTopic": "+/#",
    "ProductKey": "a1KiV*****",
    "UtcModified": "2019-02-28T06:20:58.000Z",
    "CreateUserId": "12*****85529123",
    "UtcCreated": "2019-02-28T06:14:33.000Z",
    "Name": "iotrules",
    "Status": "STOP",
    "Select": "deviceName() as deviceName",
    "Created": "Thu Feb 28 14:14:33 CST 2019",
    "Modified": "Thu Feb 28 14:20:58 CST 2019",
    "TopicType": 1,
    "Topic": "/a1KiV*****/+/#",
    "Id": 100000
  }
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<GetRuleResponse>
  <RequestId>58D4CEC0-3E95-4DBE-AFC1-809D1400E52F</RequestId>
  <Success>true</Success>
  <RuleInfo>
    <DataType>JSON</DataType>
```

```

<ShortTopic>+/#</ShortTopic>
<ProductKey>a1KiV*****</ProductKey>
<UtcModified>2019-02-28T06:20:58.000Z</UtcModified>
<CreateUserId>1231579085*****</CreateUserId>
<UtcCreated>2019-02-28T06:14:33.000Z</UtcCreated>
<Name>iotrules</Name>
<Status>STOP</Status>
<Select>deviceName() as deviceName</Select>
<Created>Thu Feb 28 14:14:33 CST 2019</Created>
<Modified>Thu Feb 28 14:20:58 CST 2019</Modified>
<TopicType>1</TopicType>
<Topic>/a1KiV*****/+/#</Topic>
<Id>100000</Id>
</RuleInfo>
</GetRuleResponse>

```

2.11.6 GetRuleAction

调用该接口查询指定规则动作的详细信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetRuleAction。
ActionId	Long	是	要查询的规则动作ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
RuleActionInfo	RuleActionInfo	调用成功时，返回的规则动作详细信息。详情参见 RuleActionInfo 。

表 2-84: RuleActionInfo

名称	类型	描述
Id	Long	规则动作ID。
RuleId	Long	该规则动作对应的规则ID。

名称	类型	描述
Type	String	规则动作类，取值： REUBLISH：转发到另一个topic。 OTS：存储到表格存储。 MNS：发送消息到消息服务。 ONS：发送数据到消息队列。 TSDB：存储到高性能时间序列数据库。 FC：发送数据到函数计算。 DATAHUB：发送数据到DataHub中。 RDS：存储数据到云数据库中。
Configuration	String	该规则动作的配置信息。
ErrorActionFlag	String	该规则动作是否为转发错误操作数据的转发动作，即转发流转到其他云产品失败且重试失败的数据。 · true：该规则动作转发错误操作数据。 · false：该规则动作不转发错误操作数据，而是正常转发操作。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetRuleAction
&ActionId=10001
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RuleActionInfo": {
    "Type": "REUBLISH",
    "RuleId": 152323,
    "Id": 100001,
    "Configuration": "{\"topic\":\"/sys/a1zSA28HUyy/device/thing/service/property/set\",\"topicType\":0,\"uid\":\"1231579*****\"}",
    "ErrorActionFlag": false
  },
  "RequestId": "F2D0755D-F350-40FE-9A6D-491859DB5E5F",
  "Success": true
}
```

