

06 使用指南

06 使用指南

文档版本 01
发布日期 2022-02-16



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 概述（联通用户专用）	1
2 应用管理	3
2.1 创建应用（联通用户专用）	3
2.2 授权应用（联通用户专用）	8
2.3 数据统计（联通用户专用）	9
3 产品模型（联通用户专用）	10
4 设备注册鉴权（联通用户专用）	14
5 订阅推送（联通用户专用）	22
6 数据上报（联通用户专用）	25
7 命令下发（联通用户专用）	27
8 设备配置更新（联通用户专用）	37
9 设备影子（联通用户专用）	41
10 规则引擎（联通用户专用）	46
11 群组与标签（联通用户专用）	56
12 设备监控（联通用户专用）	60
13 远程诊断（联通用户专用）	68
14 固件升级（联通用户专用）	71
15 软件升级	83
15.1 升级指导（联通用户专用）	83
15.2 设备侧适配开发指导（联通用户专用）	95
15.3 PCP 协议介绍（联通用户专用）	112
16 网关与子设备（联通用户专用）	119

1 概述（联通用户专用）

非联通用户请查看[设备接入服务](#)。

物联网平台提供海量设备的接入和管理能力，配合华为云其他产品同时使用，帮助快速构筑物联网应用，简化海量设备管理复杂性，节省人工操作，提升管理效率。

功能	简介
应用管理	应用可以理解为在物联网平台中为用户的业务划分一个项目空间，当用户在开发中心、线下环境里完成应用服务器侧的开发、物联网平台侧的开发以及设备侧的开发后，就需要在控制台上创建应用，将开发完的应用服务器与真实设备接入到这个项目空间中，实现设备的数据采集和设备管理。
产品模型	又称Profile，用于定义一款接入设备所具备的属性（如颜色、大小、采集的数据、可识别的指令或者设备上报的事件等信息），然后通过厂家、设备类型和设备型号，唯一标识一款设备，便于平台识别。产品模型可通过开发中心进行无码化开发。
设备注册鉴权	物联网平台对接入平台的设备进行鉴权认证。待真实设备上电后，设备可以上报数据到物联网平台，物联网平台根据应用服务器的订阅消息类型，把消息推送给应用服务器。
订阅推送	订阅：是指应用服务器通过调用物联网平台的API接口，向平台获取发生变更的设备业务信息（如设备注册、设备数据上报、设备状态等）和管理信息（软固件升级状态和升级结果）。 推送：是指订阅成功后，物联网平台根据应用服务器订阅的数据类型，将对应的变更信息推送给指定的URL地址。
数据上报	当设备完成和物联网平台对接后，一旦设备上电，设备基于在设备定义上的业务逻辑进行数据采集和上报，可以是基于周期或者事件触发。
命令下发	为能有效地对设备进行管理，设备的产品模型中定义了物联网平台可向设备下发的命令，应用服务器可以调用物联网平台开放的API接口向单个设备或批量设备下发命令，或者用户通过物联网平台直接向单个设备下发命令，配置或修改设备的服务属性值，以实现设备的远程控制。

功能	简介
设备配置更新	物联网平台提供设备配置更新功能，即用户可通过控制台对单个设备或批量设备的设备属性值进行修改，满足用户频繁、快捷、方便的管理设备的诉求。
设备影子	设备影子是一个JSON文件，用于存储设备的在线状态、设备最近一次上报的设备属性、应用服务器期望下发的配置。每个设备有且只有一个设备影子，设备可以获取和设置设备影子以此来同步状态，这个同步可以是影子同步给设备，也可以是设备同步给影子。
规则引擎	指用户可以在物联网平台上可以对接入平台的设备设定相应的规则，在条件满足所设定的规则后，平台会触发相应的动作来满足用户需求。包含设备联动和数据转发两种类型。
群组与标签	群组是一系列设备的集合，用户可以对应用下所有设备，根据区域、类型等不同规则进行分类建立群组，以便处理对海量设备的批量管理和操作。 物联网平台支持定义不同的标签，并对设备打标签，通过标签，可以快速筛选设备。
设备监控	提供查看设备详情、设备状态管理、查看报表、查看操作记录、查看审计日志、告警管理、设备消息跟踪等设备监控与运维能力，提升设备的可维护性。
远程诊断	支持用户对接入的设备进行远程维护操作，快速定位问题及恢复业务，减少近端维护引入的高成本。当前支持的远程维护操作包括设备的运行日志收集、重启模组。
固件升级	用户可以通过OTA的方式对支持LWM2M协议的设备进行固件升级，升级协议为LWM2M协议。
软件升级	用户可以通过OTA的方式支持对LWM2M协议的设备进行软件升级，升级协议为PCP协议。
网关与子设备	物联网平台支持设备直连，也支持设备挂载在网关上，作为网关的子设备，由网关直连，通过网关进行数据转发。

2 应用管理

2.1 创建应用（联通用户专用）

2.2 授权应用（联通用户专用）

2.3 数据统计（联通用户专用）

2.1 创建应用（联通用户专用）

非联通用户请查看[设备接入服务](#)。

应用可以理解为在物联网平台中为用户的业务划分一个项目空间，当用户在开发中心、线下环境里完成应用服务器侧的开发、物联网平台侧的开发以及设备侧的开发后，就需要在控制台上创建应用，将开发完的应用服务器与真实设备接入到这个项目空间中，实现设备的数据采集和设备管理。

遵循华为公有云的子项目概念，物联网平台不再以应用来区分商用场景，建议通过产品模型或者分组标签来区分商用场景（行业等）。默认一个用户只有一个应用。

创建应用

步骤1 使用华为云帐号，登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”（也可以使用华为云帐号，直接登录<https://cn-north-4-iot-sp.huaweicloud.com>）。

为保证良好的显示效果和易用性体验，请使用兼容性良好的浏览器，对于浏览器的要求如下：

浏览器类型	版本要求	分辨率
Internet Explorer	Internet Explorer 11及以上版本	推荐1366 x 768分辨率
Firefox	Firefox 51.0~61.0版本	
Google Chrome	Google Chrome 58.0~67.0版本	

步骤2 用户登录设备管理服务的控制台时，系统自动为用户创建了一个默认应用。由于默认应用的密钥无法获取，因此如果用户需要使用默认应用，则需要在应用定义里**重置应用密钥**。


如果用户需要自己重新创建应用，需要删除默认应用。

步骤3 (若使用默认应用则无需执行此步骤) 参考下表按照实际情况填写配置参数。

参数名称	参数说明
基本信息	
应用名称	定义用户的应用的名称，应用名称必须为帐号下唯一，且创建后不可更改。
所属行业	根据用户的应用的行业属性进行选择。
消息跟踪数据授权	设置物联网平台运营管理员可以跟踪发生故障的设备的权限。 <ul style="list-style-type: none">打开授权，表示平台管理员在辅助用户进行设备的故障定位时，可以跟踪设备上报的业务数据，便于快速解决问题。授权打开的状态下需要设置“授权时效”，可设置“指定时间”或者“永久有效”。为了保证用户的数据权益，物联网平台运维管理员跟踪的设备数据保留时间不超过3天。关闭授权，表示平台管理员在辅助用户进行设备的故障定位时，不能跟踪设备上报的业务数据，可能导致没有足够的信息，将会降低问题定位效率，建议您授权给平台管理员进行业务数据的跟踪。
消息推送	

参数名称	参数说明
选择协议	推送协议 应用服务器向物联网平台进行消息订阅，物联网平台在推送数据时可以采用加密的HTTPS协议或者非加密的HTTP协议。推荐采用HTTPS协议。 <ul style="list-style-type: none">• HTTPS方式：表示物联网平台与应用服务器之间采用加密的传输协议，需要应用服务器侧上传CA证书。• HTTP方式：表示物联网平台与应用服务器之间采用非加密的传输协议。此方式的安全性较低，存在物联网平台与应用服务器之间通信信息泄露风险。 CA证书 创建应用时无需上传CA证书，请创建完应用后根据 加载推送证书 上传CA证书。
平台能力	
设备数据处理	物联网平台提供设备 上报数据 的存储能力，用户可以通过“存储历史数据”的开关进行控制，默认为“打开”状态。 <ul style="list-style-type: none">• 打开开关：即物联网平台会对上报数据进行存储，存储时间以界面显示的存储时间为准。• 关闭开关：即物联网平台不对上报数据进行存储。 用户可以通过 数据转发规则 转发到华为云其他云服务上进行存储和处理。
推送服务	应用服务器向物联网平台订阅设备信息，物联网平台能够向应用服务器进行消息推送。
其他	
应用描述	对该应用的描述。
应用图标	为该应用添加自定义图标。

步骤4 （若使用默认应用则无需执行此步骤）勾选“我已阅读并同意《个人数据使用条款》”，单击“确定”，完成创建应用。创建完成后，系统弹出“成功”对话框，显示应用的基本信息，包含应用ID、应用密钥。

- 请单击“保存密钥至本地”，以保存应用密钥信息，密钥信息在应用详情页内不可见，请妥善保管。如果遗忘应用密钥时，可在“应用列表”中单击，选择“重置密钥”，或者通过应用详情页内“应用定义 > 安全”，单击“重置”进行重置密钥。

说明

应用ID和应用密钥用于应用服务器接入物联网平台，如果重置密钥，旧的密钥将不能使用，您的应用服务器需要更新为新的密钥才能重新接入平台，请谨慎操作。

- 单击“查看应用详情”进入应用详情页，具体功能介绍参照“查看应用”。
- 单击“返回应用列表”，返回到创建应用页。单击“应用列表”中应用的图标，可直接进入该应用的应用详情页。

---结束

查看应用

创建应用后，用户可根据实际情况查看或者修改应用信息。在“应用列表”中单击应用，可查看该应用的详细信息。



参数类型	参数说明
基本	可查询应用的基本信息，包括应用名称、应用ID、创建时间以及该应用所属的行业，应用ID用于应用服务器接入物联网平台。其中用户可根据实际变更修改应用所属的行业。
安全	<ul style="list-style-type: none"> 可重置密钥，重置后旧密钥自动失效，应用服务器需要更新为新的密钥才能重新接入物联网平台。 可打开或者关闭消息跟踪数据授权开关。 <ul style="list-style-type: none"> 打开授权，表示平台管理员在辅助用户进行设备的故障定位时，可以跟踪设备上报的业务数据，便于快速解决问题。授权打开的状态下需要设置“授权时效”，可设置“指定时间”或者“永久有效”。为了保证用户的数据权益，物联网平台运维管理员跟踪的设备数据保留时间不超过3天。 关闭授权，表示平台管理员在辅助用户进行设备的故障定位时，不能跟踪设备上报的业务数据，可能导致没有足够的信息，将会降低问题定位效率，建议您授权给平台管理员进行业务数据的跟踪。
接入	<ul style="list-style-type: none"> 可查询绑定在该应用下的设备与物联网平台进行数据对接的不同协议所对应的端口信息。 可查询绑定在该应用上的应用服务器与物联网平台进行数据对接的端口信息。 <p>应用和设备接入的域名和接口信息，请在设备接入服务的控制台上获取。</p>

参数类型	参数说明
消息推送	推送协议是由应用服务器在订阅物联网平台的设备信息时设置的传输协议来确定的。应用服务器设置的订阅数据推送的传输通道为HTTP时，用户可以设置采用加密的HTTPS协议或者非加密的HTTP协议进行传输数据。如果采用HTTPS协议，需要上传CA证书，证书的上传可参考 上传CA证书 。 说明 如果采用HTTPS协议，应用服务器取消了订阅，则绑定的证书会自动解绑，重新订阅后，请重新上传CA证书。
其他	用户可以修改应用图标和应用描述。

CA证书加载方法：

步骤1 在应用详情页内“应用定义 > 消息推送”，单击“证书管理”按钮，在弹出的对话框中单击“添加”按钮。

CA证书	域名/IP与端口	lb昵称	是否检查cnName	CNNAME

步骤2 按照表格填写相关信息后，单击“确定”完成证书的加载。

参数名称	参数说明
CA证书	需要提前申请和购买CA证书文件，CA证书由应用服务器侧提供。 说明 调测时可自行 制作调测证书 ，商用时建议更换为商用证书，否则会带来安全风险。
域名/IP与端口	物联网平台推送消息到应用服务器的域名或IP地址与端口信息。填写为订阅接口（“ 订阅平台业务数据 ”和“ 订阅平台管理数据 ”）里的callbackurl中对应的域名或IP地址与端口信息。例如： api.huawei.com:9001或172.0.1.2:8080。
lb昵称	证书加载的LoadBalance对应的昵称，选择默认值：default。
是否检查CNNAME	是否对CA证书的CNNAME进行校验，确保要加载的证书与申请的证书是完全匹配的。建议打开该开关。
CNNAME	“是否检查CNNAME”开关打开时出现。CA证书中携带的证书名称，请向证书申请人员获取。
支持SNI	默认不勾选，如果接收推送消息的服务器存在多个服务器使用同一个“IP+端口”，则需要勾选支持SNI，“CNNAME”填写需要接收消息的服务器的域名地址，以便服务器根据指定的域名携带对应的设备证书。
使用设备证书	保持默认值：关闭

----结束

2.2 授权应用（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

物联网平台作为一个跨行业的通用平台，面向广大的企业用户，支持不同行业、不同类型的应用接入，对设备访问有严格权限管理，每个用户、应用只能对自己创建的资源有访问权限，默认不能访问其他用户、应用创建的资源。

访问授权

物联网平台支持在控制台上填写应用ID、授权权限，将设备的管理权限授权给其他应用，包括同一个用户下的不同应用之间的授权、两用户下（如用户A开通设备接入服务，用户B开通设备管理服务）的不同应用之间授权，且两应用之间可以相互授权。可以授予访问资源部分权限（如只有查询权限）或整体赋权如管理角色(如设备管理的增删改查功能)。

授权不会传递，应用A授权给应用B，应用B授权给应用C，不等同于应用C获取了应用A的授权。

例如：用户将B应用授权给A应用管理，进行授权操作时可选择授权查询权限或者编辑权限，如果是授权编辑权限，则A应用可以管理B应用下的设备，A应用具备和B应用一样的设备管理权限，但不允许将B应用下的设备挂到A应用下的群组中；如果是查询权限，则A应用仅可以查询B应用下的设备。

解除授权

可以收回资源访问权限，即解除授权。


操作步骤

以应用A授权给应用B为例，具体操作步骤如下：


- 步骤1** 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。
- 步骤2** 单击A应用，进入应用详情。
- 步骤3** 单击“授权应用”页签。
- 步骤4** 选择“授权应用列表”，在列表中可以查看或删除已经授权过的应用。
- 步骤5** 单击“授权操作”，在弹出的“授权操作”对话框选择B应用。如果是其他用户，则选择“其他租户的应用”，并且填写B应用ID、授权权限，其中授权权限为“编辑”时，才可以操作授权应用下的设备（编辑的对象是应用下的设备资源，即对设备管理节点下进行操作，无法编辑授权应用详情中的信息）。

📖 说明

应用ID的获取方式：1、创建新应用成功时，系统会自动返回应用ID。2、对于已经创建的应用，可以在应用详情的“应用定义”页签中查看应用ID。

- 步骤6** 单击“授权”，完成应用A授权给应用B的操作。
 - 在A应用详情的“授权应用 > 授权应用列表”中，可以看到被授权的B应用，单击，可以解除授权。



- 在B应用详情的“被授权应用”中可以查看A应用的基本信息。如果应用A和B分别属于不同用户，还可以登录B应用对应的控制台，在“应用管理 > 被授权应用”页签中查看到A应用的基本信息。
- 如果应用A和B分别属于不同用户，在A应用详情的“授权应用 > 授权租户列表”中可以看到B应用所属的用户名称。单击 ，可以解除租户授权，此时该租户下所有被授权的应用都将解除授权，并从“授权应用列表”中移除。

步骤7 被授权的应用可以通过控制台查看和操作授权应用下的所有设备。

----结束

2.3 数据统计（联通用户专用）

非联通用户请查看[设备接入服务](#)。

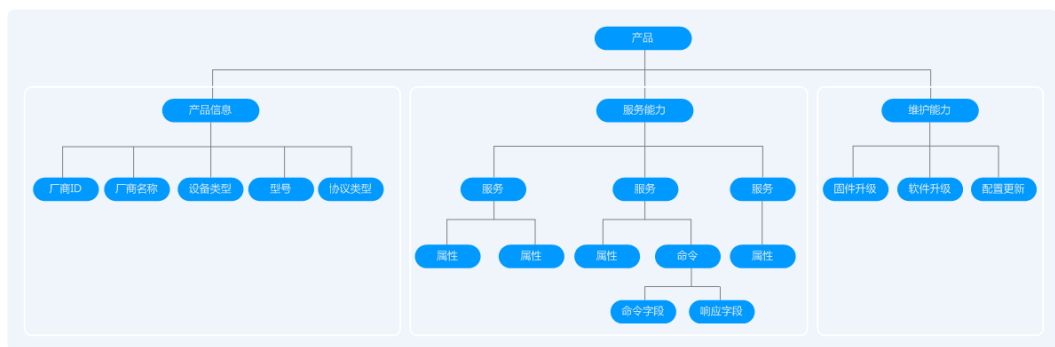
在控制台上，应用详情中单击“数据统计”，可查看该应用与物联网平台通信交互情况统计，单击每一行统计项右侧的“查看”，可以查看详细的报表统计信息。

