阿里云
物联网平台

云端开发指南

文档版本：20180925

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**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

法律声明.....I
通用约定.....I
1 云端SDK参考.....1
1.1 SDK下载.....1
1.2 使用说明-Java SDK.....1
1.3 使用说明-Python SDK.....2
1.4 使用说明-PHP SDK.....4
1.5 使用说明-.NET SDK.....4
2 云端API参考.....7
2.1 API列表.....7
2.2 概述.....9
2.3 调用API.....10
2.4 公共参数.....11
2.5 签名机制.....13
2.6 错误码.....21
2.7 产品管理.....30
2.7.1 CreateProduct.....30
2.7.2 QueryProduct.....33
2.7.3 QueryProductList.....36
2.7.4 UpdateProduct.....39
2.7.5 DeleteProduct.....41
2.8 设备管理.....42
2.8.1 RegisterDevice.....42
2.8.2 QueryDeviceDetail.....45
2.8.3 QueryDevice.....48
2.8.4 DeleteDevice.....50
2.8.5 GetDeviceStatus.....52
2.8.6 BatchGetDeviceState.....54
2.8.7 DisableThing.....56
2.8.8 EnableThing.....57
2.8.9 BatchCheckDeviceNames.....59
2.8.10 BatchRegisterDeviceWithApplyId.....61
2.8.11 BatchRegisterDevice.....63
2.8.12 QueryBatchRegisterDeviceStatus.....64
2.8.13 QueryPageByApplyId.....66
2.8.14 QueryDeviceEventData.....68
2.8.15 QueryDevicePropertyData.....72
2.8.16 QueryDeviceServiceData.....76
2.8.17 InvokeThingService.....79

2.8.18 GetGatewayBySubDevice....81
2.8.19 QueryDevicePropertyStatus....84
2.8.20 SetDeviceProperty....87
2.8.21 QueryDeviceProp....89
2.8.22 SaveDeviceProp....91
2.8.23 DeleteDeviceProp....92
2.8.24 GetThingTopo....94
2.8.25 NotifyAddThingTopo....96
2.8.26 RemoveThingTopo....99
2.8.27 QueryDeviceStatistics....100
2.9 Topic管理....102
2.9.1 CreateProductTopic....102
2.9.2 UpdateProductTopic....104
2.9.3 QueryProductTopic....105
2.9.4 DeleteProductTopic....107
2.9.5 CreateTopicRouteTable....107
2.9.6 DeleteTopicRouteTable....109
2.9.7 QueryTopicReverseRouteTable....110
2.9.8 QueryTopicRouteTable....111
2.10 消息通信....112
2.10.1 Pub....112
2.10.2 PubBroadcast....114
2.10.3 RRpc....115
2.11 规则引擎....117
2.11.1 CreateRule....117
2.11.2 CreateRuleAction....120
2.11.3 DeleteRule....125
2.11.4 DeleteRuleAction....126
2.11.5 GetRule....127
2.11.6 GetRuleAction....129
2.11.7 ListRule....130
2.11.8 ListRuleActions....132
2.11.9 StartRule....134
2.11.10 StopRule....135
2.11.11 UpdateRule....136
2.11.12 UpdateRuleAction....138
2.12 设备影子....140
2.12.1 GetDeviceShadow....140
2.12.2 UpdateDeviceShadow....141

1 云端SDK参考

1.1 SDK下载

物联网平台提供的SDK语言版本有：Java、Python、PHP、和.Net。

IoT SDK 下载

单击以下链接，进入相应的SDK源码下载地址。

- *IoT Java SDK*
- *IoT Python SDK*
- *IoT PHP SDK*
- *IoT .Net SDK*

IoT SDK Demo 下载

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。

Demo下载地址：*IoT SDK Demo*

说明：

目前，SDK Demo只支持基础版API封装，暂不支持高级版API。

1.2 使用说明-Java SDK

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>5.0.0</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

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
PubResponse response = client.getAcsResponse(request);
System.out.println(response.getSuccess());
System.out.println(response.getErrorMessage());
```

1.3 使用说明-Python SDK

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。

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。

   以管理员权限执以下命令，安装IoTPython SDK。

```
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 aliyunsdkiot.request.v20180120 import RegisterDeviceRequest
from aliyunsdkiot.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 使用说明-PHP SDK

使用物联网平台的PHP SDK便捷地对物联网平台进行操作。

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

```
<?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

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 使用说明-.NET SDK

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. 引入阿里云IoT .NET SDK核心库。

   从*.NET SDK发布列表* 中下载IoT .NET SDK核心库的压缩包，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DLL文件（*aliyun-net-sdk-Iot.dll*），单击**确认**。

初始化SDK

```
using Aliyun.Acs.Core;
using Aliyun.Acs.Core.Exceptions;
using Aliyun.Acs.Core.Profile;
IClientProfile clientProfile = DefaultProfile.GetProfile("<your-region
-id>", "<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

以调用**Pub**接口向Topic发布消息为例。

```
PubRequest request = new PubRequest();
request.ProductKey = "<productKey>";
request.TopicFullName = "/" + request.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。

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。
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	获取设备的事件历史数据。
QueryDevicePropertyData	获取设备的属性历史数据。

API	描述
QueryDeviceServiceData	获取设备的服务记录历史数据。
InvokeThingService	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。
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。

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。
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

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
```

其中：

- **Endpoint**：调用的云服务的接入地址。物联网平台的接入地址分别是：
  - 华东2（上海）：`iot.cn-shanghai.aliyuncs.com`
  - 新加坡：`iot.ap-southeast-1.aliyuncs.com`
  - 美国（硅谷）：`iot.us-west-1.aliyuncs.com`
  - 日本（东京）：`iot.ap-northeast-1.aliyuncs.com`
  - 德国（法兰克福）：`iot.eu-central-1.aliyuncs.com`
- **Action**：要执行的操作。如使用**Pub**接口向指定Topic发布消息。
- **Parameters**：请求参数。每个参数之间用（&）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。
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。
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

名称	类型	是否必需	描述
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

```
<?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>
```

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 *Access Key* 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*SDK*下载及各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=12345abcde
Z&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%26Signatur
eMethod%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 = "12345678901234567890
1234567890";

 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 UnsupportedEncodingE
xception {

  // 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。

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
iot.system.SystemException	系统异常。

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
iot.common.InvalidPageParams	分页大小或者分页页号不合法。
iot.common.InvalidTenant	不合法的租户。
iot.common.QueryDeviceActionError	查询设备失败。
iot.common.QueryDevicePropertyActionError	查询设备属性失败。
iot.common.QueryProductActionError	查询产品失败。
iot.common.QueryProductCountActionError	查询产品总数失败。
iot.common.RamActionPermissionDeny	没有资源访问控制（RAM）权限。