}

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<GetRuleActionResponse>
  <RuleActionInfo>
    <Type>REUBLISH</Type>
    <RuleId>152323</RuleId>
    <Id>100001</Id>
    <Configuration>
      <topic>/sys/alzSA28HUyy/device/thing/service/property/
set</topic>
      <topicType>0</topicType>
      <uid>1231579*****</uid>
    </Configuration>
    <ErrorActionFlag>false</ErrorActionFlag>
  </RuleActionInfo>
  <RequestId>F2D0755D-F350-40FE-9A6D-491859DB5E5F</RequestId>
  <Success>true</Success>
</GetRuleActionResponse>
```

2.11.7 ListRule

调用该接口分页查询所有规则列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：ListRule。
PageSize	Integer	是	返回结果中每页显示的记录数量。最大取值100，默认值是10。
CurrentPage	Integer	是	显示返回结果中的第几页。最大取值1000，默认值1。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

名称	类型	描述
Data	RuleInfo	调用成功时，返回的规则信息列表。详情参见 RuleInfo 。  说明： 返回规则信息按照规则创建时间倒序排列。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页码。
Total	Integer	总页数。

表 2-85: RuleInfo

名称	类型	描述
CreateUserId	Long	创建该规则的用户ID。
Created	String	该规则创建时的CST(Central Standard Time)时间。
DataType	String	该规则的数据类型，取值：JSON和BINARY。
Id	Long	规则ID。
Modified	String	该规则最近一次被修改时的CST (Central Standard Time)时间。
Name	String	规则名称。
ProductKey	String	应用该规则的产品Key。
RuleDesc	String	规则的描述信息。
Select	String	该规则SQL语句中的Select内容。
ShortTopic	String	应用该规则的具体Topic（不包含ProductKey类目），格式为： <code>\${deviceName}/topicShortName</code> 。其中， <code>\${deviceName}</code> 指具体设备的名称， <code>topicShortName</code> 是该设备的自定义类目。
Status	String	该规则的运行状态。取值： RUNNING：运行中 STOP：停止
Topic	String	应用该规则的具体Topic，格式为： <code>\${productKey}/\${deviceName}/topicShortName</code> 。
UtcModified	String	该规则最近一次被修改时的UTC时间。

名称	类型	描述
UtcCreated	String	该规则创建时的UTC时间。
Where	String	该规则SQL语句中的Where查询条件。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=ListRule
&PageSize=2
&CurrentPage=1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "Data": {
    "RuleInfo": [
      {
        "DataType": "JSON",
        "ShortTopic": "+/#",
        "ProductKey": "a1KiV*****",
        "CreateUserId": "1231579085*****",
        "UtcModified": "2019-02-28T06:20:58.000Z",
        "UtcCreated": "2019-02-28T06:14:33.000Z",
        "Name": "iotrules",
        "Status": "STOP",
        "Select": "deviceName() as deviceName",
        "Created": "Thu Feb 28 14:14:33 CST 2019",
        "Modified": "Thu Feb 28 14:20:58 CST 2019",
        "Topic": "/a1KiV*****/+/#",
        "Id": 152323
      },
      {
        "Name": "test123",
        "Status": "STOP",
        "Created": "Wed Feb 27 20:45:43 CST 2019",
        "DataType": "JSON",
        "Modified": "Wed Feb 27 20:45:43 CST 2019",
        "UtcModified": "2019-02-27T12:45:43.000Z",
        "CreateUserId": "1231579085*****",
        "Id": 151454,
        "UtcCreated": "2019-02-27T12:45:43.000Z"
      }
    ]
  },
  "PageSize": 2,
  "Page": 1,
  "RequestId": "1564B626-DE97-452D-9E9B-305888AC6105",
  "Success": true,
  "Total": 25
}
```



```
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<ListRuleResponse>
  <Data>
    <RuleInfo>
      <DataType>JSON</DataType>
      <ShortTopic>+/#</ShortTopic>
      <ProductKey>a1KiV*****</ProductKey>
      <CreateUserId>1231579085*****</CreateUserId>
      <UtcModified>2019-02-28T06:20:58.000Z</UtcModified>
      <UtcCreated>2019-02-28T06:14:33.000Z</UtcCreated>
      <Name>iotrules</Name>
      <Status>STOP</Status>
      <Select>deviceName() as deviceName</Select>
      <Created>Thu Feb 28 14:14:33 CST 2019</Created>
      <Modified>Thu Feb 28 14:20:58 CST 2019</Modified>
      <Topic>/a1KiV*****/+/#</Topic>
      <Id>152323</Id>
    </RuleInfo>
    <RuleInfo>
      <Name>test123</Name>
      <Status>STOP</Status>
      <Created>Wed Feb 27 20:45:43 CST 2019</Created>
      <DataType>JSON</DataType>
      <Modified>Wed Feb 27 20:45:43 CST 2019</Modified>
      <UtcModified>2019-02-27T12:45:43.000Z</UtcModified>
      <CreateUserId>1231579085*****</CreateUserId>
      <Id>151454</Id>
      <UtcCreated>2019-02-27T12:45:43.000Z</UtcCreated>
    </RuleInfo>
  </Data>
  <PageSize>2</PageSize>
  <Page>1</Page>
  <RequestId>87B1B7C3-FD90-4B80-93BC-A52F7996A903</RequestId>
  <Success>true</Success>
  <Total>25</Total>
</ListRuleResponse>
```