统计名称	统计说明	数据刷新周期
数据上报统计	<p>统计最近72小时直连设备上业务数据的成功和失败次数（不含设备状态变更的数据，如设备上线）。上报数据类型例如当前温度、电量等。</p> <ul style="list-style-type: none"> ● 统计指标为次数，包含“总数”、“成功数”、“失败数”和“成功率”。 ● 统计维度为时间。平台提供对统计指标和维度的筛选功能。 	30分钟
命令下发统计	<p>统计来自物联网平台或第三方应用向设备发出的控制命令。</p> <ul style="list-style-type: none"> ● 统计指标为次数，包含“总数”、“成功数”、“失败数”和“成功率”。 ● 统计维度为时间。平台提供对统计指标和维度的筛选功能。 	30分钟
消息流控统计	<p>平台对同一个应用下的所有设备在同一时间上报的消息量进行控制，以防止在终端设备异常或受到外部攻击时，设备在短时间内可能上报大量的消息，造成系统性能下降或瘫痪。记录最近24小时的数据。</p>	手动刷新

3 产品模型（联通用户专用）

概述

产品模型（也称Profile）用于描述设备具备的能力和特性。开发者通过定义Profile，在物联网平台构建一款设备的抽象模型，使平台理解该款设备支持的服务、属性、命令等信息，如颜色、开关等。当定义完一款产品模型后，在进行[设备注册](#)时，就可以选择已在控制台上导入的产品模型。



Profile包括产品信息、服务能力、维护能力三部分：

- **产品信息**

描述一款设备的基本信息，包括厂商ID、厂商名称、设备类型、型号、协议类型。其中厂商ID和型号唯一标识一款产品。

例如：水表的厂商名称为“HZYB”，厂商ID为“TestUtf8Manuld”，设备类型为“WaterMeter”，型号为“NBloTDevice”，协议类型为“CoAP”。

- **服务能力**

描述设备具备的业务能力。将设备业务能力拆分成若干个服务后，再定义每个服务具备的属性、命令以及命令的参数。

以水表为例，水表具有多种能力，如上报水流、告警、电量、连接等各种数据，并且能够接受服务器下发的各种命令。Profile文件在描述水表的能力时，可以将水表的能力划分五个服务，每个服务都需要定义各自的上报属性或命令。说明如下：

服务名	描述
基础 (WaterMeterBasic)	用于定义水表上报的水流量、水温、水压等参数，如果需要命令控制或修改这些参数，还需要定义命令的参数。
告警 (WaterMeterAlarm)	用于定义水表需要上报的各种告警场景的数据，必要的话需要定义命令。
电池 (Battery)	定义水表的电压、电流强度等数据。
传输规则 (DeliverySchedule)	定义水表的一些传输规则，必要的话需要定义命令。
连接 (Connectivity)	定义水表连接参数。

注：具体定义几个服务是非常灵活的，如上面的例子可以将告警服务拆分成水压告警服务和流量告警服务，也可以将告警服务合入到水表基础服务中。

- **维护能力**

描述设备具备的维护能力，包括固件升级、软件升级、配置更新。

关于Profile文件的规范、含义和样例等，可参考[如何离线开发？](#)

开发产品模型

Profile的开发手段有在线开发和离线开发两种，我们推荐使用开发中心在线开发。

- 在线开发：即在开发中心上，通过界面操作进行开发Profile，开发完成后可以下载，并发布到产品中心。[如何在线开发？](#)
- 离线开发：是指开发者通过了解Profile格式规范，在本地进行开发、打包。[如何离线开发？](#)

新增产品模型


Profile开发完成并测试验证后，可以将profile文件导入到控制台上使用，有三种导入方式：

- **产品中心导入**：在开发中心上完成profile的定义并发布到产品中心。
- **本地导入**：指线下完成profile的开发，通过上传profile包的方式导入。
注：通过本地导入的profile不含编解码插件，如果设备上报采用的是二进制码流，请使用产品中心导入的方式。
- **手动创建**：手动创建一个产品（支持CoAP/LWM2M、MQTT或LoRaWAN），不定义设备属性（不需要开发profile及编解码插件），仅透传数据，不解析数据（透传模式具体请参考[6 数据上报（联通用户专用）](#)）。如使用手动创建的产品模型，建议仅开通设备接入服务即可。

说明

- 当协议类型选择LoRaWAN时，无法注册为普通设备，仅能注册为LoRaServer的子设备。
- 导入产品模型后，就无法对已导入产品模型的产品信息和服务能力进行修改，例如修改协议类型、新增一个服务等。可以通过[设备配置更新](#)、[设备影子](#)对设备服务属性值进行修改，通过界面修改OM维护能力。

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角，选择“产品模型”，单击“新增产品模型”。

步骤3 产品模型可通过从产品中心导入和从本地导入两种方式。

- 从产品中心导入：
 - a. 选择“从产品中心导入”，进入到产品中心页面。
 - 导入自有产品：产品中心默认显示自有产品列表，可通过产品名称或者设备类型搜索产品。
 - 导入第三方公开产品：“产品范围”选择“第三方公开”，可通过产品名称、设备类型或者厂商名称搜索产品。
 - 导入第三方私有产品：单击页面右上角“第三方私有产品”，在弹出的窗口中输入产品验证码，该验证码由设备厂商提供。
 - b. 在搜索的产品模型列表中单击需要导入的产品名称。
- 从本地导入：
 - a. 选择“本地导入”进入到本地导入产品的页面。
 - b. 在弹出的窗口中输入产品名称，并上传资源文件。
 - c. 单击“确定”，等待导入完成。

步骤4 在“产品模型”页面查看导入结果。

- 导入失败：可在“失败原因”中查看导入失败的原因，用户可根据失败原因定位错误。
- 导入成功：可单击“详情”，查看产品模型详情。



步骤5 用户可以在产品列表中删除不再使用的产品，单击“删除”即可。

单击“详情”，用户可查看产品模型详细信息，包括查看该产品模型具备的服务列表，产品维护能力配置，以及自定义topic。

关于自定义topic，每个产品模型最多可以自定义10个，只有MQTT协议接入的设备支持自定义，参数信息配置如下：

- 设备操作权限：
 - 发布：设备侧消息上报时，可按配置中自定义的topic进行消息上报；数据流转时，设备消息中会携带topic参数标识该消息从哪个topic上报。
 - 订阅：设备侧消息下发时，可在消息内容中指定此消息下发使用哪个topic；消息发往设备时，可以根据指定的topic下发。
 - 发布和订阅：同时具备发布和订阅的权限。
- topic名称：自定义即可。支持英文字母、数字、下划线，+和#（仅权限是订阅时支持），长度限制64。
- 描述：关于该topic的描述。

说明

- 如果还未使用该产品模型注册设备，则可以删除该产品模型。
- 如果已使用该产品模型注册过设备，则无法删除该产品模型。

---结束

4 设备注册鉴权（联通用户专用）

非联通用户请查看[设备接入服务](#)。

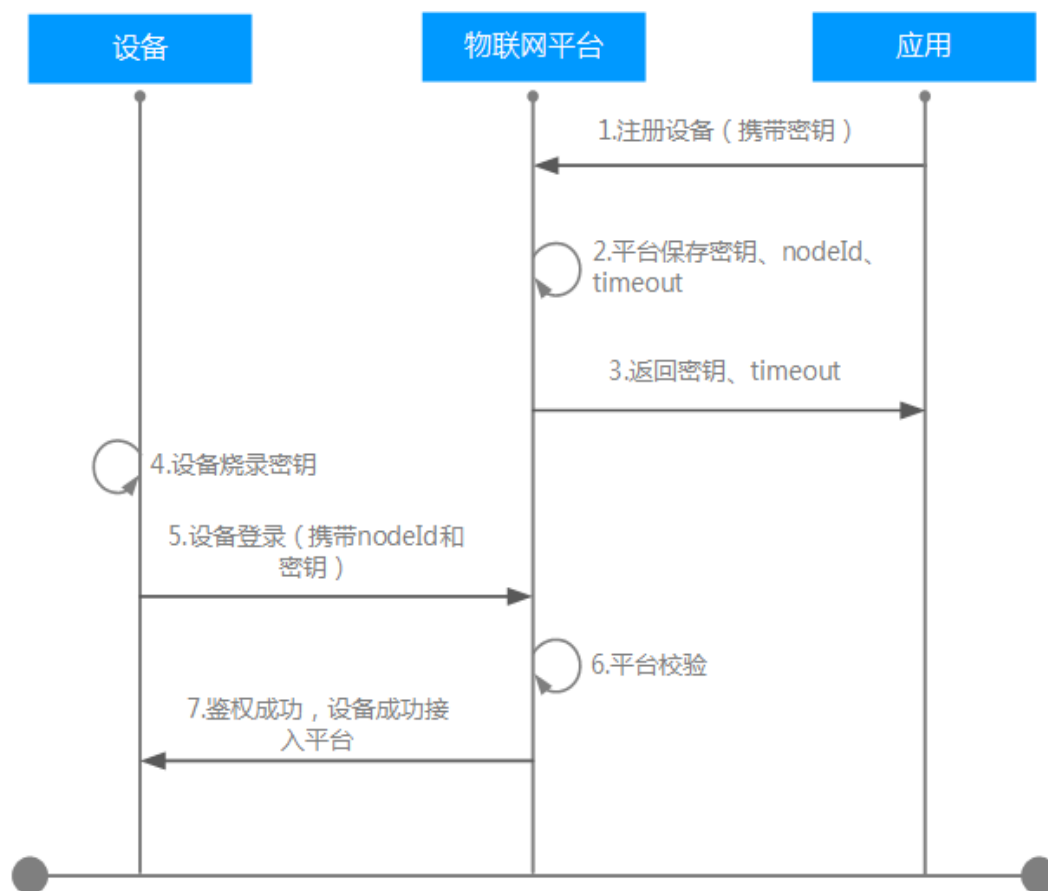
概述

设备注册鉴权是指物联网平台对接入平台的设备进行注册鉴权。对于不同接入方式的设备，注册鉴权方式不同。

接入类型	注册鉴权方式
使用LWM2M/CoAP协议接入的设备	<p>在设备接入物联网平台前，用户通过应用服务器调用注册设备（验证码方式）接口或通过控制台在物联网平台注册设备，并把设备的唯一标识码nodeId（如IMEI）设置为设备接入平台的验证码。在设备接入物联网平台时携带设备唯一标识，完成设备的接入鉴权。</p> <p>当采用DTLS/DTLS+传输层安全协议接入时，即设备为安全设备时，通过密钥进行设备和物联网平台之间的传输通道安全加密。如图所示。</p>
集成Agent Lite SDK的设备	<p>在设备接入物联网平台前，用户通过应用服务器调用注册设备（验证码方式）接口或通过控制台在物联网平台注册设备，并把设备的唯一标识码nodeId（如IMEI）设置为设备接入平台的验证码，平台返回设备ID。当设备上电时，携带nodeId等进行平台绑定，并返回设备ID和密钥。当绑定后设备重新登录时，携带设备ID和密钥完成网关的接入鉴权。如图所示。</p> <p>如果设备是网关，网关下有子设备，则当网关接入物联网平台后，在网关上通过调用Agent Lite的添加非直连设备接口，完成子设备的接入，具体请参见16 网关与子设备（联通用户专用）。</p>
使用原生MQTT协议接入的设备	<p>在设备接入物联网平台前，用户通过应用服务调用注册设备（密码方式）接口或通过控制台在物联网平台注册设备，获取设备ID和密钥，并把设备ID和密钥烧录到设备中。在设备接入物联网平台时携带设备ID和密钥、或者nodeId和密钥，通过一机一密的方式，完成设备的接入鉴权。如图所示。</p>

业务流程

使用LWM2M/CoAP协议接入的注册鉴权流程



1. 通过调用注册接口向物联网平台发送注册请求或者在控制台上注册设备。

2~3. 物联网平台向设备分配全局唯一的预置密钥，并返回timeout。

说明

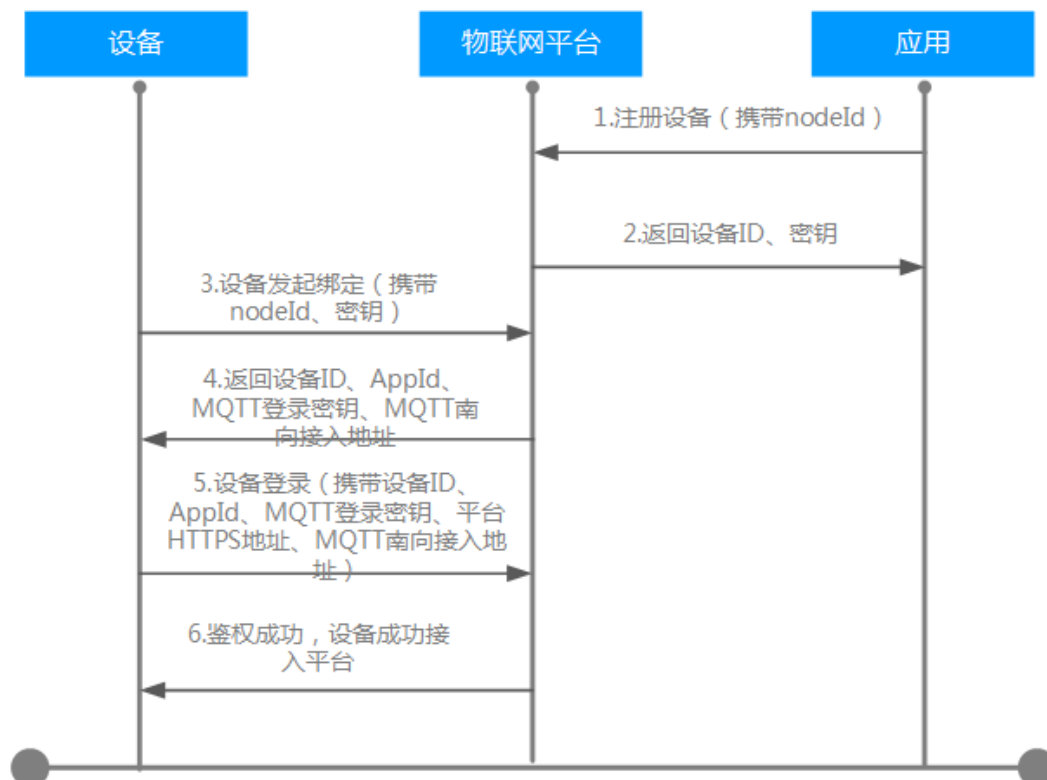
- 预置密钥可以在注册设备时自定义，如果没有定义，平台将自动分配预置密钥。
- timeout是指超时时间，若设备在有效时间未接入物联网平台，则平台会删除该设备的注册信息。

4. 用户将预置密钥烧录到设备硬件或软固件中。

5. 设备上电后，携带设备唯一标识码nodeId（如IMEI）和预置密钥发起接入鉴权请求。

6~7. 平台验证通过后，返回成功响应，设备连接物联网平台成功。

集成Agent Lite SDK的注册鉴权流程



1. 通过调用注册接口向物联网平台发送注册请求或者在控制台上注册设备。
2. 物联网平台向设备分配全局唯一的设备ID（deviceId），并返回密钥和timeout。

📖 说明

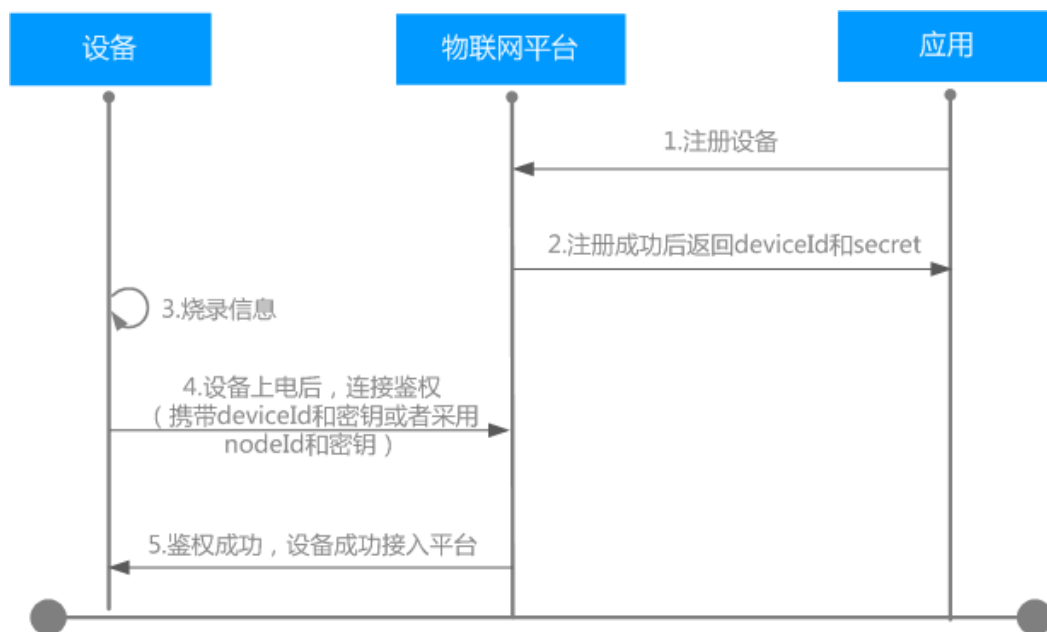
timeout是指超时时间，若设备在有效时间内未接入物联网平台，则平台会删除该设备的注册信息。