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。
iot.prod.CreateProductFailed	创建产品失败。
iot.prod.CreateProductTopicFailed	创建产品的Topic类失败。
iot.prod.InvalidAliyunCommodityCode	入参AliyunCommodityCode（IoT 平台版本）错误。
iot.prod.InvalidFormattedCatId	入参产品的设备类型格式错误。

错误码	描述
iot.prod.InvalidFormattedProductkey	入参产品ID格式错误。
iot.prod.InvalidFormattedProductName	入参产品名称格式错误。
iot.prod.LongProductDesc	产品描述字符数超出限定值。
iot.prod.InvalidNodeType	产品的节点类型错误。
iot.prod.NotExistedProduct	产品不存在。
iot.prod.NotOpenID2Service	没有开通ID2服务。
iot.prod.NotSeniorProduct	产品不是高级版产品。
iot.prod.NullProductKey	入参产品ID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。
iot.prod.QueryDeviceCountActionError	查询产品下的设备总数失败。
iot.prod.QueryProductAbilitiesFailed	获取产品功能失败。
iot.prod.QueryProductAbilityFailed	查询产品功能失败。
iot.prod.QueryProductListActionError	获取产品列表数据失败。
iot.prod.UpdateProductFailed	更新产品信息失败。

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。
iot.device.AlreadyExistedDeviceName	设备名称已经存在。
iot.device.ApplyManyDevicesFailed	申请批量创建设备失败。
iot.device.CreateDeviceFailed	创建设备失败。
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。
iot.device.DeviceCountExceeded	批量申请的设备数量超过最大值。
iot.device.DeleteDeviceFailed	删除设备失败。
iot.device.DeleteDevicePropertyFailed	删除设备属性失败。

错误码	描述
iot.device.DisableDeviceFailed	禁用设备失败。
iot.device.EnableDeviceFailed	启用设备失败。
iot.device.InactiveDevice	设备未激活。
iot.device.InvalidFormattedApplyId	创建设备的申请单ID值错误。
iot.device.IncorrentDeviceApplyInfo	设备申请信息错误。
iot.device.InvalidFormattedDeviceName	设备名称格式错误。
iot.device.InvalidFormattedDevicePropertyKey	设备属性名称格式错误。
iot.device.InvalidFormattedDevicePropertiesString	入参设备属性格式错误。
iot.device.InvalidIoTId	设备ID错误。
iot.device.InvalidTimeBucket	指定的时间区间不合法。
iot.device.InvokeThingServiceFailed	调用设备服务失败。
iot.device.LongDevicePropertiesString	入参设备属性长度超过最大值。
iot.device.NoneDeviceNameElement	设备名称列表为空。
iot.device.NoneDeviceProperties	没有有效的设备属性。
iot.device.NotExistedDevice	设备不存在。
iot.device.NullApplyId	创建设备的申请ID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。
iot.device.QueryDeviceAttrDataHistoryFailed	获取设备属性数据历史记录失败。
iot.device.QueryDeviceAttrStatusFailed	获取设备属性状态信息失败。
iot.device.QueryDeviceEventHistoryFailed	获取设备事件调用记录失败。
iot.device.QueryDeviceListActionError	查询设备列表失败。
iot.device.QueryDeviceServiceHistoryFailed	获取设备服务调用记录失败。
iot.device.QueryDeviceStatisticsFailed	查询设备统计信息失败。
iot.device.QueryDeviceStatusFailed	查询设备状态信息失败。

错误码	描述
iot.device.QueryTopoRelationFailed	查询拓扑关系失败。
iot.device.RemoveTopoRelationFailed	移除拓扑关系失败。
iot.device.SaveOrUpdateDevicePropertiesFailed	新增或者修改设备属性失败。
iot.device.SetDevicePropertyFailed	设置设备属性失败。
iot.device.TooManyDevicePropertiesPerTime	新增的设备属性个数加上已有属性个数超过限定值。
iot.device.TopoRelationCountExceeded	拓扑关系数量过多。
iot.device.VerifyDeviceFailed	验证设备失败。

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。
iot.messagebroker.CreateTopicTemplateException	创建Topic类过程发生异常。
iot.messagebroker.CreateTopicTemplateFailed	创建Topic类失败。
iot.messagebroker.DeleteTopicTemplateException	删除Topic类过程发生异常。
iot.messagebroker.DeleteTopicTemplateFailed	删除Topic类失败。
iot.messagebroker.DestTopicNameArraySizeIsLarge	同一消息源Topic配置的路由目标Topic个数量超过最大限制数。
iot.messagebroker.DeleteTopicRouteFailed	删除指定Topic间的路由失败。
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名称或者属性值。

错误码	描述
iot.messagebroker.ElementKeyOrValueIsNullInReport	**report**信息包含有空的属性名称或者属性值。
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。
iot.messagebroker.InvalidFormattedTopicName	Topic格式错误。
iot.messagebroker.InvalidFormattedTopicTemplateId	Topic类ID格式错误。
iot.messagebroker.InvalidTimeoutValue	超时时间参数设置有误。
iot.messagebroker.InvalidTopicTemplateOperationValue	Topic类的操作权限值错误。
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。
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。

错误码	描述
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。
iot.messagebroker.PublishMessageFailed	发送消息失败。
iot.messagebroker.QueryDeviceShadowActionError	查询设备影子失败。
iot.messagebroker.QueryProductTopicListActionError	获取Topic类列表失败。
iot.messageborker.QueryTopicReverseRouteTableListActionError	获取消息反向路由列表（即消息源Topic列表）失败。
iot.messageborker.QueryTopicRouteTableListActionError	获取消息路由列表失败。
iot.messagebroker.QueryTopicTemplateActionError	查询Topic类失败。
iot.messagebroker.QueryTopicTemplateException	获取Topic类过程发生异常。
iot.messagebroker.RateLimit	由于限流导致失败。
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。
iot.messagebroker.RrpcFailed	RRPC发送消息失败。
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。
iot.messagebroker.TIMEOUT	由于超时导致失败。

错误码	描述
iot.messagebroker.TooManyElementInDesire	**desire**信息中包含的属性总数超过最大限定数。
iot.messagebroker.TooManyElementInReport	**report**信息包含的属性总数不能限定最大数。
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。
iot.messagebroker.TopicTemplateIsNotFound	Topic类不存在。
iot.messagebroker.UpdateDeviceShadowMessageFailed	更新设备影子失败。
iot.messagebroker.UpdateTopicTemplateException	更新Topic类过程发生异常。
iot.messagebroker.UpdateTopicTemplateFailed	更新Topic类失败。