2.11.8 ListRuleActions

调用该接口查询指定规则下的所有规则动作列表。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：ListRuleActions。
RuleId	String	是	要查询的规则ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
RuleActionList	RuleActionInfo	调用成功时，返回的规则动作信息列表。详情参见下表RuleActionInfo。

表 2-86: RuleActionInfo

名称	类型	描述
Id	Long	规则动作ID。
RuleId	Long	该规则动作对应的规则ID。
Type	String	规则动作类型。取值： REUBLISH：转发到另一个topic。 OTS：存储到表格存储。 MNS：发送消息到消息服务。 ONS：发送数据到消息队列。 TSDB：存储到高性能时间序列数据库 FC：发送数据到函数计算。 DATAHUB：发送数据到DataHub中。 RDS：存储数据到云数据库中。
Configuration	String	该规则动作的配置信息。
ErrorActionFlag	String	该规则动作是否为转发错误操作数据的转发动作，即转发流转到其他云产品失败且重试失败的数据。 · true：该规则动作转发错误操作数据。 · false：该规则动作不转发错误操作数据，而是正常转发操作。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=ListRuleActions
&RuleId=10000
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RuleActionList": {
    "RuleActionInfo": [
      {
        "Type": "OTS",
        "RuleId": 10000,
        "Id": 139099,
        "Configuration": "{\"endPoint\":\"http://ShanghaiRegion.cn-shanghai.ots.aliyuncs.com\", \"instanceName\":\"ShanghaiRegion\", \"primaryKeys\": [{\"columnName\":\"temperature\", \"columnType\":\"INTEGER\", \"columnValue\":\"${deviceName}\"}], \"regionName\":\"cn-shanghai\", \"role\":{\"roleArn\":\"acs:ram::1231579085*:role/aliyuniotaccessingotsrole\", \"roleName\":\"AliyunIOTAccessingOTSRole\"}, \"tableName\":\"iottest\", \"uid\":\"1231579085*\"}",
        "ErrorActionFlag": false
      },
      {
        "Type": "REPUBLISH",
        "RuleId": 152323,
        "Id": 142401,
        "Configuration": "{\"topic\":\"/sys/alzSA28H***/device/thing/service/property/set\", \"topicType\":0, \"uid\":\"1231579085*\"}",
        "ErrorActionFlag": false
      }
    ]
  },
  "RequestId": "22254BDB-3DC1-4643-8D1B-EE0437EF09A9",
  "Success": true
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8" ?>
<ListRuleActionsResponse>
  <RuleActionList>
    <RuleActionInfo>
      <Type>REPUBLISH</Type>
      <RuleId>152323</RuleId>
      <Id>142401</Id>
      <Configuration>
        <topic>/sys/alzSA28HUyy/device/thing/service/property/set</topic>
        <topicType>0</topicType>
        <uid>1231579*****</uid>
      </Configuration>
      <ErrorActionFlag>>false</ErrorActionFlag>
    </RuleActionInfo>
  </RuleActionList>
</ListRuleActionsResponse>
```

```
<RequestId>22254BDB-3DC1-4643-8D1B-EE0437EF09A9</RequestId>
<Success>true</Success>
</ListRuleActionsResponse>
```

2.11.9 StartRule

调用该接口启动指定的规则。

请求参数



说明:

启动规则前需要确认该规则已经配置了SQL。配置SQL，参考[CreateRule](#)接口。

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：StartRule。
RuleId	Long	是	要启动的规则ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=StartRule
&RuleId=1000
&公共请求参数
```

返回示例

```
{
  "RequestId": "9A2F243E-17FE-4846-BAB5-D02A25155AC4",
  "Success": true
}
```

```
}
```

2.11.10 StopRule

调用该接口停止指定的规则。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：StopRule。
RuleId	Long	是	指要停止的规则ID。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=StopRule
&RuleId=100000
&公共请求参数
```

返回示例

```
{
  "RequestId": "9A2F243E-17FE-4846-BAB5-D02A25155AC4",
  "Success": true
}
```