3~4. 设备第一次上电接入物联网平台时需要进行绑定操作，从而将设备与平台进行关联。设备携带nodeId和密钥进行平台绑定，平台返回设备ID、MQTT登录密钥、AppId、MQTT设备侧接入地址。

5~6. 绑定后设备重新登录时，携带设备ID、mqtt接入密钥、AppId、平台HTTPS地址、MQTT设备侧接入地址，完成设备的接入鉴权。具体请参见[Agent Lite SDK 使用指南（C）](#)、[Agent Lite SDK 使用指南（JAVA）](#)、[Agent Lite SDK 使用指南（Android）](#)。

如果设备是网关，网关下有子设备，则当网关接入物联网平台后，用户在网关上通过调用Agent Lite的“添加非直连设备”接口，完成子设备的接入。

使用原生MQTT协议接入的注册鉴权流程



1. 通过调用注册接口向物联网平台发送注册请求或者在控制台上注册设备。

📖 说明

注册时需要填写设备的nodeId，通常使用MAC地址，Serial No或IMEI作为nodeId。

2. 物联网平台向设备分配全局唯一的设备ID（deviceId）和密钥（secret）。

📖 说明


密钥可以在注册设备时自定义，如果没有定义，平台将自动分配密钥。

3. 用户将设备ID（deviceId）和密钥（secret）烧录到设备硬件或软固件中。
4. 设备上电后，携带设备ID（deviceId）和密钥（secret）或者携带nodeId和密钥（secret）发起接入鉴权请求。
5. 平台验证通过后，返回成功响应，设备连接物联网平台成功。

单个设备注册

物联网平台支持通过应用服务器调用“[注册设备（验证码方式）](#)” / “[注册设备（密码方式）](#)”接口、或者在控制台上进行单个设备注册。控制台上注册方式如下：

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角 ，选择“设备 > 设备注册”。

步骤3 选择页签“单个注册”，单击右上角“创建”，按照如下表格填写参数后，单击“确定”。

单设备注册

* 产品模型

* 设备识别码

设备ID

* 设备认证类型 密钥 X.509证书

密钥

确认密钥

参数名称	配置原则
选择产品	选择要注册的产品名称。 只有在“ 产品模型 ”里定义了产品，此处才可以选择具体的产品。如没有，请先上传或直接创建产品模型。
设备标识码 (nodeId)	设备唯一物理标识，如IMEI、MAC地址等，用于设备在接入物联网平台时携带该标识信息完成接入鉴权。 <ul style="list-style-type: none">原生MQTT设备：通过注册成功后生成的“设备ID”（与设备标识码一一对应）和“密钥”接入平台。NB-IoT设备、集成Agent Lite SDK的设备：设备通过注册时填写的“设备标识码”和“密钥”接入平台。
设备 ID(deviceId)	由平台生成的设备唯一标识，其生成规则为“产品ID” + “_” + “设备识别码”。 <ul style="list-style-type: none">不需要用户手动填写，由产品和设备识别码按照规则生成。对于原生MQTT设备，接入平台时携带设备ID和密钥接入鉴权。
设备认证类型	<ul style="list-style-type: none">密钥：设备通过密钥方式接入。

参数名称	配置原则
密钥	当“设备认证类型”选择“密钥”时填写。 <ul style="list-style-type: none">设备密钥，可自定义，不填写系统自动生成NB-IoT设备密钥只支持十六进制，用于设备和物联网平台之间的传输通道安全加密注册完后可重置密钥。
确认密钥	当“设备认证类型”为“密钥”时，再次填写密钥。

步骤4 （可选）注册设备成功，请保存好设备ID和密钥，用于设备接入平台认证。




----结束

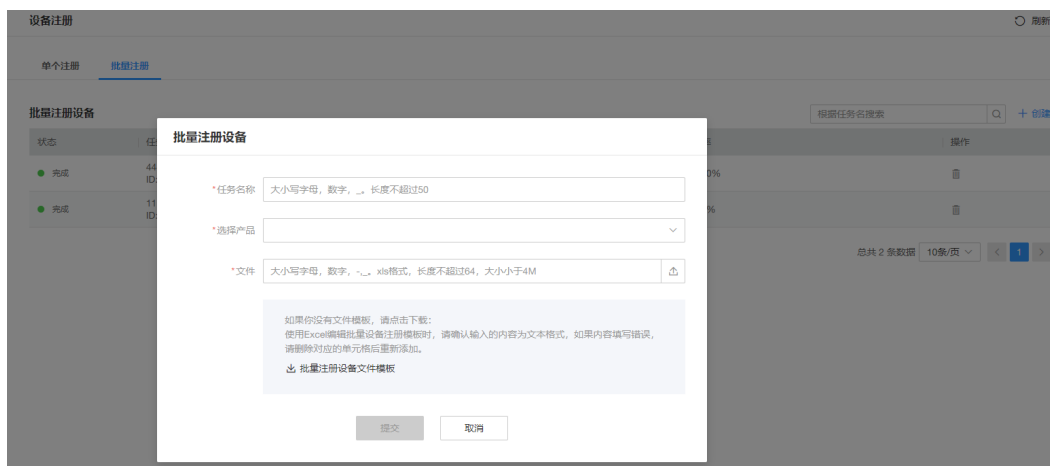
批量设备注册

物联网平台支持在控制台上进行批量设备注册。控制台上注册方式如下：

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角  ，选择“设备 > 设备注册”。

步骤3 选择页签“批量注册”，单击右上角“创建”，填写“任务名称”，“选择产品”，并上传批量注册文件，单击“提交”，完成设备的批量注册。



参数名称	配置原则
选择产品	选择要注册的产品名称。 只有在“ 产品模型 ”里定义了产品，此处才可以选择具体的产品。如没有，请先上传或直接创建产品模型。
nodeId(设备标识码)	设备唯一物理标识，如IMEI、MAC地址等，用于设备在接入物联网平台时携带该标识信息完成接入鉴权。 <ul style="list-style-type: none"> 原生MQTT设备：通过注册成功后生成的“设备ID”（与设备标识码一一对应）和“密钥”接入平台。 NB-IoT设备、集成Agent Lite SDK的设备：设备通过注册时填写的“设备识别码”和“密钥”接入平台。
secret(密钥)	<ul style="list-style-type: none"> 设备密钥，可自定义，不填写系统自动生成 NB-IoT设备密钥只支持十六进制，用于设备和物联网平台之间的传输通道安全加密 注册完后可重置密钥。

步骤4（可选）批量注册执行成功，如果是原生MQTT设备注册，请单击批量任务一行，进入任务的“执行详情”，点击“导出”按钮，保存好设备ID和密钥，用于原生MQTT设备接入平台。



----结束

重置密钥

登录控制台，进入具体的设备详情，选择“设置”页签。

- **重置密钥**：密钥用于设备采用原生MQTT、NB-IoT设备、集成Agent Lite SDK的设备接入接入物联网平台的鉴权认证。重置密钥后，需要将新的密钥信息更新到设备中，设备重新发起注册时，携带新的密钥进行认证。



相关 API 接口

- [注册设备（验证码方式）](#)
- [注册设备（密码方式）](#)
- [刷新设备密钥](#)

5 订阅推送（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

订阅：是指应用服务器通过调用物联网平台的“[订阅平台业务数据](#)”和“[订阅平台管理数据](#)”接口，向平台获取发生变更的设备业务信息（如设备注册、设备数据上报、设备状态等）和管理信息（软固件升级状态和升级结果）。

推送：是指订阅成功后，物联网平台根据应用服务器订阅的数据类型，将对应的变更信息（推送的通知内容可参考[推送通知](#)）推送给指定的URL地址，也称为回调地址（[什么是回调地址?](#)）。如果应用服务器没有订阅该类型的数据通知，即使数据发生了变更也不会进行推送。物联网平台进行数据推送时，数据格式为JSON格式，推送协议可以采用HTTP或HTTPS协议，其中HTTPS协议为加密传输协议，需要进行安全认证，更加安全，推荐使用。

订阅推送的示意图如下图所示：



应用服务器使用HTTPS协议调用API接口进行订阅时，应用服务器需要校验物联网平台的真实性，需要在应用服务器需要加载CA证书，该证书由物联网平台提供，下载[证书文件](#)并解压，在“Northbound API > code”中，根据具体的编程语言进行获取。

物联网平台采用HTTPS协议向应用服务器进行消息推送时，物联网平台需要校验应用服务器的真实性，需要在物联网平台上加载CA证书，该证书由应用服务器侧提供（调试时可自行[制作调测证书](#)，商用时建议更换为商用证书，否则会带来安全风险）。

推送机制：物联网平台向应用服务器推送消息后，如果应用服务器接收消息成功，会向物联网平台返回200 OK响应码。如果应用服务器无响应（响应超时时长15秒）或者应用服务器向物联网平台返回501、502、503、504响应码，表示消息推送失败，物联网平台会将消息进行缓存（缓存10分钟），然后通过轮询的方式向各个推送失败的应

用服务器进行推送重试，缓存时间超时后如果还未推送成功，物联网平台不再进行推送重试。在消息缓存时间内，物联网平台重试推送失败的次数达到10次后，会将该推送的URL地址设置为无效，并按照轮询的方式，轮询探测所有失败URL有效性，如果恢复，则会重新设置该URL为有效。您可以登录设备管理服务控制台的“应用管理 > 应用列表 > 应用详情 > 服务设置”中的“订阅推送服务（HTTP/HTTPS）”查看推送URL的状态。

📖 说明

只有在创建应用时打开“存储历史数据”开关的情况下（默认为打开），才能进行推送。如果平台不存储数据，直接透传的情况是不能进行消息推送的。

如何进行数据订阅

应用服务器接入到设备管理服务后，通过调用API接口或者集成SDK进行数据订阅。

- 通过API接口进行数据订阅请参考[API开发指导](#)。
- 通过集成SDK进行数据订阅请参考[SDK开发指南](#)。

📖 说明

上面的操作样例是基于开发中心的环境进行的开发调测，在商用时，需要将应用服务器对接平台的信息修改为设备管理服务的接入信息，并重新进行数据订阅。

- BASE_URL：更改为“iot-api.cn-north-4.myhuaweicloud.com（北京4）”。
- APPID和SECRET：更改为在设备管理服务控制台中创建应用时分配的“应用ID”和“应用密钥”。如果使用默认的应用，请通过“应用详情 > 应用定义”中重置“应用密钥”。

加载推送证书

应用服务器订阅消息成功后，设备的数据和状态发生变更时，物联网平台会向应用服务器进行消息推送。采用HTTPS协议进行推送时，请确保已经完成了消息订阅的操作后，再加载CA证书。

- 如果应用服务器取消了订阅后再重新订阅（URL不变），需要在物联网平台上重新上传CA证书。
- 如果应用服务器新增了订阅类型（新增URL），需要在物联网平台上加载与该URL对应的CA证书。即使新增URL使用的CA证书与原来推送的URL使用相同的证书，也需要重新上传CA证书。

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 选择“应用管理 > 应用列表”界面，在“我的应用”页面点击创建的应用。

步骤3 在“应用定义”页签的“消息推送”功能中，单击“证书管理”按钮，上传对应的证书文件。

步骤4 单击“添加”按钮，按照下表填写相关信息后，单击“确定”完成证书的加载。

参数名称	参数说明
CA证书	需要提前申请和购买CA证书文件，CA证书由应用服务器侧提供。 说明 调测时可自行 制作调测证书 ，商用时建议更换为商用证书，否则会带来安全风险。

参数名称	参数说明
域名/IP与端口	物联网平台推送消息到应用服务器的域名或IP地址与端口信息。填写为订阅接口（“ 订阅平台业务数据 ”和“ 订阅平台管理数据 ”）里的callbackurl中对应的域名或IP地址与端口信息。例如：api.huawei.com:9001或172.0.1.2:8080。
lb昵称	证书加载的LoadBalance对应的昵称，选择默认值：default。
是否检查CNNAME	是否对CA证书的CNNAME进行校验，确保要加载的证书与申请的证书是完全匹配的。建议打开该开关。
CNNAME	“是否检查CNNAME”开关打开时出现。CA证书中携带的证书名称，请向证书申请人员获取。
支持SNI	默认不勾选，如果接收推送消息的服务器存在多个服务器使用同一个“IP+端口”，则需要勾选支持SNI，“CNNAME”填写需要接收消息的服务器的域名地址，以便服务器根据指定的域名携带对应的设备证书。
使用设备证书	保持默认值：关闭

---结束

常见问题

订阅推送业务热点咨询问题如下，更多咨询问题请点击[查看更多](#)。

- [如何获取证书？](#)
- [调用订阅接口时，回调地址如何获取？](#)
- [回调地址可以使用域名吗？](#)
- [订阅后消息推送失败，例如提示503如何处理？](#)
- [为什么设备上报一条数据后应用服务器会收到多条推送？](#)

相关 API 接口

- [订阅平台业务数据](#)
- [订阅平台管理数据](#)
- [查询单个订阅](#)
- [批量查询订阅](#)
- [删除单个订阅](#)
- [批量删除订阅](#)

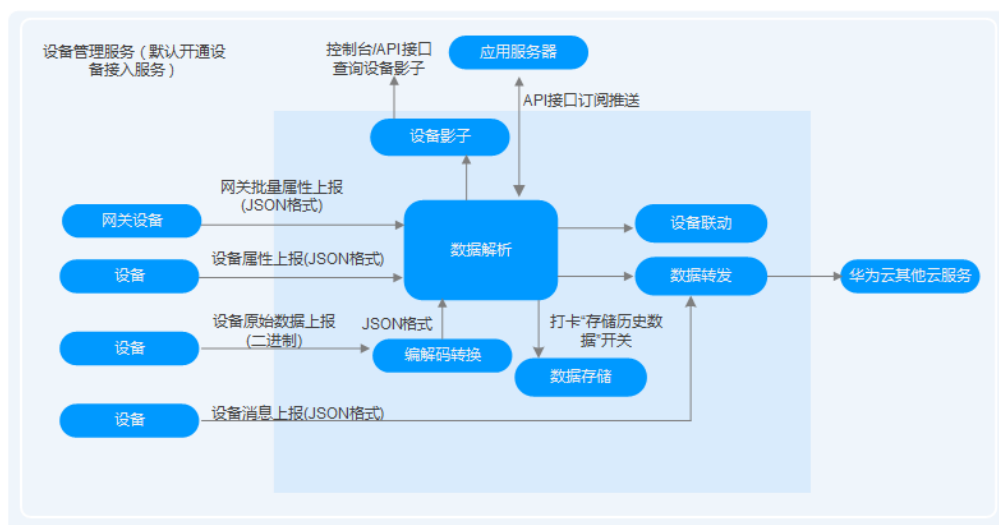
6 数据上报（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

当设备完成和物联网平台对接后，一旦设备上电，设备基于在设备上定义的业务逻辑进行数据采集和上报，可以是基于周期或者事件触发。设备可通过以下方式发送数据到物联网平台：

- 设备消息上报：设备可以通过消息上报接口将设备的自定义数据上报到平台，平台对设备上报的消息不进行解析和存储，通过[数据转发规则](#)转发到华为云其他云服务上进行存储和处理，然后通过其他云服务的控制台或者API接口进行进一步的数据处理。
- 设备原始数据（二进制）上报：设备可以通过二进制上报接口上报设备的原始码流，平台通过编解码插件将设备原始数据解析为产品模型定义的JSON格式，解析后的数据上报给设备管理服务进行相关业务处理。
- 设备属性上报：设备通过属性上报接口，将产品模型中定义的属性数据上报给平台，平台解析后的数据上报给设备管理服务进行相关业务处理。
- 网关批量属性上报：网关设备将批量设备的数据一次性上报到平台，平台解析后的数据上报给设备管理服务进行相关业务处理。



📖 说明

CoAP协议接入的设备考虑到功耗和带宽的影响，当前只支持设备上报原始的二进制码流，由平台根据编解码插件，将码流转换为产品模型中定义的JSON格式数据后，再进行后续处理。

查看历史数据

如果物联网平台在创建应用时开启了“[存储历史数据](#)”功能，则可以在控制台上查看设备上报的历史数据。

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 在设备列表中，单击具体的设备进入到设备的详情页面。

步骤3 在“历史数据”页签中，可以根据设备的服务、属性、时间段进行过滤历史数据，并提供对应的历史数据报表展示。



另外，界面上还支持查看全量的所有服务的历史数据，并支持导出历史数据。

---结束

相关 API 接口

- [查询设备历史数据](#)
- [查询设备服务能力](#)