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。

错误码	描述
iot.rule.CreateRuleException	创建规则过程发生异常。
iot.rule.DeleteRuleFailed	删除规则失败。
iot.rule.IncorrentRuleActionId	规则动作ID错误。
iot.rule.IncorrentRuleActionType	规则动作类型错误。
iot.rule.IncorrentRuleId	规则ID错误。
iot.rule.NullForwardDestForRule	转发数据目的地不能为空。
iot.rule.NullSqlForRule	规则的SQL语句不能为空。
iot.rule.NotFoundRule	规则不存在。
iot.rule.NotFoundRuleAction	规则动作不存在。
iot.rule.ParseRuleActionConfigError	无法正常解析规则动作的配置。
iot.rule.QueryRuleActionListError	查询规则动作列表失败。
iot.rule.QueryRulePageActionError	分页获取规则列表失败。

错误码	描述
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。
iot.rule.RuleNameIsAlreadyExisted	规则名称已经存在。
iot.rule.StartRuleFailed	启动规则失败。
iot.rule.StopRuleFailed	停止规则失败。
iot.rule.TooManyRuleAction	规则动作数量超过最大限制。
iot.rule.UpdateRuleFailed	更新规则失败。
iot.ruleng.CreateRuleActionFailed	创建规则动作失败。
iot.ruleng.DeleteRuleActionFailed	删除规则动作失败。
iot.ruleng.IncorrectType	应用规则的Topic类型错误。
iot.ruleng.IncorrectSysTopic	错误的系统Topic。
iot.ruleng.InvalidRamRole	非法的RAM角色。
iot.ruleng.QueryRuleActionFailed	获取规则动作失败。
iot.ruleng.RuleActionConfigurationIsNotJson	规则动作配置不是JSON格式。
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	规则动作配置不能为空。
iot.ruleng.NullRuleActionType	规则动作类型不能为空。
iot.messagebroker.IncorrectRuleSql	规则的SQL配置错误。
iot.messagebroker.QueryRuleConfigActionException	获取规则配置信息过程出现异常。

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	主题格式错误。
iot.prod.NotExistedProduct	产品不存在。

错误码	描述
iot.common.QueryProductActionError	查询产品失败。
iot.ruleng.IncorrectSysTopic	系统Topic错误。
iot.messagebroker.NullTopicName	Topic名称不能为空。

表 **2-2:** 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。

表 **2-3:** 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。
iot.ruleng.QueryMnsTopicListActionError	获取消息服务主题列表失败。

表 **2-4:** 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。

表 **2-5:** 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。
NodeType	Integer	否	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。 如果不传入该参数，则使用默认值0。
DataFormat	Integer	否	高级版产品的数据格式： • 0：透传/自定义格式（CUSTOM_FORMAT）。 • 1：Alink协议（ALINK_FORMAT）。 此参数为创建高级版产品的特有参数。
CategoryId	Long	否	设备类型。 高级版特有参数，可以为空。
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。
AliyunCommodityCode	String	否	产品版本类型。 • iothub_senior：高级版。 • iothub：基础版。 若不传入此参数默，则默认为基础版。

名称	类型	是否必需	描述
ProtocolType	String	否	设备接入网关协议类型。 • modbus：Modbus协议。 • opc-ua：OPC UA协议。 • customize：自定义协议。 此参数为高级版产品特有参数，可以为空。
Id2	Boolean	否	是否为ID²产品。(本功能暂未开放)
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。
ProductKey	String	产品ID。
Data	Data	调用成功时返回的新建产品信息。详情参见表 2-6: ProductInfo。

表 2-6: ProductInfo

名称	类型	描述
ProductName	String	新建产品的名称。
ProductKey	String	IoT为新建产品颁发的产品ID，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）。

名称	类型	描述
		• 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。
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。 此参数为高级版产品的特有参数。
Id2	Boolean	是否为ID²产品。

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

```
{
      RequestId:8AE93DAB-958F-49BD-BE45-41353C6DEBCE,
      Success:true,
      ProductKey:a1QDKAU****,
      Data:{
          ProductKey:a1QDKAU****,
          DataFormat:1,
   NodeType:0,
          ProtocolType:modbus,
          ProductName:Test,
```

```
    Description:Product test,
           AliyunCommodityCode:iothub_senior
       }
  }       }
  }
```

- XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
  <CreateProductResponse>
      <RequestId>8AE93DAB-958F-49BD-BE45-41353C6DEBCE</RequestId>
      <ProductKey>a1QDKAU****</ProductKey>
      <Data>
         <Description>Pruduct test</Description>
         <DataFormat>1</DataFormat>
         <ProductKey>a1QDKAU****</ProductKey>
         <ProtocolType>modbus</ProtocolType>
         <NodeType>0</NodeType>
         <ProductName>Test</ProductName>
         <AliyunCommodityCode>iothub_senior</AliyunCommodityCode>
      </Data>
      <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。

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。
Data	Data	调用成功时，返回的数据。详情见表格Data。

表 2-7: 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ID。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	产品秘钥。

名称	类型	描述
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。
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	是否为ID2产品（目前尚未开放）。

示例

请求示例

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

返回示例

- JSON格式

```
{
    "RequestId":"57b144cf-09fc-4916-a272-a62902d5b207",
    "Success": true ,
    "Data": {
        "DataFormat": 1,
        "ProductKey": "a1*******",
        "NodeType": 0,
        "ProductName": "iot_test",
 "ProductSecret":"4alWjhgiTG****"
 "CategoryName": "",
 "CategoryKey": "",
        "DeviceCount": 7221,
        "GmtCreate": 1516534408000,
 "AliyunCommodityCode":"iothub_senior",
        "Description": "Test"
    }
}
```

- XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
<QueryProductResponse>
    <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
    <Success>true</Success>
    <Data>
        <DataFormat>1</DataFormat>
        <ProductKey>a1*********</ProductKey>
        <NodeType>0</NodeType>
        <ProductName>iot_test</ProducName>
```

```
 <ProductSecret>4alWjhgiTGU*****</Productecret>
 <CategoryName></CategryName>
 <CategoryKey></CategoryKey>
        <DeviceCount>7221</DeviceCount>
        <GmtCreate>1516534408000</GmtCreate>
        <AliyunCommodityCode>iothub_senior</AliyunCommodityCode>
        <Description>Test</Description>
    </Data>
</QueryProductResponse>
```

2.7.3 QueryProductList

调用该接口查看所有产品列表。

限制说明

该接口可用于查看当前账号下所有的IoT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：QueryProductList。
CurrentPage	Integer	是	指定从返回结果中的第几页开始显示。
PageSize	Integer	是	指定返回结果中每页显示的记录数量，最大值是200。
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	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。

名称	类型	描述
ErrorMessage	String	调用失败时，返回的出错信息。
Data	Data	调用成功时，返回的产品列表。详情参见下表Data。

表 2-8: Data

名称	类型	描述
PageSize	Integer	每页显示的记录数。
PageCount	Integer	总页数。
CurrentPage	Integer	当前页号。
Total	Integer	总记录数。
List	List	产品信息列表。请参见下表*ProductInfo*。

表 2-9: 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。
ProductKey	String	产品的Key。新建产品时，IoT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
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": 4,
          "PageSize": 20,
          "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": 79
      }
```

```
  }
```

- XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
 <QueryProductListResponse>
      <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
      <Success>true</Success>
      <Data>
          <PageCount>4</PageCount>
          <PageSize>20</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>79</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ID。
ProductName	String	否	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）。

参数	类型	是否必需	描述
			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。

示例

请求示例

```
http://iot.cn-shanghai.aliyuncs.com/?Action=UpdateProduct
&ProductKey=al*********
&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ID（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ID。
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	错误码。具体信息请参见错误码。

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>
```

2.8 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

注册设备指在产品下新建设备。在指定产品下成功注册设备后，阿里云IoT平台为注册的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IoT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。请参见*BatchRegisterDevice*。
- 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。请参见*BatchCheckDeviceNames*和*BatchRegisterDeviceWithApplyId*。

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	指定要为其注册设备的产品的ID。
DeviceName	String	否	为要注册的设备命名。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

名称	类型	是否必需	描述
			符（连字符、下划线、@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。
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。
Data	Data	调用成功时，返回注册的设备信息。详情参见*DeviceInfo*。

表 2-10: DeviceInfo

名称	类型	描述
ProductKey	String	设备隶属的产品ID。
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	IoT平台为该设备颁发的设备ID，作为该设备的唯一标识符。

名称	类型	描述
		说明： 请妥善保管该信息。

示例

请求示例

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

返回示例

- JSON格式

```
{
    "RequestId":"57b144cf-09fc-4916-a272-a62902d5b207",
    "Success": true ,
    "Data": {
        "DeviceName": "CqXL5h5ysRTA4NxjABjj",
        "ProductKey": "a1*********",
        "DeviceSecret": "tXHf4ezGEHcwdyMwoCDHGBmk9avi****",
        "IotId": "CqXL5h5ysRTA4NxjABjj0010fa****"
    }
}
```

- XML格式

```
<?xml version='1.0' encoding='utf-8'?>
<RegisterDeviceResponse>
    <RequestId>57b144cf-09fc-4916-a272-a62902d5b207</RequestId>
    <Success>true</Success>
    <Data>
        <DeviceName>CqXL5h5ysRTA4Nxj****</DeviceName>
        <ProductKey>a1rYuVF****</ProductKey>
        <DeviceSecret>tXHf4ezGEHcwdyMwoCDHGBmk9avi****</DeviceSecret
>
        <IotId>CqXL5h5ysRTA4NxjABjj0010fa****</IotId>
    </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。
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ID。 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。

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。
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 **2-11: Data**

名称	类型	描述
ProductKey	String	设备隶属的产品ID。
ProductName	String	设备隶属的产品名称。
DeviceName	String	设备名称。
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。

名称	类型	描述
		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。
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=a1rYuV*****
&DeviceName=SR8FiTu1R9tlUR2V1bmi
&公共请求参数
```

返回示例

- JSON格式

```
{
  "RequestId":"57b144cf-09fc-4916-a272-a62902d5b207",
  "Success": true ,
  "Data": {
    "DeviceName": "SR8FiTu1R9tlUR2V1bmi",
    "GmtActive": "2018-08-06 10:48:41",
    "UtcActive": "2018-08-06T02:48:41.000Z",
    "ProductKey": "a1rYuV*****",
    "DeviceSecret": "CPwUjMUgzdvaZv56TMy6773V3v3****",
    "GmtCreate": "2018-08-06 10:47:50",
    "UtcCreate": "2018-08-06T02:47:50.000Z",
    "IotId": "SR8FiTu1R9tlUR2V1bmi0010*****",
    "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>SR8FiTu1R9tlUR2V1bmi</DeviceName>
        <GmtActive>2018-08-06 10:48:41</GmtActve>
 <UtcActive>2018-08-06T02:48:41.000Z</UtcActive>
```

```
        <ProductKey>a1rYuVF****</ProductKey>
        <DeviceSecret>CPwUjMUgzdvaZv56TMy6773V3v3W****</DeviceSecret
>
        <GmtCreate>2018-08-06 10:47:50</GmtCreate>
 <UtcCreate>2018-08-06T02:47:50.000Z</UtcCreate>
        <IotId>SR8FiTu1R9tlUR2V1bmi0010a5****</IotId>
        <Status>ONLINE</Status>
        <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

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	是否必须需	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ID。
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。如默认值是 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。
PageCount	Integer	总页数。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页面号。

名称	类型	描述
Total	Integer	总记录数。
Data	Data	调用成功时，返回设备信息列表。详情参见*DeviceInfo*。

表 2-12: 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。如果您想要查找具体设备的IotId（即有效的设备ID，设备的唯一标识），您可以调用**QueryDeviceDetail**接口，传入**ProductKey**和**DeviceName**组合。
DeviceName	String	设备名称。
DeviceSecret	String	设备密钥。
GmtCreate	String	设备创建时间。
GmtModified	String	设备信息修改时间。

示例

请求示例

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

返回示例

- JSON格式

```
{
      PageCount:1,
      Data:{
          DeviceInfo:[
              {DeviceId:SR8FiTu1R9tlUR2V1bmi0010*****, DeviceName:
device1, DeviceSecret:CPwUjMUgzdvaZv56TMy6773V3v3****, GmtCreate:Thu
, 17-Nov-2016 02:08:12 GMT, GmtModified:Thu, 17-Nov-2016 02:08:12
GMT}
          ]
```

```
        },
        PageSize:10,
        Page:1,
        Total:9
        RequestId:06DC77A0-4622-42DB-9EE0-26A6E1FA08D3,
        Success:true,
    }
```

- XML格式

```
<?xml version='1.0' encoding='utf-8'?>
 <QueryDeviceResponse>
      <RequestId>9D25B7F9-DF08-4F77-B1E9-008F78522B37</RequestId>
      <Success>true</Success>
      <Page>1</Page>
      <PageSize>10</PageSize>
      <PageCount>1</PageCount>
      <Total>6</Total>
      <Data>
          <DeviceInfo>
              <DeviceId>SR8FiTu1R9tlUR2V1bmi0010*****</DeviceId>
              <DeviceName>device1</DeviceName>
              <DeviceSecret>CPwUjMUgzdvaZv56TMy6773V3v3****</
DeviceSecret>
              <GmtCreate>Thu, 10-Nov-2016 05:35:56 GMT</GmtCreate>
              <GmtModified>Thu, 10-Nov-2016 05:35:56 GMT</GmtModifie
d>
          </DeviceInfo>
          ...
      </Data>
  </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