```
}
```

2.11.11 UpdateRule

调用该接口修改指定的规则。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：UpdateRule。
RuleId	Long	是	要修改的规则ID。
Name	String	是	规则名称。支持使用中英文字符、数字和下划线（_），长度应为4~30位（一个中文字符算2位）。
ProductKey	String	否	应用该规则的产品Key。

名称	类型	是否必需	描述
ShortTopic	String	否	应用该规则的具体Topic，格式为：\${deviceName}/topicShortName。其中，\${deviceName}指具体设备的名称，topicShortName是自定义类目。 调用QueryDevice接口，查看产品下的所有设备。返回结果中包含所有的DeviceName。 调用QueryProductTopic接口，查看产品下的所有Topic类。返回结果中包含所有的TopicShortName。 示例： · 系统Topic的ShortTopic，如：\${deviceName}/thing/event/property/post，其中\${deviceName}可以使用通配符+代替，表示产品下所有设备名称。 · 自定义Topic的ShortTopic，如：\${deviceName}/user/get。 指定自定义Topic时，可以使用通配符+和#。 - \${deviceName}可以使用通配符+代替，表示产品下所有设备； - 之后字段可以用/user/#，#表示/user层级之后的所有层级名称。 使用通配符，请参见Topic类中的通配符。 · 设备状态Topic的ShortTopic：\${deviceName}。

名称	类型	是否必需	描述
Select	String	否	要执行的SQL Select语句。具体内容参照SQL表达式。  说明: 此处传入的是Select下的内容。例如，如果Select语句为Select a,b,c, 则此处传入a,b,c。
RuleDesc	String	否	规则的描述信息。长度限制为100字符（一个汉字占一个字符）。
Where	String	否	规则的触发条件。具体内容参照SQL表达式。  说明: 此处传入的是Where中的内容。例如，如果Where语句为Where a>10, 则此处传入a>10。
TopicType	Integer	否	0: 系统Topic, 包含: /thing/event/property/post 设备上报的属性消息。 /thing/event/\${tsl.event.identifier}/post, \${}中的是产品物模型中事件identifier。设备上报的事件消息 - thing/lifecycle 设备生命周期变更消息。 /thing/downlink/reply/message 设备响应云端指令的结果消息。 /thing/list/found网关上报发现子设备消息。 - thing/topo/lifecycle设备拓扑关系变更消息。 1: 自定义Topic。 2: 设备状态消息Topic: /as/mqtt/status/\${productKey}/\${deviceName}。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=UpdateRule
&RuleId=1000
&Name=test_2
&ProductKey=a1*****
&ShortTopic=+/get
&Select=a,b,c
&RuleDesc=test
&Where=a>10
&TopicType=1
&公共请求参数
```

返回示例

```
{
  "RequestId": "9A2F243E-17FE-4846-BAB5-D02A25155AC4",
  "Success": true
}
```

2.11.12 UpdateRuleAction

调用该接口修改指定的规则动作。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：UpdateRuleAction。
ActionId	Long	是	要修改的规则动作ID。

名称	类型	是否必需	描述
Type	String	是	规则动作类型，取值： DATAHUB：将根据规则处理后的Topic数据转发至阿里云DataHub，进行流式数据处理。 ONS：将根据规则处理后的Topic数据转发至阿里云消息队列，进行消息分发。 MNS：将根据规则处理后的Topic数据发送至阿里云消息服务（Message Service）中，进行消息传输。 FC：将根据规则处理后的Topic数据发送至阿里云函数计算服务，进行事件计算。 OTS：将根据规则处理后的Topic数据发送至阿里云表格存储，进行NoSQL数据存储。 REPUBLISH：将根据规则处理后的Topic数据转发至另一个物联网平台 Topic。  说明： · 服务地域不同，规则引擎所支持的数据转发目标云产品不同。具体请参见规则引擎相关地域和可用区。 · 数据格式为二进制的规则（即规则的DataType参数是BINARY）不支持转发数据至OTS（表格存储）。
Configuration	String	是	该规则动作的配置信息。不同规则动作类型所需配置内容不同。具体要求，请参见 CreateActionRule 中的各规则动作类型的Configuration描述。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。

名称	类型	描述
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=UpdateRuleAction
&ActionId=10003
&Type=REPUBLISH
&Configuration={"topic":"/a1*****/device5/get","topicType":1}
&公共请求参数
```

返回示例