7 命令下发（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

为能有效地对设备进行管理，设备的产品模型中定义了物联网平台可向设备下发的命令，应用服务器可以调用物联网平台开放的API接口向单个设备或批量设备下发命令，或者用户通过物联网平台直接向单个设备下发命令，以实现设备的远程控制。

物联网平台有两种命令下发机制，如下表所示。

命令下发机制	定义	适用场景	LWM2M/CoAP协议设备	集成Agent Lite SDK设备/原生MQTT设备
立即下发	不管设备是否在线，平台收到命令后立即下发给设备。如果设备不在线或者设备没收到指令则下发失败。支持给本应用的设备和被授予权限的其它应用的设备下发命令。	立即下发适合对命令实时性有要求的场景，比如路灯开关灯，燃气表开关阀。使用立即下发时，命令下发的时机需要由应用服务器来保证。	建议用户使用缓存下发模式，工作模式设置为PSM模式（系统默认值）。如需使用立即下发模式（工作模式设置为DRX或eDRX模式），需要在设备上上报数据后立即下发命令才能成功。	适用

命令下发机制	定义	适用场景	LWM2M/CoAP协议设备	集成Agent Lite SDK设备/原生MQTT设备
缓存下发	物联网平台在收到命令后先缓存，等设备上线或者设备上报数据时再下发给设备，如果单个设备存在多条缓存命令，则进行排队串行下发。支持给本应用的设备和被授予权限的其它应用的设备下发命令。	缓存下发适合对命令实时性要求不高的场景，比如配置水表的参数。	适用。工作模式设置为PSM模式（系统默认值）。	不适用

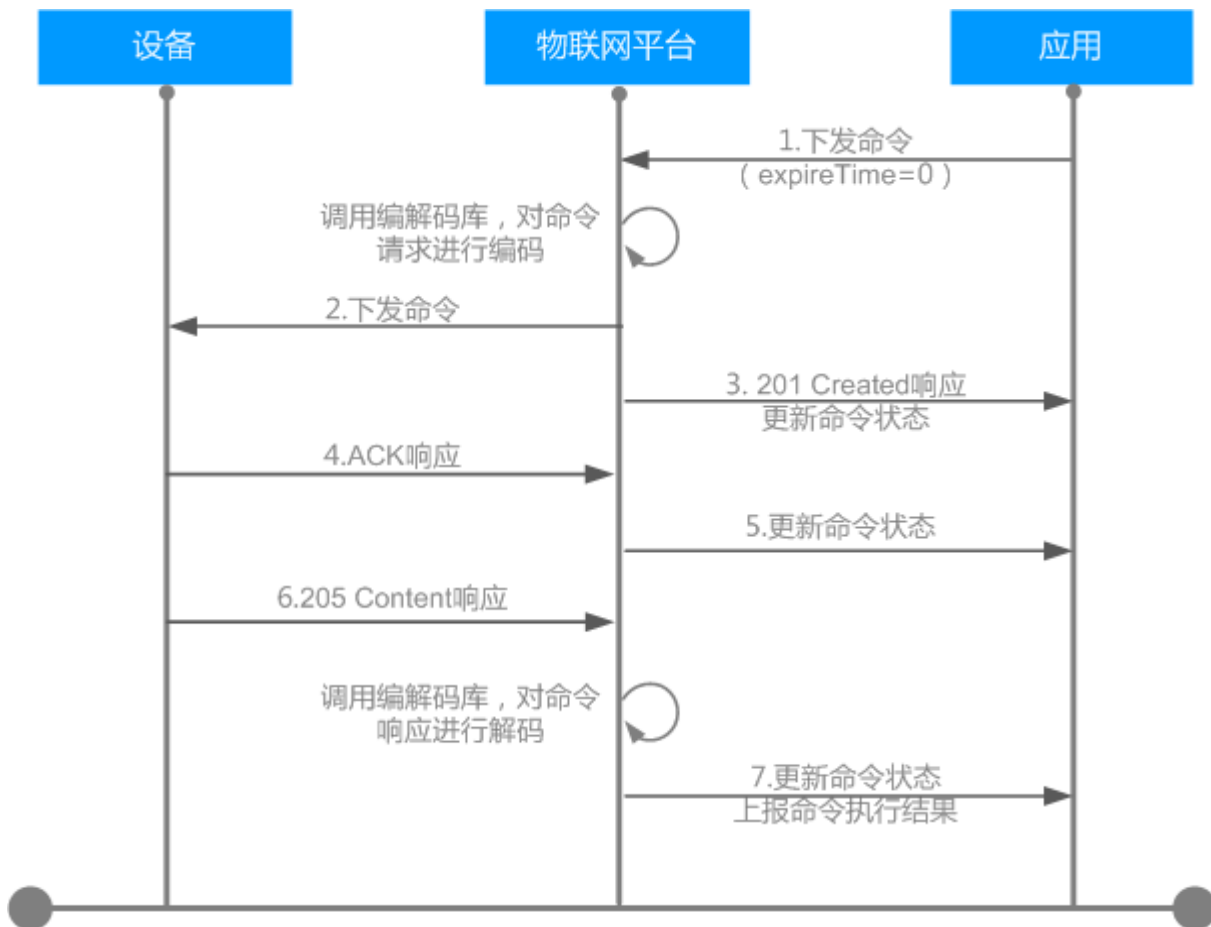
业务流程

LWM2M/CoAP协议设备场景

物联网平台提供立即下发和缓存下发两种机制。应用服务器向物联网平台下发命令时，携带参数expireTime（简称TTL，表示最大缓存时间）。如果不带expireTime，则默认expireTime为48小时。

- expireTime=0：命令立即下发。
- expireTime>0：命令缓存下发。

图 7-1 命令立即下发流程



1. 应用调用**创建设备命令**接口，下发请求到物联网平台，携带expireTime参数为0。消息样例如下：

```

Method: POST
request: https://server:port/iocm/app/cmd/v1.4.0/deviceCommands
Header:
"app_key: *****"
"Authorization:Bearer*****"
Content-Type:application/json
Body:
{
  "deviceId": "*****",
  "command": {
    "serviceId": "Brightness",
    "method": "SET_DEVICE_LEVEL",
    "paras": {
      "value": "1",
    }
  },
  "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
  "expireTime": 0,
  "maxRetransmit": 1
}
    
```

2. 物联网平台调用**编解码插件**对命令请求进行编码后，会通过LWM2M协议定义的设备管理和服务实现接口的**Execute**操作下发命令，消息体为二进制格式。
3. 物联网平台向应用返回201 Created，携带命令状态为**SENT**。（如果设备不在线或者设备没收到指令则下发失败，命令状态为**FAIL**）

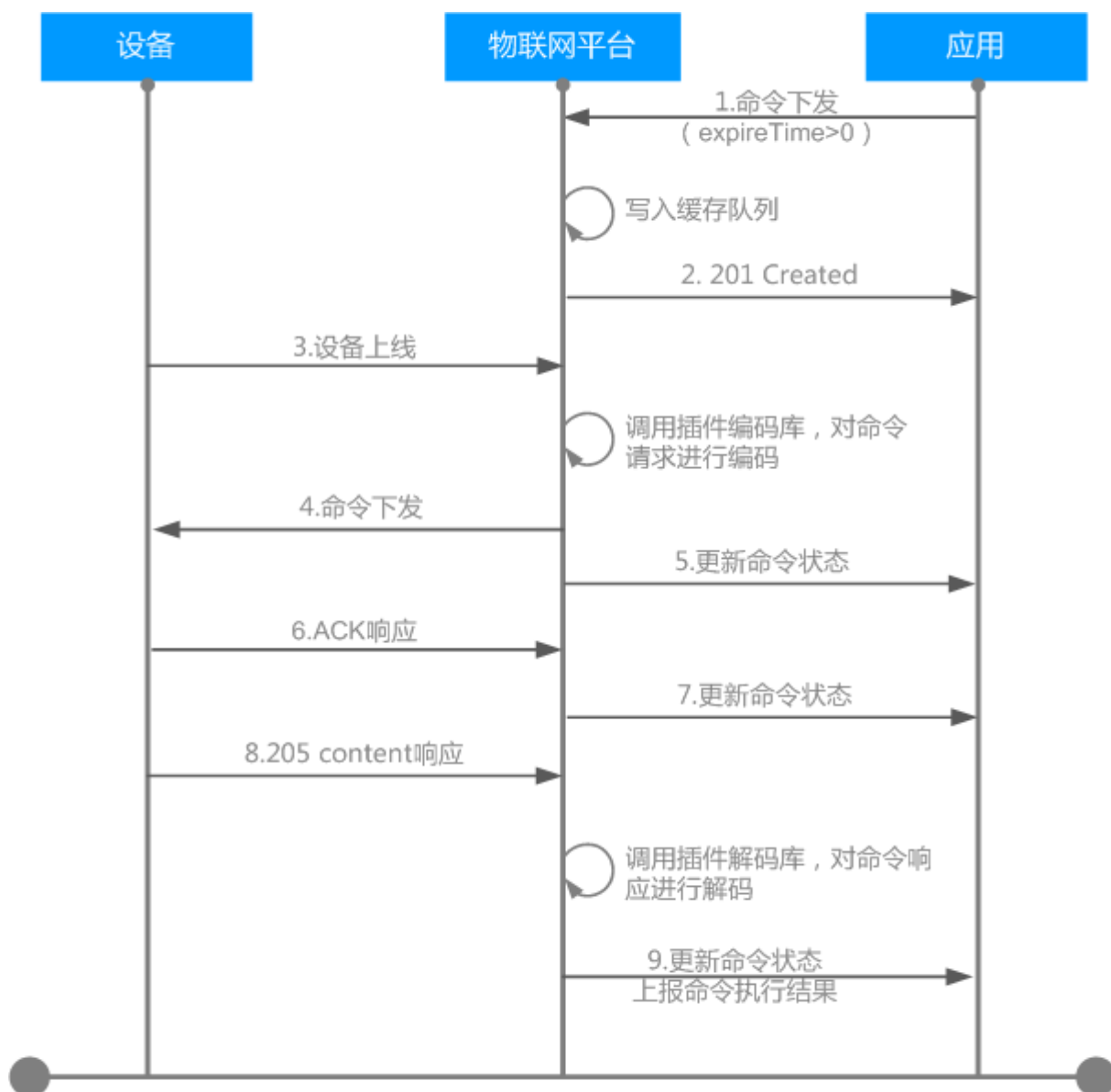
4. 设备收到命令后返回ACK响应。
5. 若应用下发命令时携带了"callbackUrl"，物联网平台通过**命令状态变化通知**接口推送消息给应用，携带命令状态为**DELIVERED**。消息样例如下：

```
Method: POST
request:
Body:
{
  "deviceId": "6543214c-72bb-4131-9e14-fad974123456",
  "commandId": "108a9c71462a48e09426e06e844d47ba3",
  "result": {
    "resultCode": "DELIVERED",
    "resultDetail": null
  }
}
```

6. 设备执行命令后通过205 Content响应返回命令执行结果。
7. 若应用下发命令时携带了"callbackUrl"，物联网平台会调用编解码插件对设备响应进行解码，然后通过**命令状态变化通知**接口推送消息给应用，携带命令状态为**SUCCESSFUL**。消息样例如下：

```
Method: POST
request:
Body:
{
  "deviceId": "6543214c-72bb-4131-9e14-fad974123456",
  "commandId": "108a9c71462a48e09426e06e844d47ba",
  "result": {
    "resultCode": "SUCCESSFUL",
    "resultDetail": null
  }
}
```

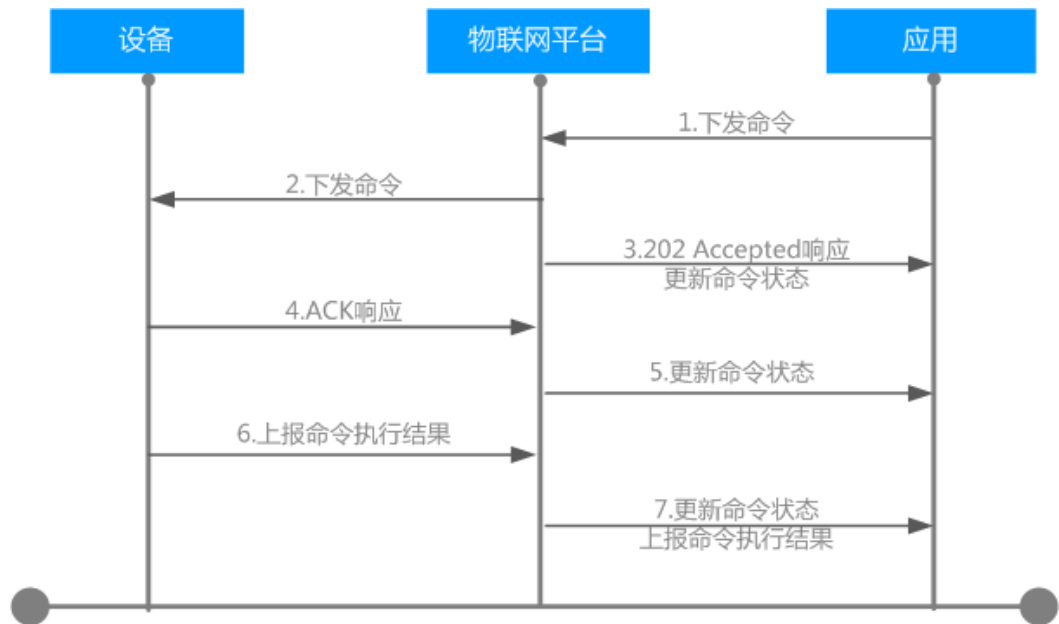
图 7-2 命令缓存下发流程



1. 应用调用**创建设备命令**接口，下发请求到物联网平台，携带expireTime参数大于0。
2. 物联网平台将命令写入缓存队列，并上报201 Created，携带命令状态为**PENDING**。
3. 设备上线或设备上报数据到平台。
4. 物联网平台调用**编解码插件**对命令请求进行编码后，根据协议规范下发命令给设备。
5. 若应用下发命令时携带了"callbackUrl"，物联网平台通过**命令状态变化通知**接口推送消息给应用，携带命令状态为**SENT**。
6. 后续流程请参考“命令立即下发”的步骤4到步骤7。

集成Agent Lite SDK设备/原生MQTT设备场景

对于集成Agent Lite SDK设备/原生MQTT设备，物联网平台只支持直接下发模式，设备不在线时命令下发失败。



1. 应用调用**设备服务调用**接口，下发请求到物联网平台。消息样例如下：

```
Method: POST
request: https://server:port/iocm/app/cmd/v1.4.0/deviceCommands
Header:
"app_key: *****"
"Authorization:Bearer*****"
Content-Type:application/json
Body:
{
  "deviceId": "*****",
  "command": {
    "servicId": "Brightness",
    "method": "SET_DEVICE_LEVEL",
    "paras": {
      "value": "1",
    }
  },
  "callbackUrl": "http://127.0.0.1:9999/cmd/callbackUrl",
  "expireTime": 0,
  "maxRetransmit": 1
}
```

2. 物联网平台根据协议规范下发命令给设备。
MQTT设备必须先订阅**平台命令下发**接口对应的Topic才能收到平台下发的命令，消息样例如下：

```
{
  "msgType": "cloudReq",
  "servicId": "Brightness",
  "mid": 2016,
  "cmd": "SET_DEVICE_LEVEL",
  "paras": {
    "value": "1"
  }
}
```

3. 物联网平台向应用返回202 Accepted，携带命令状态为**SENT**。
4. 设备收到命令后返回ACK响应。
5. 若应用下发命令时携带了"callbackUrl"，物联网平台通过**命令状态变化通知**接口推送消息给应用，携带命令状态为**DELIVERED**。消息样例如下：

```
Method: POST
request:
Body:
```

```
{
  "deviceId": "6543214c-72bb-4131-9e14-fad974123456",
  "commandId": "108a9c71462a48e09426e06e844d47ba",
  "result": {
    "resultCode": "DELIVERED",
    "resultDetail": null
  }
}
```

6. 设备执行命令后通过**设备命令响应通知**接口返回命令执行结果，消息样例如下：

```
response: Status
Code: 200 OK
```

7. 若应用下发命令时携带了"callbackUrl"，物联网平台会调用编解码插件对设备响应进行解码，然后通过**命令状态变化通知**接口推送消息给应用，携带命令状态为**SUCCESSFUL**。消息样例如下：

```
Method: POST
request:
Body:
{
  "deviceId": "6543214c-72bb-4131-9e14-fad974123456",
  "commandId": "108a9c71462a48e09426e06e844d47ba",
  "result": {
    "resultCode": "SUCCESSFUL",
    "resultDetail": null
  }
}
```