名称	类型	是否必需	描述
			一一对应的关系。如果您同时传入**IotId**和**ProductKey&DeviceName**组合，则以**IotId**为准。
ProductKey	String	否	要删除的设备所隶属的产品ID。 说明： 如果传入该参数，需同时传入**DeviceName**。
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。

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>
```

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&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ID。 **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。
Data	Data	调用成功时，返回设备状态信息。详情参见*DeviceStatusInfo*。

表 **2-13: 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	是	要查看运行状态的设备所隶属的产品ID。
DeviceName	List<String>	是	要查看运行状态的设备的名称列表。 说明： 支持单次查询最多50个设备。
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。
DeviceStatusList	DeviceStatus	调用成功时，返回设备状态信息列表。详情参见*DeviceStatus*。

表 2-14: DeviceStatus

名称	类型	描述
DeviceName	String	设备名称。

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=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, DeviceName:device1},
              {Status:UNACTIVE, DeviceName:device2}
          ]
      },
      RequestId:"1A540BD7-176C-42D4-B3C0-A2C549DD00A3",
      Success:true
  }
```

- XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
  <BatchGetDeviceStateResponse>
      <RequestId>1AB5E6B0-AFCB-47B1-B6D4-C2BD32D63E14</RequestId>
      <Success>true</Success>
      <DeviceStatusList>
          <DeviceStatus>
           <Status>UNACTIVE</Status>
           <DeviceName>device1</DeviceName>
          </DeviceStatus>
          <DeviceStatus>
           <Status>UNACTIVE</Status>
           <DeviceName>device2</DeviceName>
          </DeviceStatus>
      </DeviceStatusList>
```

```
</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ID。 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。

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。

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

名称	类型	是否必需	描述
			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	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。

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。

限制说明

该接口需要和**BatchRegisterDeviceWithApplyId**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。单次调用，最多能传入1,000 个设备名称。

首先，您必须先调用本接口，传入要批量注册的设备名称。IoT平台检查您提交的设备名称符合要求后，为您返回申请批次ID（**ApplyId**）。然后，您可以通过使用**ApplyId**调用**BatchRegisterDeviceWithApplyId**接口，批量注册设备。如果设备名称列表中包含不合法的设备名称，批量申请设备失败。可调用QueryBatchRegisterDeviceStatus 接口检查，其返回结果中会提示不合法的设备名称列表。

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。

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。

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：BatchCheck DeviceNames。
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。

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。
Data	Data	调用成功时，返回的数据。详情见下表Data。

表 **2-15: 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=al**********
&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。

限制说明

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。

首先，您必须调用**BatchCheckDeviceNames**接口，传入要批量注册的设备的名称。IoT平台检查您提交的设备名称符合要求后，为您返回申请批次ID（ApplyId）。然后，您可以通过使用ApplyId调用本接口，批量注册设备。

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。

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。

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。
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。
Data	Data	调用成功时，返回的数据。详情见下表Data。

表 **2-16: 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=BatchRegisterDeviceW
ithApplyId
&ProductKey=al*********
&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>
```

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

您可以调用本接口，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。

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ID。
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。
Data	Data	调用成功时，返回的数据。详情见下表Data。

表 **2-17: 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=al*********
&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 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：QueryBatch RegisterDeviceStatus。
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**BatchRegist**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。
Data	DeviceApplyStatusInfo	调用成功时返回的申请单信息。详情参见*DeviceApplyStatusInfo*。

表 **2-18: 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	创建成功的设备名称（DeviceName）集合。
InvalidList	List	创建失败的设备名称（DeviceName）集合。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=QueryBatchRegisterDeviceStatus
&ProductKey=al*********
&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>
```

2.8.13 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**BatchRegiste**口返回结果中查看。
PageSize	Integer	是	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。
CurrentPage	Integer	是	指定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。
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	*ApplyDeviceInfo*	调用成功时生成的已注册的设备列表。详情参见下表。

表 2-19: 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。如果您想要查找具体设备的IotId（即有效的设备ID，设备的唯一标识），您可以调用**QueryDeviceDetail**接口，传入**ProductKey**和**DeviceName**，即可在返回结果中查看有效的IotId。其中设备名称（**DeviceName**）可在本接口的返回结果中查看。
DeviceName	String	设备名称。
DeviceSecret	String	设备密钥。 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

```
{
    "RequestId":"BB71E443-4447-4024-A000-EDE09922891E",
    "Success":true,
    Page:1,
    PageSize:10,
    PageCount:1,
    Total:4,
    ApplyDeviceList:{
        ApplyDeviceInfo:[
            {DeviceId:abcdef1234*************,DeviceName:device1,
DeviceSecret:abcdefsddd},
            {DeviceId:edcba4321*************,DeviceName:device2,
DeviceSecret:daffdfdeaa}
            ]
    }
}
```

2.8.14 QueryDeviceEventData

调用该接口查询指定设备的事件记录。

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&DeviceName**组合，则以**IotId**为准。
ProductKey	String	否	要查询的设备所隶属的产品ID。

名称	类型	是否必需	描述
			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**。
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。
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。
Data	Data	调用成功时，返回的设备事件记录。详情参见*Data*。

表 2-20: 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	表示下一页面是否可用。true表示可用，false表示不可用。
NextTime	Long	下一页面中的事件记录的起始时间。

表 2-21: 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 2-22: 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\":{\"structchildFLOATf71c
20e\":1.23,\"structchildINT6b6b626\":3,\"structchildDATE663436a\":
\"1516517966152\",\"structchildDOUBLE08d0f74\":1.23,\"structchil
dTEXTdc764f9\":\"07b68264b0ba42c18e5f\",\"structchildBOOLd260729
\":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\":{\"structchildFLOATf71c
20e\":1.23,\"structchildINT6b6b626\":3,\"structchildDATE663436a\":
\"1516449212507\",\"structchildDOUBLE08d0f74\":1.23,\"structchil
dTEXTdc764f9\":\"a1f3583dde3944289639\",\"structchildBOOLd260729
\":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":{"structchildFLOATf71c
20e":1.23,"structchildINT6b6b626":3,"structchildDATE663436a":"
```

```
1516517966152","structchildDOUBLE08d0f74":1.23,"structchildTEXTdc764
f9":"07b68264b0ba42c18e5f","structchildBOOLd260729":0,"structchil
dENUMbe62590":1},"enumArgs":0,"boolArgs":0,"floatArgs":2.3,"
dateArgs":"1516517966152","intArgs":1,"doubleArgs":2.3,"textArgs":"
dV56zbkzjBjw1Ti1dA52"}</OutputData>
                <EventType>info</EventType>
                <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","structchildBOOLd260729":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>
```


2.8.15 QueryDevicePropertyData

调用该接口查询指定设备的属性记录。

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。
ProductKey	String	否	指定要查询的设备隶属的产品ID。 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。 高级版设备的属性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。 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。
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。
Data	*Data*	调用成功时，返回的设备属性记录。详情参见下表。

表 2-23: 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*PropertyInfo*。
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	下一页面中的属性记录的起始时间。

表 2-24: 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=al*********
&DeviceName=device1
&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": 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>
```