```
{
  "RequestId": "21D327AF-A7DE-4E59-B5D1-ACAC8C024555",
  "Success": true
}
```

2.12 消息通信

2.12.1 Pub

调用该接口向指定Topic发布消息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：Pub。
ProductKey	String	是	要发送消息产品Key。

名称	类型	是否必需	描述
TopicFullName	String	是	要接收消息的Topic，如/a1Q5XoY****/device1/user/update。  说明： · 不支持系统Topic。 · 指定Topic的操作权限须为发布或发布和订阅。 您可以调用QueryProductTopic接口查询产品下的Topic类列表，或在设备详情页的Topic列表页签下查看设备的具体Topic。
MessageContent	String	是	要发送的消息主体。您需要将消息原文转换成二进制数据，并进行Base64编码，从而生成消息主体。
Qos	Integer	否	指定消息的发送方式。取值： 0：最多发送一次。 1：最少发送一次。 如果不传入此参数，则使用默认值0。  说明： QoS=1的消息在物联网平台中最多可以保存7天。物联网平台不保存QoS=0的消息。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
MessageId	String	成功发送消息后，云端生成的消息ID，用于标识该消息。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=Pub
&ProductKey=a1Q5XoY****
&TopicFullName=%2a1Q5XoY****%2Fdevice1%2Fuser%2Fget
&MessageContent=aGVsbG8gd29ybGQ%3D
&Qos=0
&公共请求参数
```

返回示例

· JSON 格式

```
{
  "RequestId": "BB71E443-4447-4024-A000-EDE09922891E",
  "Success": true,
  "MessageId": 889455942124347329
}
```

· XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
  <PubResponse>
    <RequestId>BB71E443-4447-4024-A000-EDE09922891E</RequestId>
    <Success>true</Success>
    <MessageId>889455942124347329</MessageId>
  </PubResponse>
```

2.12.2 PubBroadcast

调用该接口向订阅了指定Topic的所有设备发布广播消息。

限制说明

服务器SDK每秒只可发送一条广播消息。消息发送到物联网平台后，可以被多个设备订阅接收。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：PubBroadcast。
ProductKey	String	是	要发送广播消息的产品Key。

名称	类型	是否必需	描述
TopicFullName	String	是	要接收广播消息的Topic全称。格式为：/broadcast/\${productKey}/自定义字段。其中，\${productKey}是要接收广播消息的具体产品Key；自定义字段中您可以指定任意字段。  说明： - 广播Topic是在设备开发时编码定义的，无需控制台创建。 - 一个广播Topic最多可被1,000个设备订阅。 如果您的设备超过数量限制，您可以对设备进行分组。例如，如果您有5,000个设备，您可以将设备按每组1,000个，而分成5组。您需要分5次调用广播Topic，自定义字段分别设置为group1/2/3/4/5，然后让每组设备分别订阅各自分组的广播Topic。
MessageContent	String	是	要发送的消息主体。您需要将消息原文转换成二进制数据，并进行Base64编码，从而生成消息主体。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
MessageId	String	成功发送消息后，云端生成的消息ID，用于标识该消息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=PubBroadcast
&ProductKey=a1*****
&TopicFullName=%252Fbroadcast%252FUPqSxj2vXXX%252Fxxx
&MessageContent=aGVsbG93b3JsZA==
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "BB71E443-4447-4024-A000-EDE09922891E",
  "Success": true,
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
  <PubBroadcastResponse>
    <RequestId>BB71E443-4447-4024-A000-EDE09922891E</RequestId>
    <Success>true</Success>
  </PubBroadcastResponse>
```

2.12.3 RRpc

调用该接口向指定设备发送请求消息，并同步返回响应。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：RRpc。
ProductKey	String	是	要发送消息的产品Key。
DeviceName	String	是	要接收消息的设备名称。
RequestBase64Byte	String	是	要发送的请求消息内容经过Base64编码得到的字符串格式数据。
Timeout	Integer	是	等待设备回复消息的时间，单位是毫秒，取值范围是1,000 ~5,000。
Topic	String	否	使用自定义的RRPC相关Topic。需要设备端配合使用，请参见设备端开发 自定义Topic 。 不传入此参数，则使用系统默认的RRPC Topic。