单个设备命令下发

LWM2M/CoAP协议设备

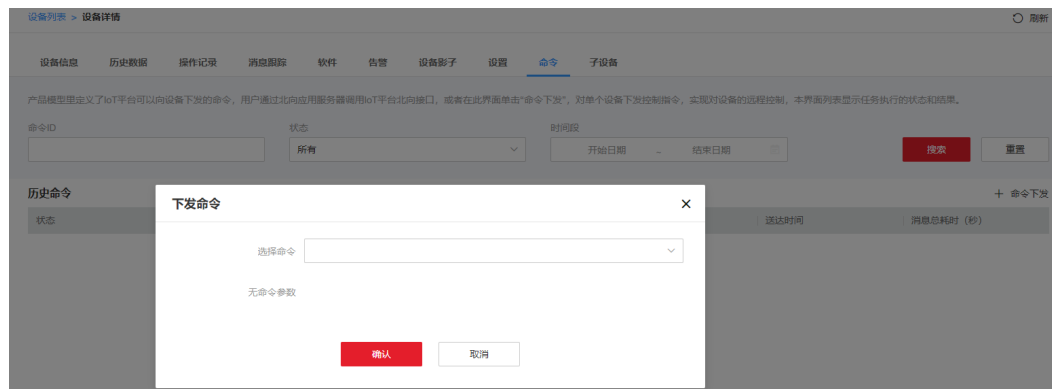
两种方式创建单个NB-IoT设备的命令下发：

- 调用**创建设备命令**接口。
- 在控制台上创建命令下发，操作步骤如下。

步骤1 登录**物联网平台控制台**，点击右上角“进入设备管理服务”。

步骤2 在设备列表中，单击具体的设备进入到设备的详情页面。

步骤3 在“命令”页签中，点击右上角“命令下发”，在弹出的窗口中选择需要下发的命令并设置命令参数。



📖 说明

- 仅NB-IoT设备支持通过控制台进行下发命令操作。
- 设备支持的命令，需要由**产品模型**定义后，才能在此进行命令下发操作。
- 下发命令的参数中，带“*”的参数为必选配置项，选择不同的命令，界面所呈现的参数不同。

步骤4 在界面上，展示通过界面或“**创建设备命令**”接口给设备下发的历史命令列表，通过该功能可以详细查看命令下发任务的创建时间、平台发送命令的时间、送达的时间、发送的状态等信息，便于用户了解命令的**命令执行状态说明**。

----结束

同时，还支持通过调用相关API接口查询、修改、撤销设备命令等操作，如下表所示。

API接口	功能
查询设备命令	应用服务器向设备下发命令后，可调用此接口在物联网平台查询下发命令的状态及内容信息，以了解命令的执行情况。
修改设备命令	应用服务器向设备下发命令后，如果由于命令排队、设备不在线等原因，物联网平台还未把命令下发给设备执行，此时应用服务器可调用此接口修改该命令的状态信息。当前仅支持把命令状态修改为CANCELED状态，即撤销命令的执行。
创建设备命令撤销任务	应用服务器向设备下发命令后，如果由于命令排队、设备不在线等原因，物联网平台还未把命令下发给设备执行（PENDING状态），此时应用服务器可调用此接口撤销指定设备的所有未下发的命令，对于已下发成功的命令不可撤销。
查询设备命令撤销任务	应用服务器创建了设备命令撤销任务后，可调用此接口查询设备命令撤销任务的详细信息和执行状态，以了解撤销任务的执行情况。

集成Agent Lite SDK设备/原生MQTT设备

通过调用**设备服务调用**接口，向设备下发命令。

支持通过控制台设备详情里的“命令”页签，查看给设备下发的历史命令列表，可以详细查看命令下发任务的创建时间、平台发送命令的时间、送达的时间、发送的状态等信息，便于用户了解命令的执行状态。

当前不支持通过API接口进行历史命令下发任务的查询。

批量设备命令下发

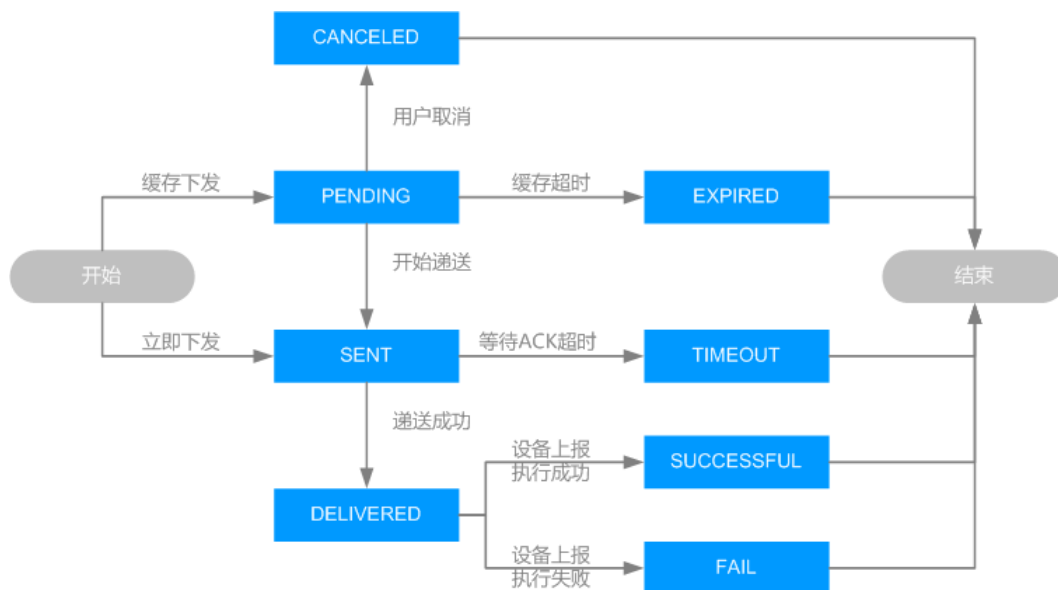
当前物联网平台仅支持对批量NB-IoT设备进行相同命令的下发。应用服务器可以调用**创建批量任务**接口，对批量设备下发控制指令。

批量任务下发之后，可以登录**物联网平台控制台**，点击右上角“进入设备管理服务”。在“批量任务”->“批量命令下发”界面查看任务执行的状态和结果，如果成功率低于100%，则可以单击具体的任务名称，进入任务详情，查看执行失败的原因。



命令执行状态说明

命令执行状态以及状态变化机制如下所示。



命令执行状态	说明
等待 (PENDING)	<ul style="list-style-type: none"> NB-IoT设备采用缓存下发模式下发命令时，如果设备未上报数据，物联网平台会将命令进行缓存，此时任务状态为“等待”状态。 NB-IoT设备采用立即下发模式下发命令时，无此状态。 MQTT设备下发命令时，无此状态。
超期 (EXPIRED)	<ul style="list-style-type: none"> NB-IoT设备采用缓存下发模式下发命令时，如果在设置的超期时间内，物联网平台未将命令下发给设备，则状态变更为“超期”。超期时间会根据应用侧接口中携带的expireTime为准，如果未携带，默认48h。 NB-IoT设备采用立即下发模式下发命令时，无此状态。 MQTT设备下发命令时，无此状态。
取消 (CANCELED)	如果命令下发任务为“等待”状态时，用户人工取消了命令下发任务，则任务状态变更为“取消”。

命令执行状态	说明
已发送 (SENT)	<ul style="list-style-type: none">• NB-IoT设备采用缓存下发模式下发命令时，设备上报数据，物联网平台会将缓存的命令发送给设备，此时状态会由“等待”变为“已发送”。• NB-IoT设备采用立即下发模式下发命令时，如果设备在线，状态为“已发送”。• MQTT设备下发命令时，如果设备在线，状态为“已发送”。
超时 (TIMEOUT)	NB-IoT设备收到命令后，物联网平台在180秒内未收到设备反馈的收到命令响应，此时状态会变为“超时”。MQTT设备无此状态。
已送达 (DELIVERED)	物联网平台收到设备反馈的已收到下发命令响应后，状态变为“已送达”。
成功 (SUCCESSFUL)	如果设备在执行完命令后，会给物联网平台反馈命令执行成功的结果，将任务状态变更为“成功”。
失败 (FAIL)	<ul style="list-style-type: none">• 如果设备在执行完命令后，会给物联网平台反馈命令执行失败的结果，将任务状态变更为“失败”。• NB-IoT设备采用立即下发模式下发命令时，如果设备离线，状态为“失败”。• MQTT设备下发命令时，如果设备离线，状态为“失败”。

相关 API 接口

- [创建设备命令](#)
- [设备服务调用](#)
- [查询设备服务能力](#)
- [查询设备命令](#)
- [修改设备命令](#)
- [创建设备命令撤销任务](#)
- [查询设备命令撤销任务](#)
- [创建批量任务](#)
- [查询指定批量任务信息](#)
- [查询批量任务的子任务信息](#)

8 设备配置更新（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

物联网平台提供设备配置更新功能，即用户可通过控制台对单个设备或批量设备的设备属性值进行修改，满足用户频繁、快捷、方便的管理设备的诉求。比如，某路灯最初设置温度属性达到A摄氏度时上报告警，用户需要修改该属性值为B摄氏度，即可修改其配置文件，然后通过控制台上传到物联网平台，以实现配置更新。

- 针对LWM2M协议设备的设备配置更新，物联网平台提供了[设备影子](#)功能，将修改设备的属性信息存储在设备影子中，待设备上线或上报数据时，将修改的设备属性值同步给设备，从而完成设备属性的修改。
- 针对原生MQTT协议接入或集成Agent Lite SDK的设备，物联网平台直接配置更新下发，修改设备的设备属性值。

说明


物联网平台第一次给设备下发配置更新，如果当前的配置文件还没有下发给设备前，又进行了第二次的配置下发，则将两次的配置文件进行合并后再下发（如果配置项重复，则后一次会覆盖前一次）。

前提条件

设备的[产品模型](#)中已定义该设备服务属性，且该属性支持可配置，在界面上传的配置文件的JSON格式，才能完成配置更新下发。

单个设备配置更新

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角 ，在“产品模型”界面上，单击设备所属产品模型的“详情”按钮，“维护能力配置”中的“设备配置”设置为“支持”。

步骤3 进入“所有设备”页面，单击设备所在行的 按钮，可对设备进行配置更新。

<input type="checkbox"/>	状态	设备名称	设备识别码	设备类型	厂商名称	操作
<input type="checkbox"/>	未激活	--	wm000002	WaterMeter	HZYB	
<input type="checkbox"/>	未激活	--	aefae0e0efae0efa	WaterMeter	HZYB	
<input type="checkbox"/>	未激活	--	wm000001	WaterMeter	HZYB	

步骤4 在弹出窗口中，按照下表配置参数。

参数名称	参数说明
任务名	创建设备配置下发的任务名称。
执行类型	任务的执行类型。分为“现在”、“设备在线时”、“自定义”。选择“自定义”时需要设置“执行时间。”
执行时间	执行类型选择“自定义”时生效，设置任务执行的时间。
重试策略	<ul style="list-style-type: none"> 否：不重试。 自定义：需要设置重试次数。
重试次数	当“重试策略”为“自定义”时需要设置此参数。
重试间隔（秒）	当“重试策略”为“自定义”时需要设置此参数。
配置文件	<p>需要导入设备的配置文件，可点击下载上一次下发成功的配置文件信息进行修改后再导入。</p> <p>如果在此操作前，未对设备进行过配置下发的操作，点击下载设备配置文件会提示“找不到资源”，这种情况下只能通过用户自定义的方式，手动制作配置文件，支持配置下发修改的项需要与设备支持的能力（设备对应的产品模型）保持一致，且配置文件需要为JSON格式，例如：</p> <pre>{ "sensitivity": "0", "dataReportInterval": "20" }</pre>

---结束

批量设备配置更新

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角 ，在“产品模型”界面上，单击设备所属产品模型的“详情”按钮，“维护能力配置”中的“设备配置”设置为“支持”。

步骤3 选择“批量任务>批量设备配置”，单击页面右上角的“创建”。

步骤4 在弹出的“批量设备配置”窗口中，按照下表配置参数。

参数名称	参数说明	配置原则
任务名称	批量设备配置的任务名称。	按照实际情况填写。长度不超过50的字符串。
是否使用策略	批量设备配置执行策略是否需要配置。	请根据实际情况选择，默认勾选。
执行类型	选择批量设备配置的任务执行时间。	<ul style="list-style-type: none"> ● 现在：任务创建完成后开始。 ● 设备在线时：当物联网平台与设备互联时执行任务。 ● 自定义：可以自定义选择开始和结束时间。“开始时间”和“结束时间”在“执行类型”参数设置为“自定义”时生效。
重试策略	任务失败后是否需要自动重新执行，默认为“否”。	<ul style="list-style-type: none"> ● 否：不重新执行。 ● 自定义：可自定义重试次数。
重试次数	“重试策略”参数设置为“自定义”时生效。	-
重试间隔（秒）	“重试策略”参数设置为“自定义”时生效。	-

步骤5 单击“下一步”，进入下一步配置。按照实际情况配置参数。

步骤6 选择需要批量配置下发的设备群组，单击“下一步”，跳转至文件上传页面。单击



，将整理好的Json格式设备配置信息文件上传。

配置文件制作方法：

- 之前有成功下发的配置文件：选择“设备 > 所有设备”，在右侧的窗口中单击相同类型设备所在行的 按钮，在“配置文件”一栏，下载上一次下发成功的配置文件文件进行修改后再导入。
- 之前未对设备进行过配置下发的操作：手工制作配置文件，支持配置下发修改的项需要与产品模型里支持的能力保持一致，且配置文件需要为JSON格式，例如：

```
{  "sensitivity": "0",  
  "dataReportInterval": "20"  
}
```

步骤7 单击“提交”，完成批量设备配置任务创建。

步骤8 界面列表显示任务执行的状态和结果。如果成功率低于100%，则可以单击具体的任务名称，进入任务详情，查看执行失败的原因。

----结束

相关 API 接口

[查询设备服务能力](#)

9 设备影子（联通用户专用）

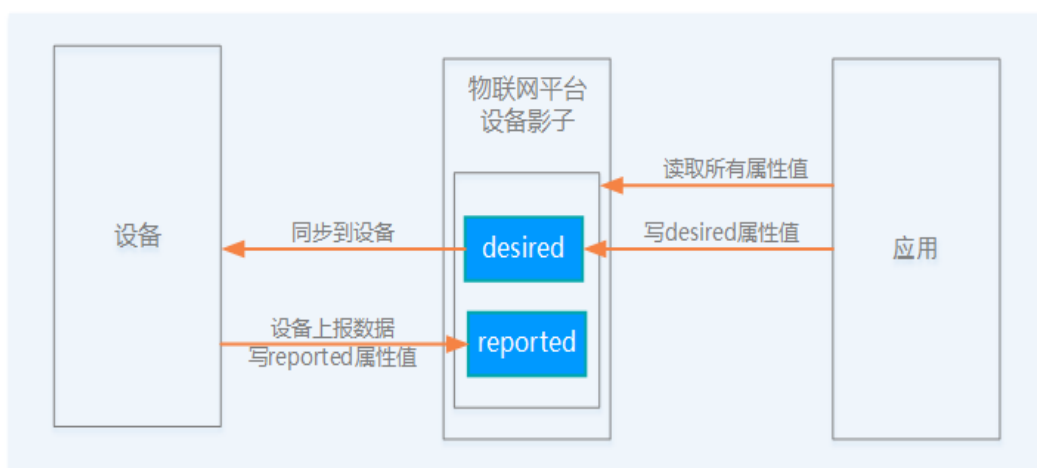
非联通用户请查看[设备接入服务](#)。

概述

物联网平台支持创建设备的“影子”。设备影子是一个JSON文件，用于存储设备的在线状态、设备最近一次上报的设备属性值、应用服务器期望下发的配置。每个设备有且只有一个设备影子，设备可以获取和设置设备影子以此来同步设备属性值，这个同步可以是影子同步给设备，也可以是设备同步给影子。

设备影子有desired区和report区。

- desired区用于存储对设备属性的配置，即期望值。当需要修改设备的服务属性值时，可修改设备影子的desired属性值，设备在线时，desired属性值立即同步到设备。如果设备不在线，待设备上线或上报数据时，desired属性值同步到设备。
- report区用于存储设备最新上报的设备属性值，即上报值。当设备上报数据时，平台刷新report属性值为设备上报的设备属性值。



说明

物联网平台仅支持对LWM2M协议的设备提供设备影子功能，对于希望通过设备影子修改的属性，在定义Profile文件时，需遵循LWM2M协议中定义的属性，且属性类型为“W”。LWM2M标准规范参见《[OMA-TS-LightweightM2M-V1_0_2-20180209-A.pdf](#)》。