2.8.16 QueryDeviceServiceData

调用该接口查询指定设备的服务记录。

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ID。 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。
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。

名称	类型	是否必需	描述
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。
Data	Data	调用成功时，返回的设备服务记录。详情参见*Data*。

表 2-25: Data

名称	类型	描述
List	List<ServiceInfo>	事件集合。每个元素代表一个服务。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 2-26: 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=al**********
&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>
```

2.8.17 InvokeThingService

调用该接口在一个设备上执行指定的服务。

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	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	设备所隶属的产品ID。 说明： 如果传入该参数，需同时传入**DeviceName**。
DeviceName	String	否	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。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。
Args	Map	是	要启用服务的入参信息，如，Args={"param1":1}（实际使用时，参数值需经URL编码，见请求示例）。Args详情参见下表*Args*。
公共请求参数	-	是	请参见公共参数。

表 **2-27: 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高级版产品的**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 **2-28: Data**

名称	类型	描述
Result	String	同步调用服务返回的调用结果。
MessageId	String	云端向设备下发执行服务的消息ID。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=InvokeThingService
&ProductKey=al*********
&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>
```

2.8.18 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：GetGateway BySubDevice。
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

名称	类型	是否必需	描述
			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ID。 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。
Data	*DeviceDetailInfo*	调用成功时，返回的网关设备的详细信息。详情请参见下表。

表 2-29: DeviceDetailInfo

名称	类型	描述
ProductKey	String	网关设备隶属的产品ID。
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。

名称	类型	描述
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。
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=al*********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": "nICOJkFJnG***********TWnvXHydEjX",
        "UtcOnline": "",
        "IotId": "9a1MTdk9brqQ2bhdG4OY001094fd00",
        "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>nICOJkFJnG**********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>
```

2.8.19 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

名称	类型	是否必需	描述
			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**。
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。
Data	Data	调用成功时，返回的数据。详情参见下表。

表 2-30: 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 2-31: PropertyStatusInfo

名称	类型	描述
Identifier	String	属性标识符。

名称	类型	描述
Name	String	属性名称。
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>
```

2.8.20 SetDeviceProperty

调用该接口设置设备的属性。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的请求，设备属性才能真正设置成功。

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ID。 说明：

名称	类型	是否必需	描述
			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**。
Items	Map	是	要设置的属性信息，详情参见*Item*。
公共请求参数	-	是	请参见公共参数。

表 2-32: Item

名称	类型	描述
Key	String	要设置的属性的标识符（identifier）。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。 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。取值需跟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。
Data	Data	调用成功时，返回的数据。详情请参见下表Data。

表 2-33: 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=al*********
&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>
```

2.8.21 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。
ProductKey	String	是	要查询的设备所隶属的产品ID。
DeviceName	String	是	要查询的设备的名称。

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。
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

```
{
    "Props":"{"firmwareVersion":"12","color":"red"}",
    "RequestId":"F28BDA06-D8DB-4F70-BAED-DDA59AD6EF46",
    "Success":true
}
```

- XML格式

```
<?xml version='1.0' encoding='UTF-8'?>
<QueryDevicePropResponse>
    <RequestId>F28BDA06-D8DB-4F70-BAED-DDA59AD6EF46</RequestId>
    <Success>true</Success>
    <Props>{"firmwareVersion":"12","color":"red"}</Props>
```

```
</QueryDevicePropResponse>
```

2.8.22 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	是	要操作的设备所隶属的产品ID。
DeviceName	String	是	要操作的设备的名称。
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"}（取值需经URL编码处理，见请求示例）。 • 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 2-34: Prop

名称	类型	是否必需	描述
Key	String	是	设置标签键值。标签键值只能是英文，且长度在2-32字符之间。如果设置的键值已存在，则将覆盖之前的标签值；如果不存在，则新增标签。
Value	Object	是	标签值。

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。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=SaveDeviceProp
&ProductKey=al**********
&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>
```

2.8.23 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。
ProductKey	String	是	要操作的设备所隶属的产品ID。
DeviceName	String	是	要操作的设备的名称。

名称	类型	是否必需	描述
PropKey	String	是	要删除的设备属性键值（Key）。 **说明：** IoT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。

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>
```

2.8.24 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&DeviceName**组合，则以**IotId**为准。
ProductKey	String	否	要查询的设备所隶属的产品ID。 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。

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。
Data	Data	调用成功时，返回的设备信息集合。详情参见*DeviceDataInfo*。

表 2-35: 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。
Total	Integer	总记录数。

表 2-36: DeviceInfo

名称	类型	描述
IotId	String	设备的ID。
ProductKey	String	设备所隶属的产品ID。
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>
```

2.8.25 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**。
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 **2-37: DeviceList**

名称	类型	是否必需	描述
ProductKey	String	是	子设备所隶属的产品Key。
DeviceName	String	是	子设备名称。

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 **2-38: 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=...
&GwDeviceName=...
&DeviceListStr=[{"productKey":"...","deviceName":"..."},{"productKey
":"...","deviceName":"..."}]
&公共请求参数
```

返回示例

- JSON格式

```
{
  "RequestId":"419A3FC1-B517-4958-9414-5546765FA51F",
  "Success": true,
  "Data":{
  "MessageId": "abcabc123"
  }
}
```

- XML格式

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

```
    <Data>
 <MessageId>abcabc123</MessageId>
    </Data>
</NotifyAddThingTopoResponse>
```

2.8.26 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	要删除的设备所隶属的产品ID。 说明： 如果传入该参数，需同时传入**DeviceName**。
DeviceName	String	否	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	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。
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>
```

2.8.27 QueryDeviceStatistics

调用该接口查询指定产品下的设备统计数据。

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	是	要查询的设备所隶属的产品ID，IoT平台会返回目标产品下的设备统计数据。

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。
Data	Data	调用成功时，返回的设备统计信息。详情参见*StatisticsInfo*。

表 2-39: 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=al**********
&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>
```

2.9 Topic管理

2.9.1 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ID。
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。

名称	类型	是否必需	描述
Desc	String	否	Topic类的描述信息。
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。
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
```

```
}
```

2.9.2 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。
TopicId	Long	是	要修改的Topic类的 ID。
TopicShortName	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。
公共请求参数	-	是	请参见公共参数。