名称	类型	是否必需	描述
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。
MessageId	String	成功发送请求消息后，云端生成的消息ID，用于标识该消息。
RrpcCode	String	调用成功时，生成的调用返回码，标识请求状态。取值： UNKNOW：系统异常 SUCCESS：成功 TIMEOUT：设备响应超时 OFFLINE：设备离线 HALFCONN：设备离线（设备连接断开，但是断开时间未超过一个心跳周期）
PayloadBase64Byte	String	设备返回结果Base64编码后的值。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=RRpc
&ProductKey=a1*****
&DeviceName=device1
&RequestBase64Byte=aGVsbG8gd29ybGQ%3D
&TimeOut=1000
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "41C4265E-F05D-4E2E-AB09-E031F501AF7F",
  "Success": true,
  "RpcCode": "SUCCESS",
  "PayloadBase64Byte": "d29ybGQgaGVsbG8="
  "MessageId": 889455942124347392
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
  <RRpcResponse>
    <RequestId>41C4265E-F05D-4E2E-AB09-E031F501AF7F</RequestId>
    <Success>true</Success>
    <RpcCode>SUCCESS</RpcCode>
    <PayloadBase64Byte>d29ybGQgaGVsbG8=</PayloadBase64Byte>
    <MessageId>889455942124347392</MessageId>
  </RRpcResponse>
```

2.13 设备影子

2.13.1 GetDeviceShadow

调用该接口查询指定设备的影子信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：GetDeviceShadow。
ProductKey	String	是	要查询的设备所隶属的产品Key。
DeviceName	String	是	要查询的设备名称。
公共请求参数	-	是	请参见 公共参数 。

返回参数

名称	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

名称	类型	描述
ShadowMessage	String	调用成功时，返回的设备影子信息。  说明： 根据影子设备的不同状态，查询到的影子信息结构有所差别，详情请参考设备影子开发。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=GetDeviceShadow
&ProductKey=a1*****
&DeviceName=device1
&公共请求参数
```

返回示例

· JSON格式

```
{
  "RequestId": "BB71E443-4447-4024-A000-EDE09922891E",
  "Success": true,
  "ShadowMessage": { "method": "update", "state": { "desired": { "color": "green" }, "reported": "" }, "version": 1 }
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
  <GetDeviceShadowResponse>
    <RequestId>BB71E443-4447-4024-A000-EDE09922891E</RequestId>
    <Success>true</Success>
    <ShadowMessage>{ "method": "update", "state": { "desired": { "color": "green" }, "reported": "" }, "version": 1 }</ShadowMessage>
  </GetDeviceShadowResponse>
```

2.13.2 UpdateDeviceShadow

调用该接口修改指定设备的影子信息。

请求参数

名称	类型	是否必需	描述
Action	String	是	要执行的操作，取值：UpdateDeviceShadow。
ProductKey	String	是	要修改影子信息的设备所隶属的产品Key。
DeviceName	String	是	要修改影子信息的设备名称。

名称	类型	是否必需	描述
ShadowMessage	String	是	修改后的设备影子信息。示例如下： <pre>{ "method": "update", "state": { "desired": { "color": "green" } }, "version": 2 }</pre> 详情参见 ShadowMessage 。
公共请求参数	-	是	请参见 公共参数 。

表 2-87: ShadowMessage

名称	类型	是否必需	描述
method	String	是	指定操作类型，取值：update。
state	String	是	设备发送给shadow的具体状态。其中，desired部分是期望的影子状态。
version	String	是	设备影子的版本，必须大于当前影子版本。

返回参数

参数	类型	描述
RequestId	String	阿里云为该请求生成的唯一标识符。
Success	Boolean	表示是否调用成功。true表示调用成功，false表示调用失败。
ErrorMessage	String	调用失败时，返回的出错信息。
Code	String	调用失败时，返回的错误码。错误码详情，请参见 错误码 。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=UpdateDeviceShadow  
&ProductKey=a1*****  
&DeviceName=device1  
&ShadowMessage=[{"method":"update","state":{"desired":{"color":"green"},  
"reported":""},"version":1}]
```

&公共请求参数**返回示例****· JSON格式**

```
{
    "RequestId": "BB71E443-4447-4024-A000-EDE09922891E",
    "Success": true,
}
```

· XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
  <UpdateDeviceShadowResponse>
    <RequestId>BB71E443-4447-4024-A000-EDE09922891E</RequestId>
    <Success>true</Success>
  </UpdateDeviceShadowResponse>
```