应用场景

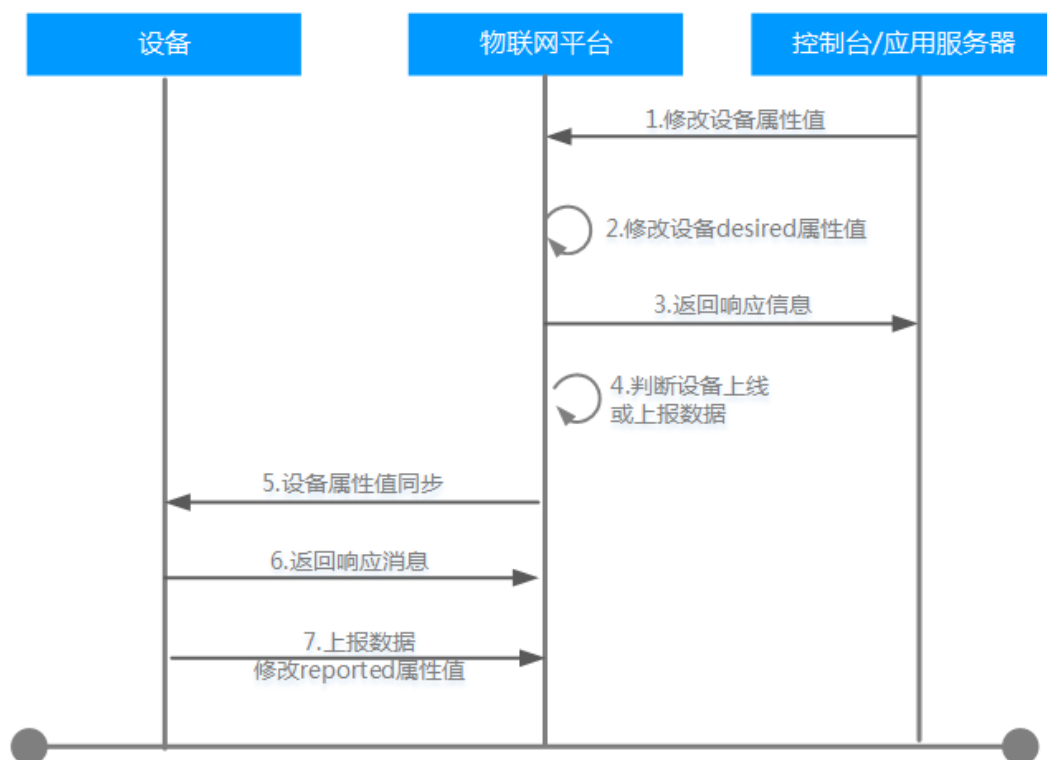
适合采用LWM2M协议的设备，特别是资源受限低功耗设备，长期处于休眠状态的场景。

- 查询设备最新上报数据和设备最新在线状态：
 - 当在控制台上查询设备上报数据时，由于设备可能长时间处于离线状态或因网络不稳定掉线，而无法获取到最新数据。通过设备影子机制，设备影子中始终保持设备最新上报的数据和设备当前状态，控制台上只需要查询设备影子中存储的数据，即可获取设备最新上报的数据和设备状态。
 - 很多应用服务器频繁地查询设备在线状态，由于设备处理能力有限，频繁查询会损耗设备性能。使用设备影子机制，设备只需要主动同步状态给设备影子一次，多个应用程序请求设备影子获取设备状态，即可获取设备最新状态，从而将应用程序和设备解耦。
- 修改设备属性值：用户通过**设备配置更新**功能修改设备的属性值，由于设备可能长时间处于离线状态，修改设备属性值的操作不能及时下发给设备。在这种情况下，物联网平台可以将修改设备的属性信息存储在设备影子中，待设备上线后，将修改的设备属性值同步给设备，从而完成设备属性值的修改。

业务流程

修改设备属性值

修改desired属性值，如果设备在线，则设备影子直接同步设备属性值到设备，否则等待设备上线或上报数据时，再同步设备属性值到设备。



1. 用户通过控制台或应用服务器修改设备属性值。消息样例如下：

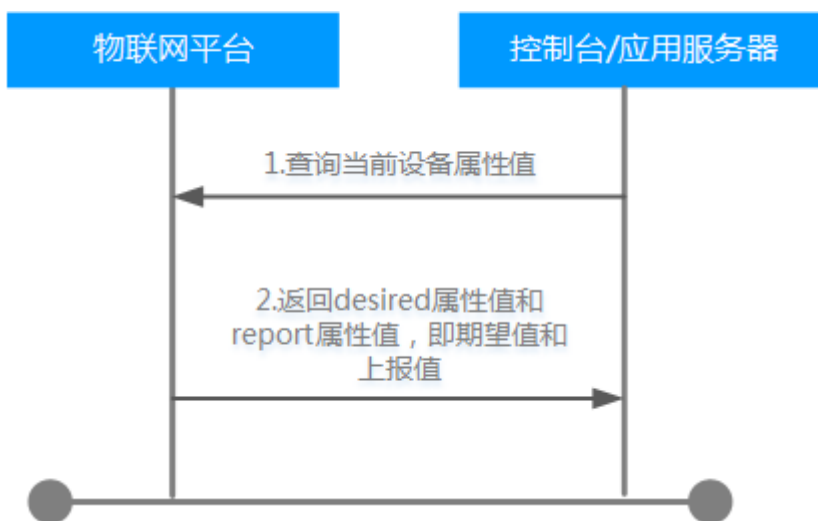
```
Method: PUT
request:
https://server:port/iocm/app/shadow/v1.5.0/devices/devices/{deviceId}?appld={appld}
Header:
```

```
app_key: *****
Authorization: Bearer *****
Content-Type: application/json
Body:
{
  "serviceDesireds": [
    {
      "serviceld": "Temperature",
      "desired": {
        "targetTemperature": 35
      }
    }
  ]
}
```

2. 物联网平台修改desired属性值。
3. 物联网平台返回响应消息。
4. 物联网平台判断设备上或上报数据。
5. 物联网平台将设备属性同步到设备。
6. 设备返回响应消息。
7. 当设备上上报数据时，物联网平台修改report属性值为设备上报的设备属性值。

查询设备属性值

设备影子保存的是设备最新的设备属性值，一旦设备属性值产生变化，设备会将设备属性值同步到设备影子。用户便可以及时获取查询结果，无需关注设备是否在线。



1. 用户通过控制台或应用服务器查询设备属性值。消息样例如下：

```
Method: GET
request:
https://server:port/iocm/app/shadow/v1.5.0/devices/{deviceld}
Header:
app_key: *****
Authorization: Bearer *****
Content-Type: application/json
```

2. 物联网平台返回desired属性值和report属性值，即期望值和上报值。消息样例如下：

```
response:
Status Code: 200 OK
Content-Type: application/json
Body:
{
  "deviceld": "*****",
  "gatewayld": "*****",
```


```
"nodeType": "*****",
"createTime": "*****",
"lastModifiedTime": "*****",
"deviceInfo": "*****"services": [
  {
    "serviceld": "*****",
    "reportedProps": "*****",
    "desiredProps": "*****",
    "eventTime": "*****",
    "serviceType": "*****"
  },
]
}
```

查询和修改设备影子

查询设备影子

方法1：应用服务器调用[查询设备影子](#)接口。

方法2：登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角

，在设备列表中，单击具体的设备进入到设备的详情页面，在“设备影子”页签中，可以查看当前设备属性数据，包括“上报值”和“期望值”，如下图所示。

- 如果当前界面中看到“上报值”与“期望值”不一致，原因可能是设备未在线，暂时存储在设备影子中，待同步给设备，期望值会存在深色底纹。



设备列表 > 设备详情 刷新

设备信息 历史数据 设备追踪 操作记录 消息跟踪 告警 **设备影子** 设置 命令 子设备

快照存储和检索设备属性信息。您可以通过命令、配置下发或者直接快照中修改设备属性配置，如果因为设备异常或者离线导致无法配置下发成功，则可以待设备正常上线后同步生效。

当前设备属性数据 属性配置 配置历史

服务	属性	上报值	期望值
Light	<RWE>light	11111 2019-02-26 19:35:32	11112 2019-02-26 19:35:32

- 如果当前界面看到的“上报值”与“期望值”一致，则表示设备最近一次上报的属性值与用户期望下发的属性值一致，期望值为白色底纹。



设备列表 > 设备详情 刷新

设备信息 历史数据 设备追踪 操作记录 消息跟踪 告警 **设备影子** 设置 命令 子设备

快照存储和检索设备属性信息。您可以通过命令、配置下发或者直接快照中修改设备属性配置，如果因为设备异常或者离线导致无法配置下发成功，则可以待设备正常上线后同步生效。

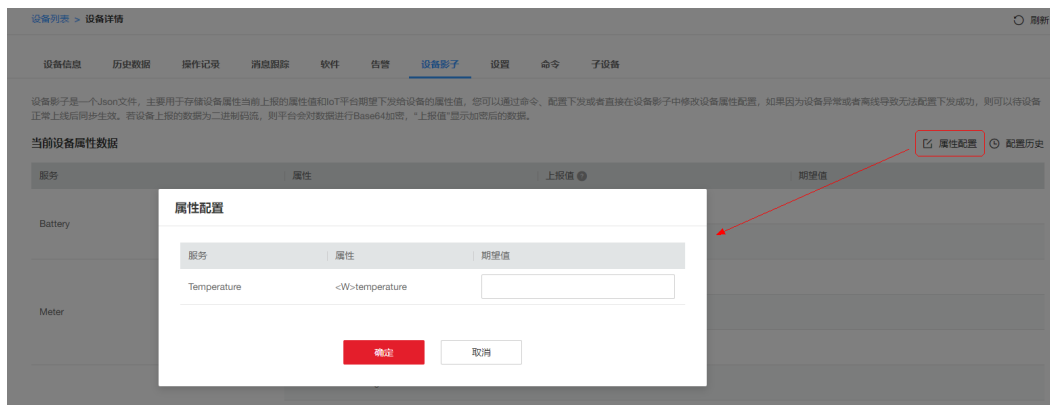
当前设备属性数据 属性配置 配置历史

服务	属性	上报值	期望值
Light	<RWE>light	11112 2019-02-26 19:36:21	11112 2019-02-26 19:36:21

修改设备影子

方法1：应用服务器调用[修改设备影子](#)接口。

方法2：登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。在设备列表中，单击具体的设备进入到设备的详情页面，在“设备影子”页签右上角选择“属性配置”，在弹出窗口中输入服务属性对应的期望值。



相关 API 接口

[查询设备影子](#)

[修改设备影子](#)

[查询设备影子历史数据](#)

10 规则引擎（联通用户专用）

非联通用户请查看[设备接入服务](#)。

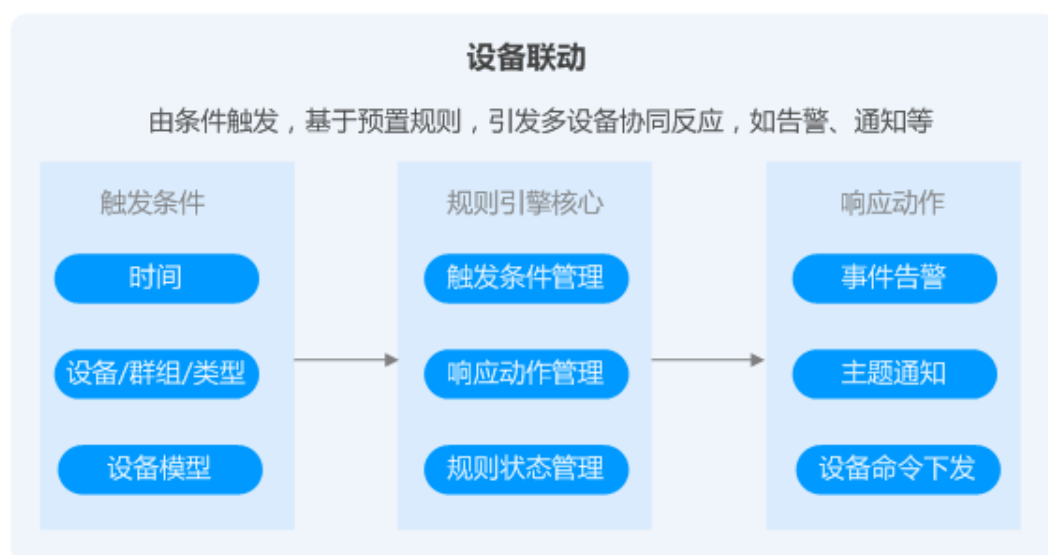
概述

规则引擎是指用户可以在物联网平台上可以对接入平台的设备设定相应的规则，在条件满足所设定的规则后，平台会触发相应的动作来满足用户需求。包含设备联动和数据转发两种类型。

设备联动规则

设备联动通过条件触发，基于预设的规则，引发多设备的协同反应，实现设备联动、智能控制。当响应动作为“主题通知”时，物联网平台对接华为云的[消息通知服务 SMN](#)，进行主题消息的设置和下发。

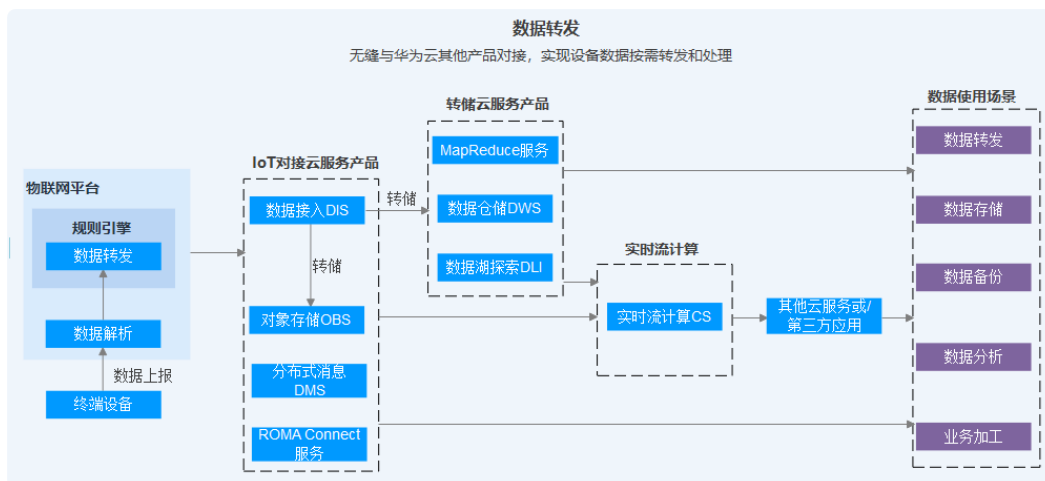
例如：设置水表的电池电量阈值为小于等于20%时，上报电池电量过低的告警，用户就能及时了解设备的供电情况，以便及时更换电池。



数据转发规则

数据转发无缝与华为云其他服务对接，实现设备数据按需转发和处理，您无需线下购买和部署服务器，即可实现设备数据的存储、计算、分析的全栈服务：

- 对接**数据接入服务DIS**，实现数据高效采集、传输、分发。用户可以通过DIS提供的SDK/API等方式下载数据，完成后续自定义使用数据的业务开发场景；也可以通过转储任务进一步将数据转发到其他云服务（**OBS**、**MapReduce**、**DWS**、**DLI**），进行数据存储、数据分析等后续数据处理，便于用户进行更灵活、多样化的数据使用。
- 对接**分布式消息服务DMS**，为设备数据提供消息队列服务。DMS是一项基于高可用分布式集群技术的消息中间件服务，用于收发消息。IoT作为生产者发送消息到DMS消息队列里，用户的应用程序作为消费者从消息队列里消费消息，从而做到往用户多个应用程序组件之间传输消息。
- 对接**对象存储服务OBS**，实现设备数据持久存储（设备管理服务支持设备数据存储，存储7天）。OBS是一个基于对象的海量存储服务，为客户提供海量、安全、高可靠、低成本的数据存储能力，适用于对设备上报数据进行归档和备份存储。OBS也支持对接**实时流计算CS云服务**，实时分析数据流，分析结果对接到其他云服务或者第三方应用进行数据可视化等。
- 对接**应用与数据集成平台ROMA Connect**的消息集成（Message Queue Service，简称MQS）组件，为物联网平台与应用服务器之间提供安全、标准化的消息通道。MQS是一款企业级消息中间件，基于Kafka协议，使用统一的消息接入机制，并具备消息发布订阅、Topic管理、用户权限管理、资源统计、监控报警等基础功能，以及消息轨迹、网络隔离、云上云下集成等高级特性，为企业数据管理提供统一的消息通道。



数据转发方案对比

在许多场景中，您需要将设备上报给物联网平台的数据进行加工处理或用于业务应用。使用物联网平台提供的**订阅推送**功能和数据转发功能，均可实现设备数据的转发。

- **数据转发**：提供初级的数据过滤能力。支持对设备数据进行过滤，然后再转发到其他华为云服务。
- **订阅推送**：第三方应用可在物联网平台订阅相关业务数据，当业务信息发生变化时（如设备注册、设备数据上报、设备状态变更等），平台会向第三方应用发送通知消息，通知其具体的变化信息。可快速地获取设备消息，无数据过滤能力，功能较为单一，但是简单易用且高效。