返回示例

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。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=UpdateProductTopic
&ProductKey=al*********
&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
}
```

2.9.3 QueryProductTopic

调用该接口查询指定产品的Topic类。

请求示例

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。
Data	ProductTopicInfo	调用成功时，返回的Topic类信息列表。详情参见*ProductTopicInfo*。

表 2-40: ProductTopicInfo

名称	类型	描述
ProductKey	String	产品ID。
Id	String	Topic类的Topic ID。
TopicShortName	String	Topic类中除*productKey*和*deviceName*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
```

```
}
```

2.9.4 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。

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
}
```

2.9.5 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。

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。 说明： 不支持以 sys 开头的系统 Topic。
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。) 说明： 不支持以 sys 开头的系统 Topic。

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。
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=/CXi4***/device1/get
&DstTopic=[{/CXi4***/device2/get},{/CXi4***/device3/get}]
&公共请求参数
```

返回参数

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

```
}
```

2.9.6 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。
DstTopics	List	是	目标Topic列表，即从**`SrcTopic`**订阅消息的Topic列表。

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。
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=/CXi4***/device1/get
&DstTopics=[{/CXi4***/device2/get},{/CXi4***/device3/get}]
&公共请求参数
```

返回示例

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

```
}
```

2.9.7 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。

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。
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=QueryTopicReverseRou
teTable
&Topic=/CXi4***/devic21/get
&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"]
```

```
}
```

2.9.8 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。

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。
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=/CXi4***/device1/get
&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"]
```

```
}
```

2.10 消息通信

2.10.1 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ID。
TopicFullName	String	是	要接收消息的Topic全称。可以是用户Topic类和您自定义的Topic类（不支持系统Topic类）。 • 用户Topic类结构：`/productKey/deviceName/update` • 自定义Topic类结构：`productKey/deviceName/topicShortName` 其中，*topicShortName*是您创建Topic时自定义的类目。 您可以调用**QueryProductTopic**接口查询产品下所有的Topic列表。
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：** 一条消息在物联网平台中最多可以保存7天。

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。
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=al*********
&TopicFullName=%2FproductKey%2FdeviceName%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>
```

2.10.2 PubBroadcast

调用该接口向指定产品对应的所有Topic发布广播消息。

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：PubBroadcast。
ProductKey	String	是	要发送广播消息的产品Key。
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。 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5个广播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。

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。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?&Action=PubBroadcast
&ProductKey=al**********
&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>
```

2.10.3 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ID。
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。
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=al*******
&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,
      "RrpcCode":"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>
    <RrpcCode>SUCCESS</RrpcCode>
    <PayloadBase64Byte>d29ybGQgaGVsbG8=</PayloadBase64Byte>
    <MessageId>889455942124347392</MessageId>
</RRpcResponse>
```

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。
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`*是该设备的自定义类目。 您可以调用**QueryDevice**接口，查看产品下的所有设备。返回结果中包含所有的**DeviceName**。 您可以调用**QueryProductTopic**接口，查看产品下的所有Topic类。返回结果中包含所有的**TopicShortName**。
Select	String	否	要执行的SQL Select语句。具体内容参照SQL表达式。 **说明：**

名称	类型	是否必需	描述
			此处传入的是**Select**下的内容。例如，如果**Selcet**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。
DataType	String	否	规则使用的数据格式，取值： `JSON`：JSON数据。 `BINARY`：二进制数据。 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`。
topicType	Integer	是	• 0表示高级版产品的上行Topic。 高级版产品的上行Topic有： • `/thing/event/property/post` • `/thing/event/${tsl.event.identifier}/post`，${}部分由该产品物模型决定。 • `/thing/list/found`，仅支持网关类型的产品。 • `/thing/downlink/reply/message` 示例：`/sys/${productKey}/${deviceName}/thing/event/property/post`，其中`${deviceName}`可以使用通配符`+`。 • 1表示用户自定义Topic。

名称	类型	是否必需	描述
			示例：/${productKey}/${deviceName}/get，其中${deviceName}可以使用通配符+。

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。
RuleId	Long	调用成功时，规则引擎为该规则生成的规则ID，作为其标识符。 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=al*********
&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

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

```
}
```

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数据转发至支持的其他阿里云服务。

限制说明

规则引擎支持的地域及目标云产品，请参见地域与可用区。

在调用**CreateRule**接口创建规则后，您可以调用本接口，在规则下创建具体的规则动作，定义将Topic数据转发至支持的其他阿里云服务。

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，定义将数据转发至消息服务、函数计算和表格存储。

说明：

如果您想将数据转发至云数据库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： 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IoT 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。

REPUBLISH类型Configuration定义

表 2-41: REPUBLISH Configuration

名称	描述
topic	转发至的目标Topic（系统Topic或自定义Topic）。 高级版设备的下行系统Topic如下： • `/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Topic。 • `1`表示用户自定义Topic。

REPUBLISH 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}
```

OTS类型Configuration定义

表 2-42: 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IoT指定的系统服务角色，您可以授权IoT访问您的表格存储。授权角色信息如下： `{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingotsrole", "roleName": "AliyunIOTAccessingOTSRole" }` 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IoT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 **2-43: 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 **2-44: 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IoT访问您的消息服务。授权角色信息如下： `{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingmnsrole", "roleName": "AliyunIOTAccessingMNSRole" } }` 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IoT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 **2-45: 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IoT访问您的函数计算服务。授权角色信息如下： `{ "roleArn": "acs:ram::6541***:role/aliyuniotaccessingfcrole", "roleName": "AliyunIOTAccessingFCRole" }` 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IoT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"
}
```

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。
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=REPUBLISH
&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。

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。

示例

请求参数

```
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。

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。

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。

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。
RuleInfo	RuleInfo	调用成功时，返回的规则详细信息。详情参见*RuleInfo*。

表 **2-46: RuleInfo**

名称	类型	描述
CreateUserId	Long	创建该规则的用户ID。
Created	String	该规则的创建时间。
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的时间。
Name	String	规则名称。
ProductKey	String	应用该规则的产品Key。

名称	类型	描述
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：`${deviceName}/topicShortName`。其中，*`${deviceName}`*指具体设备的名称，*`topicShortName`*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：`${productKey}/${deviceName}/ topicShortName`。
Where	String	该规则SQL语句中的**Where**查询条件。
topicType	Integer	若用户设置过规则SQL语句，则返回： • 0表示高级版产品的上行Topic。 • 1表示用户自定义Topic。 若用户未设置过规则SQL语句，则返回-1。

示例

请求示例

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

返回示例

```
{
    "RequestId": "A8F48485-44B9-40D8-A56D-F716F384F387",
    "RuleInfo": {
        "CreateUserId": 32164***,
        "Created": "Sun Apr 08 13:42:50 CST 2018",
        "DataType": "JSON",
        "Id": 1000,
        "Modified": "Sun Apr 08 13:42:50 CST 2018",
        "Name": "rule_test_pop123",
        "ProductKey": "a1GMr***",
        "RuleDesc": "创建规则 测试",
        "Select": "a,b,c",
        "ShortTopic": "test_topic/get",
        "Status": "STOP",
        "Topic": "/a1GMr***/test_topic/get",
```