数据转发方案	适用场景	优缺点
数据转发	<ul style="list-style-type: none">设备上报数据上云。复杂场景。	优点： <ul style="list-style-type: none">支持将数据转发至其他华为云服务产品。支持对数据进行条件过滤。 缺点： <p>只能转发设备上报数据，不支持设备注册、设备状态变更等数据的转发。</p>
订阅推送	<ul style="list-style-type: none">设备数据推送至第三方应用。单纯的接收设备数据的场景。	优点： <ul style="list-style-type: none">支持将设备注册、设备数据上报、设备状态变更等多种数据推送至第三方应用。 缺点： <ul style="list-style-type: none">缺少过滤能力。第三方应用需要自实现对推送数据的存储、分析等操作，无法使用其他华为云服务。 <p>物联网平台默认只提供较弱的Http推送能力，高于10TPS的推送建议使用数据转发规则。</p>

云服务配置管理

物联网平台支持与华为云其它服务进行对接。首先通过华为云管理控制台获取“访问密钥（AK）”和“加密访问密钥（SK）”，在界面上输入AK和SK完成物联网平台与华为云的对接；然后用户在购买华为云其它云服务后，通过规则引擎功能将物联网平台的数据转发到用户购买的其它服务，或通过规则引擎功能与华为云的SMN服务对接。

📖 说明

AK（Access Key ID）：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和加密访问密钥一起使用，对请求进行加密签名。SK（Secret Access Key）：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

AK和SK的获取方法

步骤1 登录华为云，并进入管理控制台。

步骤2 单击用户名下的“账号中心”，如下图所示。




步骤3 在基本信息中，单击“管理我的凭证”，如下图所示。



步骤4 在“我的凭证”页面中，选择“访问密钥”页签，并单击“新增访问密钥”。

步骤5 输入当前用户的华为云登录密码，并输入短信验证码，单击“确定”按钮后，会自动下载访问密钥信息，包括“访问密钥（AK）”和“加密访问密钥（SK）”信息，请妥善保管。

步骤6 登录园区物联网平台Console，点击右上角“进入园区物联网平台服务”。单击页面右上角的用户头像，在下拉菜单中选择“云服务配置管理”，输入AK和SK，单击“保存”，完成和华为云的对接。

云服务配置管理

云服务密钥配置

[? 点击查看如何生成AK和SK密钥](#)

* 访问密钥 (AK)


* 加密访问密钥 (SK)

保存

----结束

创建设备联动规则

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角，选择“规则”，单击页面右上角的“创建规则”，选择“设备联动规则”。

步骤3 参考如下参数说明，设置规则。

参数名称	参数说明	最佳实践
规则名称	创建的规则名称。	<ul style="list-style-type: none">● 温度过高时自动关闭设备● 设备触发告警并邮件或短信通知
立即触发	平台创建规则前最后一次上报的数据满足触发条件，可以设置是否立即触发该规则。 <ul style="list-style-type: none">● 是：创建规则后规则立即生效，系统会对最近一次上报的数据做判断来触发规则。● 否：创建规则后，系统对新上报的数据进行判断，然后触发规则。	
规则类型	云端规则：设置直接接入到云端物联网平台的设备的规则。 边缘规则：设置通过边缘节点接入云端物联网平台的设备的规则。	
边缘节点	“规则类型”选择为“边缘规则”时出现，选择规则针对的节点设备。	
时间设置	<ul style="list-style-type: none">● 总是：没有时间限制，持续检查当前规则条件是否满足。● 在特定时间：可以选择时间段，在特定的时间检查规则条件是否满足。	

参数名称	参数说明	最佳实践
条件	<p>设备行为：单击“添加”，可以选择设置条件的设备，支持对某一设备或者某一类设备进行设置。</p> <ul style="list-style-type: none">● 条件类型：<ul style="list-style-type: none">- 设备：表示满足条件的数据是由单个设备上报到平台。- 设备类型：表示满足条件的数据是由选择的某一设备类型的设备上报到平台。- 设备群组：表示满足条件的数据是由所选设备群组的设备上报到平台。● 选择设备模型：选择满足条件后上报数据的设备的模型。选择设备模型后需要选择对应的服务类型，并设置上报数据的规则。● 数据时效(秒)：数据有效时间，例如：设备产生数据时间为19:00，时效设为30分钟，平台收到数据时间为20:00，该情况下即使满足触发条件也不触发动作。● 延后触发(分钟)：满足条件后，动作延迟触发时间。默认为0，例如：设置门磁打开状态时触发门未锁告警。当监测到门磁打开时，则立即触发告警。 <p>时间：单击“添加”，可以设置规则触发的时间点，该条件一般用于周期性的触发条件，如每天7:00，关闭路灯。</p> <ul style="list-style-type: none">● 时间点：可以选择规则触发的起始时间点。● 重复次数：规则重复触发的次数（1~1440次）。● 重复间隔（分钟）：在起始时间点后，重复触发规则的时间间隔（1~1440分钟）。	

参数名称	参数说明	最佳实践
动作	<p>设备行为：单击“添加”，可以设置在规则触发后，需要执行相应动作的设备。</p> <ul style="list-style-type: none">● 动作类型：执行动作的类型，默认选择“设备”，表示给单个设备下发命令。● 选择设备模型：选择设置动作对应的设备模型，然后选择该模型下执行动作的设备。选择设备模型后需要选择对应的服务类型。● 命令状态：动作有效性标志，默认选择“启用”。<ul style="list-style-type: none">- 启用：表示规则的执行动作有效，规则满足条件后会执行该动作。- 禁用：表示规则执行的动作无效，规则满足条件后不会执行该动作。● 命令请求ID：用于标识此次命令下发的ID。● 回调地址：命令状态变化通知地址，当命令状态变化时(执行失败，执行成功，超时，发送，已送达)会通知应用服务器。● 过期时间（秒）：命令有效的超期时间，单位为秒。当为0时，表示立即下发命令，其他值表示缓存下发命令。不传时，默认为48小时。 <p>发送消息或告警：单击“添加”，可以设置主题通知、事件告警。</p> <ol style="list-style-type: none">1. 主题通知：需要开通SMN云服务，然后再此页面进行主题内容的配置。<ol style="list-style-type: none">a. 在系统管理->云服务配置管理，与华为云对接。b. 在本界面上点击SMN云服务链接，跳到华为云官网上开通SMN云服务。c. 在本界面上选择SMN云服务对应所在区域，选择主题名称，设置消息标题和消息内容。2. 事件告警：定义告警级别、告警名称、告警内容等，表示在告警管理界面产生一条对应的设备告警。 <p>说明 设置上报事件告警的执行动作，则条件中不能设置“时间”的触发条件。</p>	
描述	对该规则的描述。	

步骤4 单击“提交”，完成设备联动规则的创建。

新创建的规则默认状态为“激活”，您可以在列表中“规则状态”列，对规则进行禁用。

---结束

创建数据转发规则

- 步骤1** 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。
- 步骤2** 进入“规则”页面，单击页面右上角的“创建规则”，选择“数据转发规则”。
- 步骤3** 参考如下参数说明，设置规则。

参数名称	参数说明	最佳实践
规则名称	创建的规则名称	<ul style="list-style-type: none"> 设备数据转发至OBS长期储存
条件	<ul style="list-style-type: none"> 对象类型：选择“所有设备”（只支持所有设备的数据转发）。 数据类型：转发的数据类型。 <ul style="list-style-type: none"> 设备上报的数据：设备根据Profile中定义的字段上报的数据。 设备信息：设备注册时填写的设备静态信息数据，如设备位置信息、IMEI等。 添加过滤条件：开关默认为“关闭”状态，当开关打开时，需要设置“属性名称”、“操作”、“值”参数，物联网平台对符合过滤条件的设备进行转发，详见转发数据样例。 	
动作	<ul style="list-style-type: none"> 动作类型：根据实际情况，选择“DIS服务”、“DMS服务”、“OBS服务”、或者“ROMA服务”。 <p>说明 如果您没有开通服务，请参照以下步骤操作：</p> <ol style="list-style-type: none"> 在系统管理->云服务配置管理，与华为云对接。 在本界面上点击具体的云服务链接，跳到华为云官网上开通相应的服务。其中： <ul style="list-style-type: none"> DMS服务：只支持转发到Kafka专享版。 ROMA服务：开通ROMA服务后，还需要在“应用详情->服务设置”里填写“订阅推送服务（ROMA）”，完成与ROMA服务的对接。 <ul style="list-style-type: none"> 数据转发至：如图创建数据转发规则界面所示，选择您需要转发的服务所在的区域、通道、数据类型等信息，不同服务填写的信息稍有差别，请以界面选项为准。 	
描述	对该规则的描述。	

- 转发数据样例：**
物联网平台对编解码后的Json格式的设备数据进行解析匹配：

```
{
  "deviceId": "6543214c-72bb-4131-9e14-fad974123456",
  "services": [{
    "serviceId": "serviceId1",
    "data": {
      "property1": "reportValue1",
```

```
    "property2": "reportValue2",  
    "property3": "reportValue3"  
  },  
  "eventTime": "20190425T021157Z"  
}  
}
```

数据转发添加过滤条件：

- 根据deviceId过滤，转发单设备数据。

条件

对象类型: 所有设备

添加过滤条件: 打开

属性名称	操作	值
deviceId	=	6543214c-72bb-4131-9e14-fad974123456

- 根据服务ID、或者上报数据属性property1/property2/property3过滤，转发批量设备数据。

条件

对象类型: 所有设备

添加过滤条件: 打开

属性名称	操作	值
property2	=	reportValue2

● 创建数据转发规则界面

动作

动作类型: DIS服务

关于DIS

数据转发至

区域	网络
cn-north-1	dis-8CQ8

数据类型: JSON

描述: <=256字

步骤4 单击“提交”，完成数据转发规则的创建。

新创建的规则默认状态为“激活”，您可以在列表中“规则状态”列，对规则进行禁用。

----结束

相关推荐

相关文档	相关产品
<ul style="list-style-type: none">• 6 数据上报（联通用户专用）• 5 订阅推送（联通用户专用）	<ul style="list-style-type: none">• 消息通知服务SMN• 数据接入服务DIS• 分布式消息服务DMS• 对象存储服务OBS• 应用与数据集成平台ROMA Connect

11 群组与标签（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

群组

群组是一系列设备的集合，用户可以对应用下所有设备，根据区域、类型等不同规则进行分类建立群组，以便处理对海量设备的批量操作，例如，对应用下所有水表设备的群组进行固件升级。支持群组的增删改查操作，支持给群组绑定和解绑设备，支持将某个群组里的设备解绑后移动到其他群组。

一个设备只能绑定一个群组。

标签管理

物联网平台支持定义不同的标签，并对设备打标签，通过标签，可以快速筛选设备。







管理群组

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角，选择“群组”。

步骤3 点击相关按钮，进行添加群组、解绑群组、挪动群组、删除群组操作，相关说明如下表所示。



图标	图标说明
	单击可增加根群组，用户自定义群组名称、群组的描述。
	单击可添加子群组，需要填写群组名称和群组描述。
	单击后可将子群组从父级群组中解绑。
	单击可将子群组从现有父群组移动到选择的父群组中。
	单击可删除当前选择的群组，并且删除操作不可撤回。默认群组不可删除。 删除分为“级联删除”和“删除当前”： <ul style="list-style-type: none"> “级联删除”：删除群组时，该群组下的子群组会同时删除。 “删除当前”：删除群组后，该群组下的子群组会升级为根群组。当存在多个子群组时，会产生相应数量的根群组。例如根群组A，其子群组为B和C，若删除A，则B和C都将成为独立的根群组。
	选择“所有群组”后在界面右边呈现，单击可导出所有群组的拓扑结构。

----结束

给群组绑定/解绑设备


当创建完群组后，可以进行设备和群组的绑定和解绑，也可以进行设备在不同群组之间的移动。



步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角 ，选择“群组”。

步骤3 选择某个群组，进入“设备”页面，单击相关按钮，进行设备绑定、设备解绑、设备移动操作，相关说明如下表所示。



图标	图标说明
	单击可绑定设备。

图标	图标说明
	选中设备后，单击后可将选中的设备从群组中解绑。 说明 具有权限的SP帐户或者全局用户（普通用户无法进行设备解绑）进行解绑操作，如果该群组下已绑定若干设备，解绑后，该用户无权限查看该群组和其下的设备，对于该用户来说，解绑后的群组及设备会消失。
	选择一个设备，单击该图标可以移动选择的设备至其他群组。

----结束

管理标签

标签是一种分类方式，在此处定义不同的标签，在注册设备时为设备选择标签，或者在已经注册设备的“设备详情”中为设备选择标签，这样就可以在“所有设备”列表中通过标签筛选设备。

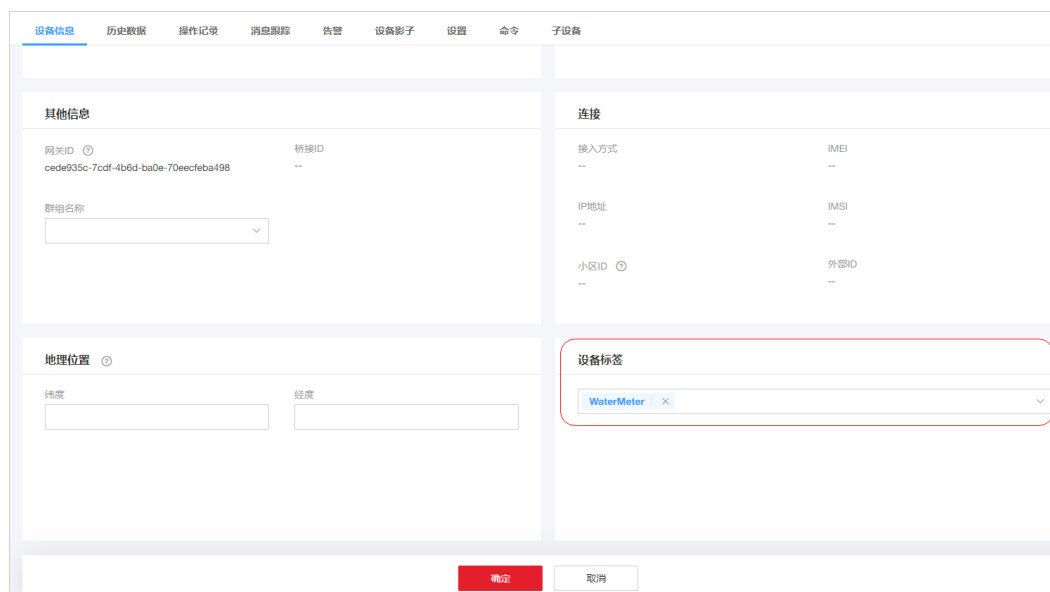
步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角 ，选择“设备 > 标签”。

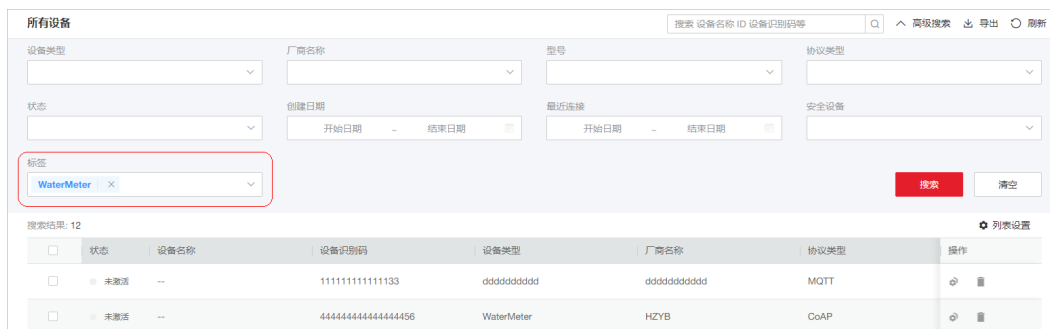
步骤3 单击页面右上角的“添加标签”，填写标签信息。



步骤4 在设备列表中，点击具体的设备进入到设备的详情页面，点击“编辑”按钮，给设备添加标签。



步骤5 访问“所有设备”页面，在高级搜索里通过标签，对目标设备进行快速筛选。



----结束

相关 API 接口

- [创建设备组](#)
- [删除设备组](#)
- [修改设备组](#)
- [查询设备组列表](#)
- [查询指定设备组](#)
- [增加设备组成员](#)
- [删除设备组成员](#)
- [查询指定设备组成员](#)