```
            "Where": "a>1"
            "topicType": "1"
        },
        "success": true
}
```

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。

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。
RuleActionInfo	RuleActionInfo	调用成功时，返回的规则动作详细信息。详情参见*RuleActionInfo*。

表 2-47: 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，取值： REPUBLISH：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。 HiTSDB：存储到高性能时间序列数据库。 FC：发送数据到函数计算。 RDS：存储数据到云数据库中。

名称	类型	描述
Configuration	String	该规则动作的配置信息。

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

```
{
    "RequestId": "8FC9E36B-E0DC-4802-84EE-184E255B4E95",
    "RuleActionInfo": {
        "Configuration": "{\"endPoint\":\"http://ons.aliyun.com:8080
/rocketmq/nsaddr4client\",\"regionName\":\"cn-shanghai\",\"topic\":\"
test_hhf\",\"uid\":\"3216***\"}",
        "Id": 10001,
        "RuleId": 1000,
        "Type": "ONS"
    },
    "Success": true
}
```

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	从返回结果中的第几页开始显示。最大取值1000，默认值 1。

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。
Data	RuleInfo	调用成功时，返回的规则信息列表。详情参见*RuleInfo*。
PageSize	Integer	每页显示的记录数。
Page	Integer	当前页码。
Total	Integer	总页数。

表 **2-48: RuleInfo**

名称	类型	描述
CreateUserId	Long	创建该规则的用户ID。
Created	String	该规则的创建时间。
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的时间。
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：`${deviceName}/ topicShortName`。其中，*`${deviceName}`*指具体设备的名称，*`topicShortName`*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：`${productKey}/${deviceName}/ topicShortName`。
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=10
&CurrentPage=1
&公共请求参数
```

返回示例

```
{
    "Data": [{
        "CreateUserId": 32164***,
        "Created": "Sun Apr 08 13:42:50 CST 2018",
        "DataType": "JSON",
        "Id": 1000,
        "Modified": "Sun Apr 08 13:42:50 CST 2018",
        "Name": "rule_test_pop123",
        "ProductKey": "HMyBnl***",
        "RuleDesc": "创建规则测试",
        "Select": "a,b,c",
        "ShortTopic": "test_topic/get",
        "Status": "STOP",
        "Topic": "/HMyBnl***/test_topic/get",
        "Where": "a>1"
    },
    {
        "CreateUserId": 32164***,
        "Created": "Sun Apr 08 11:29:05 CST 2018",
        "DataType": "BINARY",
        "Id": 1001,
        "Modified": "Sun Apr 08 11:33:05 CST 2018",
        "Name": "创建规则 测试",
        "ProductKey": "HMyBnl***",
        "Select": "*",
        "ShortTopic": "test_topic/get",
        "Status": "RUNNING",
        "Topic": "/HMyBnl***/test_topic/get"
    }],
    "Page": 1,
    "PageSize": 2,
    "RequestId": "43ED5445-90B1-4A97-95A1-8BF6F48CD6E9",
    "Success": true,
    "Total": 30
}
```

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。

名称	类型	是否必需	描述
RuleId	String	是	要查询的规则ID。

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。
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 **2-49: 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： REPUBLISH：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。 HiTSDB：存储到高性能时间序列数据库 FC：发送数据到函数计算。 RDS：存储数据到云数据库中。
Configuration	String	该规则动作的配置信息。

示例

请求参数

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

返回示例

```
{
```

```
    "RequestId": "21D327AF-A7DE-4E59-B5D1-ACAC8C024555",
    "RuleActionList": [{
        "Configuration": "{\"topic\":\"/a1GM***/bike/get\",\"uid\":\"
3216***\"}",
        "Id": 10001,
        "RuleId": 1000,
        "Type": REPUBLISH
    },
    {
        "Configuration": "{\"endPoint\":\"http://ons.aliyun.com:8080
/rocketmq/nsaddr4client\",\"regionName\":\"cn-shanghai\",\"topic\":\"
test_ons\",\"uid\":\"3216***\"}",
        "Id": 10002,
        "RuleId": 1000,
        "Type": ONS
    }],
    "Success": true
}
```

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。

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。

示例

请求参数

```
https://iot.cn-shanghai.aliyuncs.com/?Action=StartRule
&RuleId=1000
```

```
&公共请求参数
```

返回参数

```
{
    "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。

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。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=StopRule
&RuleId=10000
&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。
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*是该设备的自定义类目。 您可以调用**QueryDevice**接口，查看产品下的所有设备。返回结果中包含所有的**DeviceName**。 您可以调用**QueryProductTopic**接口，查看产品下的所有Topic类。返回结果中包含所有的**TopicShortName**。
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**Selcet**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。
Where	String	否	规则的触发条件。具体内容参照SQL表达式。 **说明：**

名称	类型	是否必需	描述
			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表示高级版产品的上行Topic。 高级版产品的上行Topic有： • /thing/event/property/post • /thing/event/${tsl.event.identifier}/post，${}部分由该产品物模型决定。 • /thing/list/found，仅支持网关类型的产品。 • /thing/downlink/reply/message 示例：/sys/${productKey}/${deviceName}/thing/event/property/post，其中${deviceName}可以使用通配符+。 • 1表示用户自定义Topic。 示例：/${productKey}/${deviceName}/get，其中${deviceName}可以使用通配符+。

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。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?Action=UpdateRule
&RuleId=1000
&Name=test_2
&ProductKey=al*********
&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。
Type	String	是	规则动作类型，取值： 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IoT Topic。

名称	类型	是否必需	描述
			说明： • 新加坡地域服务（接入endpoint为ap-southeast）不支持FC。 • 美国西部1地域服务（接入endpoint为us-west-1）不支持、FC、ONS。 • 二进制数据格式的规则（即规则的**DataType**参数是BINARY）不支持OTS。
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*中的各类型的Configuration描述。

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。

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 设备影子

2.12.1 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ID。
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。
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=al*********
&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>
```

2.12.2 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ID。
DeviceName	String	是	要修改影子信息的设备名称。
ShadowMessage	String	是	修改后的设备影子信息。示例如下： `{ "method": "update", "state": { "desired": { "color": "green" } }, "version": 2 }` 详情参见 *ShadowMessage*。
公共请求参数	-	是	请参见公共参数。

表 2-50: 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。

示例

请求示例

```
https://iot.cn-shanghai.aliyuncs.com/?&Action=UpdateDeviceShadow
&ProductKey=al*********
&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>
```