12 设备监控（联通用户专用）


非联通用户请查看[设备接入服务](#)。

概述

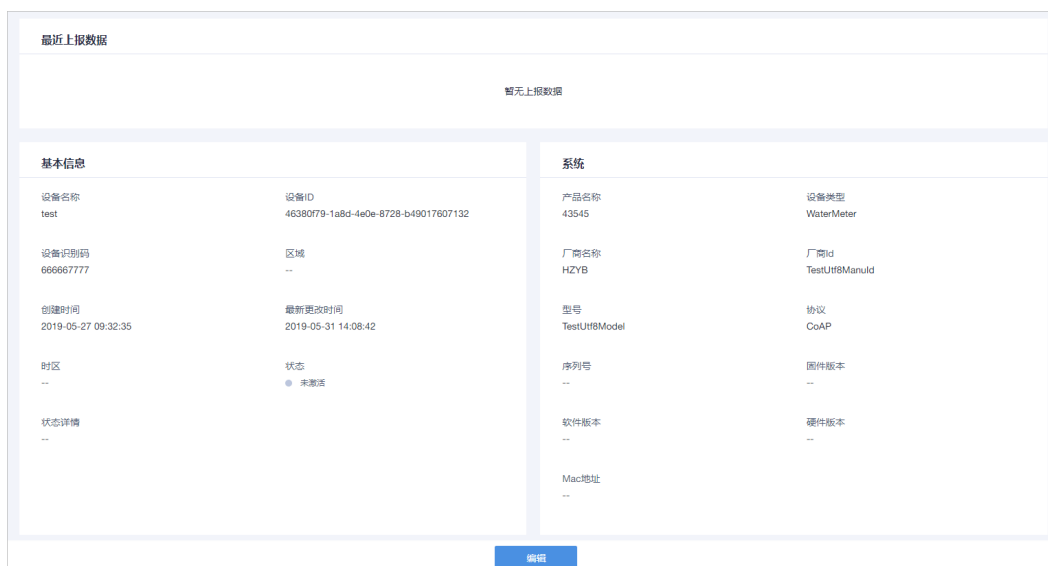
物联网平台提供查看设备详情、设备状态管理、查看报表、查看操作记录、查看审计日志、告警管理、设备消息跟踪等设备监控与运维能力，提升设备的可维护性。

功能	简述
查看设备详情	在控制台上可以查看每个设备的设备注册和接入时的基本信息。
查看设备状态	在控制台上可以查看设备当前状态，如在线、离线、未激活。用户也可以通过订阅方式获取设备的状态信息。
查看报表	控制台提供了丰富的报表功能，方便用户查看设备的使用情况，如设备在线数、API调用数等。
查看操作记录	可以通过控制台查看用户对设备的操作日志，操作类型包括重启模组、配置更新、软件升级等，方便用户在进行问题定位时，查看历史操作记录，以及操作的执行结果。
查看审计日志	用户在使用物联网平台的过程中，系统会以日志形式收集并记录用户及平台的操作及结果，当某项功能发生异常时，用户可以根据日志的记录信息定位并处理故障问题。
告警管理	如果用户在控制台上设置 规则引擎 时，定义了响应动作为上报告警，且定义了告警属性、告警级别等，则当满足触发条件时，系统就会上报告警，需要用户密切关注设备的告警并及时进行处理，确保设备的正常运行。
设备消息跟踪	在设备绑定、命令下发、数据上报、设备信息更新和设备监控业务场景中出现故障时，物联网平台可以通过消息跟踪功能进行快速的故障定位和原因分析。

查看设备详情

登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，选择“设备 > 所有设备”，在设备列表中，单击具体的设备进入到设备的详情页面，即可查看设备注册和接入时的基本信息，比如设备标识码，设备ID，接入协议等。如下图所示，单击“编辑”按钮，可以对部分信息进行修改。

用户也可通过调用[修改设备信息](#)接口修改设备的基本信息。



最近上报数据

暂无上报数据

基本信息		系统	
设备名称 test	设备ID 4638079-1a8d-4e0e-8728-b49017607132	产品名称 43545	设备类型 WaterMeter
设备标识码 666667777	区域 --	厂商名称 HZYB	厂商Id TestUHBManuld
创建时间 2019-05-27 09:32:35	最新更新时间 2019-05-31 14:08:42	型号 TestUHBModel	协议 CoAP
时区 --	状态 <input checked="" type="radio"/> 未激活	序列号 --	固件版本 --
状态详情 --		软件版本 --	硬件版本 --
		Mac地址 --	

编辑


查看设备状态

在控制台上可以查看设备当前状态，如在线、离线、未激活、异常。用户也可以通过[订阅方式](#)获取设备的状态信息。设备状态定义如下：

状态	短连接设备（如NB-IoT设备）	长连接设备（采用Agent Lite SDK或者原生MQTT）
在线	如果在25小时内设备有上报过数据，设备的状态为“在线”；如果在25小时内未上报过数据，设备的状态会变为“离线”。	设备与平台之间一直连接，无断开。
离线	设备接入平台后，设备在超过25小时未上报数据，平台会将设备置为“离线”状态。	设备与平台之间的连接断开1分钟后（数据自动刷新周期为1分钟），置为“离线”状态。 如果在界面上手动刷新状态，则直接显示“离线”。
异常	设备接入平台后，设备在超过169小时未上报数据，平台会将设备置为“异常”状态。	无此状态。

状态	短连接设备（如NB-IoT设备）	长连接设备（采用Agent Lite SDK或者原生MQTT）
未激活	已在平台上完成设备注册但真实设备还未接入平台。请根据 迁移设备 操作完成设备的接入。	已在平台上完成设备注册但真实设备还未接入平台。请根据 迁移设备 操作完成设备的接入。

查看方法：

登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，选择“设备 > 所有设备”，在设备列表中，单击具体的设备进入到设备的详情页面，即可查看设备当前状态。



同时，在控制台上的“所有设备”界面，还可以查看当前应用下的所有设备总数、在线数、离线数、未激活数、异常数。数据的刷新周期为每分钟刷新一次。




查看报表

物联网平台为用户提供了丰富的报表功能，能够将数据直观地呈现出来，同时支持以月或者天为周期，导出统计数据。选择“设备管理 > 首页”进入报表页面，报表名称及功能详细说明如下表所示。

报表名称	报表说明	数据刷新频率
设备总数	统计当前应用下注册的所有设备总数、在线数、离线数、未激活数和异常设备数。单击查看以月或者天为周期的趋势图。	每分钟
API调用数（日）	统计每日API的调用次数，单击查看每个API被调用的统计、当前应用所调用各种API的统计。	每小时
上报消息数（日）	统计每日设备上报的消息数。单击查看上报数据趋势和平均消息上报速率（次数/秒）。	每小时
下发命令数（日）	统计每日平台下发的命令数。单击查看下发消息趋势。	每小时
设备总趋势	统计设备总数和设备在线数的趋势。单击查看以月或者天为周期的趋势图。	每小时
设备在线率	统计设备在线情况，以百分比形式呈现。	每小时
设备数量统计图	统计新增设备数、删除设备数、激活设备数、离线设备数和异常设备数。	每小时
应用侧推送统计	统计应用侧接口调用推送消息数。	每天0点刷新
命令状态统计	统计平台下发的所有命令数。包括等待命令，超时命令，失败命令，成功命令，已发送命令，取消命令，已送达命令和超期命令。	每天0点刷新
应用告警统计	统计当前应用下所有设备每天产生的告警数。	每天0点刷新
活跃设备统计	统计过去1小时内有上报过数据的设备数。	每小时
引导激活设备统计	统计通过“设备引导服务”方式注册并激活的设备数。	5分钟
软件升级状态	统计当前应用下，历史上所有设备的软件升级任务，执行成功和失败的设备数量。	实时
固件升级状态	统计当前应用下，历史上所有设备的固件升级任务，执行成功和失败的设备数量。	实时
设备配置状态	统计当前应用下，历史上所有设备的设备配置更新任务，执行成功和失败的设备数量。	实时

查看操作记录

操作记录功能用于记录用户对于该设备进行的一系列历史操作，方便用户在进行问题定位时，查看历史操作情况，以及操作的执行结果（成功或失败）。

登录**物联网平台控制台**，点击右上角“进入设备管理服务”。单击左下角，选择“设备 > 所有设备”，在设备列表中，单击具体的设备进入到设备的详情页面，选择“操作记录”页签，即可查看设备操作记录。

在查看操作记录时，首先需要对“操作类型”进行过滤，精确的查看具体任务类型的操作情况。操作类型包括配置更新、同步任务、软件升级和固件升级等。



查看审计日志

用户在使用物联网平台的过程中，系统会以日志形式收集并记录用户及平台的操作及结果，当某项功能发生异常时，用户可以根据日志的记录信息定位并处理故障问题。

物联网平台提供在控制台上根据日志类型（包括操作日志、安全日志、个人数据查询日志、业务日志）、结果、创建时间对日志进行搜索和查看，同时能够导出筛选后的日志。

日志类型	日志说明
操作日志	记录用户和系统所做的操作和结果，用于跟踪和审计。 对应用的操作： 如在应用详情下，进行保存配置、恢复默认配置、上传 logo、查询详情等操作。 对资源的操作： 如导入产品模型、查看应用首页报表、查询设备详情等。
安全日志	记录涉及系统安全操作信息，如创建用户、登录、用户确认隐私声明等，用户了解系统安全操作相关信息，及时发现潜在的安全隐患并进行处理。
个人数据查询	记录用户对账户进行的操作和结果，如查询用户列表、查询企业列表等，用于跟踪和安全审计。
业务日志	记录涉及平台业务的操作及结果，例如创建设备、设备登录等，用于跟踪设备侧、应用侧业务相关信息，及时发现问题并处理。

日志管理						
日志类型	结果	创建日期				
操作	全部	开始日期	结束日期			搜索 重置
总计:1210 导出日志						
账户	操作	对象	详情	时间	IP地址	结果
e00270744	Query or download a...	Active Devices Statist...	body : {conditions : r...	2019/08/24 11:40:10...	100.125.64.129	SUCCESS
e00270744	Query or download a...	BootStrap Devices St...	body : {conditions : r...	2019/08/24 11:40:10...	100.125.64.135	SUCCESS
e00270744	Query or download a...	Application Alarm St...	body : {conditions : r...	2019/08/24 11:40:10...	100.125.64.139	SUCCESS
e00270744	Query or download a...	Device Statistics	body : {conditions : r...	2019/08/24 11:40:10...	100.125.64.136	SUCCESS
e00270744	Query or download a...	Push Message Statist...	body : {conditions : r...	2019/08/24 11:40:10...	100.125.64.138	SUCCESS

告警管理

如果用户在控制台上设置**规则引擎**时，定义了响应动作为上报告警，且定义了告警属性、告警级别等，则当满足触发条件时，系统就会上报告警，需要用户密切关注设备的告警并及时进行处理，确保设备的正常运行。

例如：智能水表设备3天未上报数据，可能存在水表设备发生故障导致，平台会产生对应的告警，维护人员可通过告警信息找到对应告警的水表设备，进行快速定位维修。

告警的等级划分及处理建议：

- 致命：已造成业务中断或可能引起设备不可用的故障，需要立即处理。
- 严重：对设备产生部分影响或对系统性能造成影响的故障，需要采取纠正措施，防止更为严重的故障发生。
- 一般及警告：对设备或系统没有影响，系统检测到潜在或即将发生影响业务的故障，需排查潜在故障。

用户在控制台上可以查看某应用下的所有设备的告警信息，也可以查看单个设备的告警信息。

所有设备告警信息

在“告警”界面上：

告警 高级搜索 ▾						
呈现用户通过规则引擎定义产生的该应用下所有的告警信息。可以通过定义的规则来对设备的状态进行管理。						
等级	告警名称	来源	地点	告警状态	上报时间	
 暂无数据						


单个设备告警信息

在设备列表中，单击具体的设备进入到设备的详情页面，选择“告警”页签。



设备消息跟踪

在设备绑定、命令下发、数据上报、设备信息更新和设备监控业务场景中出现故障时，物联网平台可以通过消息跟踪功能进行快速的故障定位和原因分析。目前物联网平台支持NB-IoT设备和MQTT设备的消息跟踪，单个应用下，最多支持同时进行跟踪的设备数上限为10。

步骤1 登录**物联网平台控制台**，点击右上角“进入设备管理服务”。单击左下角，进入“设备 > 所有设备”界面。

步骤2 通过检索条件，快速找到需要跟踪的设备，并点击进去设备详情。

步骤3 在“设备详情 > 消息跟踪”页签中，点击“启动”或“重启”按钮，并设置设备的消息跟踪时间，如下图所示，消息跟踪时长表示从启动消息跟踪功能开始，在设置的时间段内进行消息跟踪，重启消息跟踪后，该时间重新开始计算。

说明


再次启动消息跟踪，将会清除上次消息跟踪的历史数据。



步骤4 用户可以在“设备管理”->“消息跟踪设备”界面上，看到所有正在消息跟踪的设备列表（如果单击停止消息跟踪按钮，则该条消息跟踪记录从列表上消失），方便集中查看。



步骤5 跟踪结果和示例如下图所示，相同流水号的跟踪记录表示为通一次业务处理的结果。其中：

- 蓝色字体：表示该节点对消息的处理状态为成功。
- 红色字体：表示节点对消息的处理失败。可以通过点击跟踪记录后的按钮，查看跟踪结果的详情信息，并根据失败处理建议，进行问题的定位分析。



启动消息跟踪任务后，用户即可对设备绑定、命令下发、数据上报、设备信息更新和设备监控业务场景进行消息跟踪。如果在跟踪的结果中数据量较多时，用户可以根据“业务类型”、“节点名称”和“消息状态”进行过滤；如果对跟踪结果数据需要做进一步的分析，用户可以通过“导出数据”功能，将跟踪的结果数据导出，以便物联网平台运维人员做进一步的分析。

----结束

相关 API 接口

- [查询单个设备信息](#)
- [批量查询设备信息](#)
- [修改设备信息](#)

13 远程诊断（联通用户专用）

非联通用户请查看[设备接入服务](#)。

概述

物联网平台支持用户对接入的设备进行远程维护操作，快速定位问题及恢复业务，减少近端维护引入的高成本。当前仅支持通过控制台对NB-IoT设备进行日志收集、重启模组。

物联网平台支持基于LWM2M协议的日志收集和重启模组能力，如下表所示。

资源路径	对象名	资源名	对应物联网平台的功能
/3/0/4	Device	Reboot	远程重启模组
/20/0/4014	Event Log	LogData	模组日志收集

因此，NB-IoT设备需要支持上述LWM2M协议接口里定义的这2个资源，同时设备对应的产品模型的omCapability不为null，才能接收和响应物联网平台下发的指令，进行远程的日志收集、重启模组。

日志收集

前提条件

- 仅NB-IoT设备支持日志收集。
- 设备要支持远程日志收集的能力，即设备支持上述LWM2M协议接口里定义的LogData资源。
- 设备对应的**3 产品模型（联通用户专用）**的protocolType必须为CoAP或LWM2M，且omCapability不为null。

操作步骤

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 在设备列表中，单击具体的设备进入到设备的详情页面，选择“设置”页签。

步骤3 点击“收集”按钮后，需要等待设备上线或上报数据时，物联网平台才会将日志收集的指令下发到设备。



步骤4 日志收集的任务下发后，可以点击“下载”按钮，查看和下载收集的日志文件。日志收集的任务，会存在以下几种状态。

- **Waiting**: 表示任务处于等待状态，创建任务后，只有设备上线或上报数据后，物联网平台把日志收集的指令下发给设备后，状态才会变更为Processing；如果超过25h设备未上报数据，则状态会变更为Fail。
- **Processing**: 表示日志收集指令已下发给设备，正等待设备反馈日志收集结果，设备处理完成后会给平台返回处理结果，根据反馈的结果变更未Success或Fail状态；如果超过30分钟未收到反馈结果，则状态变更为Fail。
- **Success**: 表示日志收集成功，根据设备反馈的日志收集执行结果，如果收集日志成功，则变更为Success状态。
- **Fail**: 表示日志收集失败。物联网平台超过25h未将日志收集的指令下发给设备，状态会变更为Fail；设备收到物联网平台下发的日志收集指令后，30分钟未收到反馈结果，则状态变更为Fail；物联网平台收到设备反馈的日志收集结果为失败，则状态会变更为Fail。

----结束

重启模组

前提条件

- 仅针对NB-IoT模组进行重启。
- 设备要支持远程重启模组的能力，即设备支持上述LWM2M协议接口里定义的Reboot资源。
- 设备对应的**3 产品模型（联通用户专用）**的protocolType必须为CoAP或LWM2M，且omCapability不为null。

操作步骤

步骤1 登录**物联网平台控制台**，点击右上角“进入设备管理服务”。

步骤2 在设备列表中，单击具体的设备进入到设备的详情页面，选择“设置”页签。

步骤3 点击“重启”按钮，需要等待设备上线或者上报数据时，物联网平台才会将重启的指令下发到设备。



步骤4 重启模组的任务执行后，可以在**查看操作记录**中，查看任务状态。在重启模组的任务中，存在以下几种状态。

- **Waiting**: 表示任务处于等待状态，创建任务后，系统设置了一个10秒的定时器，10秒过后，变更为Processing状态。
- **Processing**: 表示物联网平台正在处理任务状态，如果设备在25h内未上线或上报数据，则变更为Fail状态；如果设备在25h内上线或上报数据，则平台会将重启模块的指令下发给设备，设备处理完成后会给平台返回处理结果，根据反馈的结果变更为Success或Fail状态。
- **Success**: 表示重启模块成功。根据设备反馈的重启模块执行结果，如果重启模块成功，则变更为Success状态。
- **Fail**: 表示重启模块失败。物联网平台超过25h未将重启模块的指令下发给设备，状态会变更为Fail；设备上报数据后，物联网平台将重启模块指令下发给设备，如果30分钟内未收到设备的反馈结果，则状态会变更为Fail；物联网平台收到设备反馈的模块重启结果为重启失败，则状态会变更为Fail。

----结束

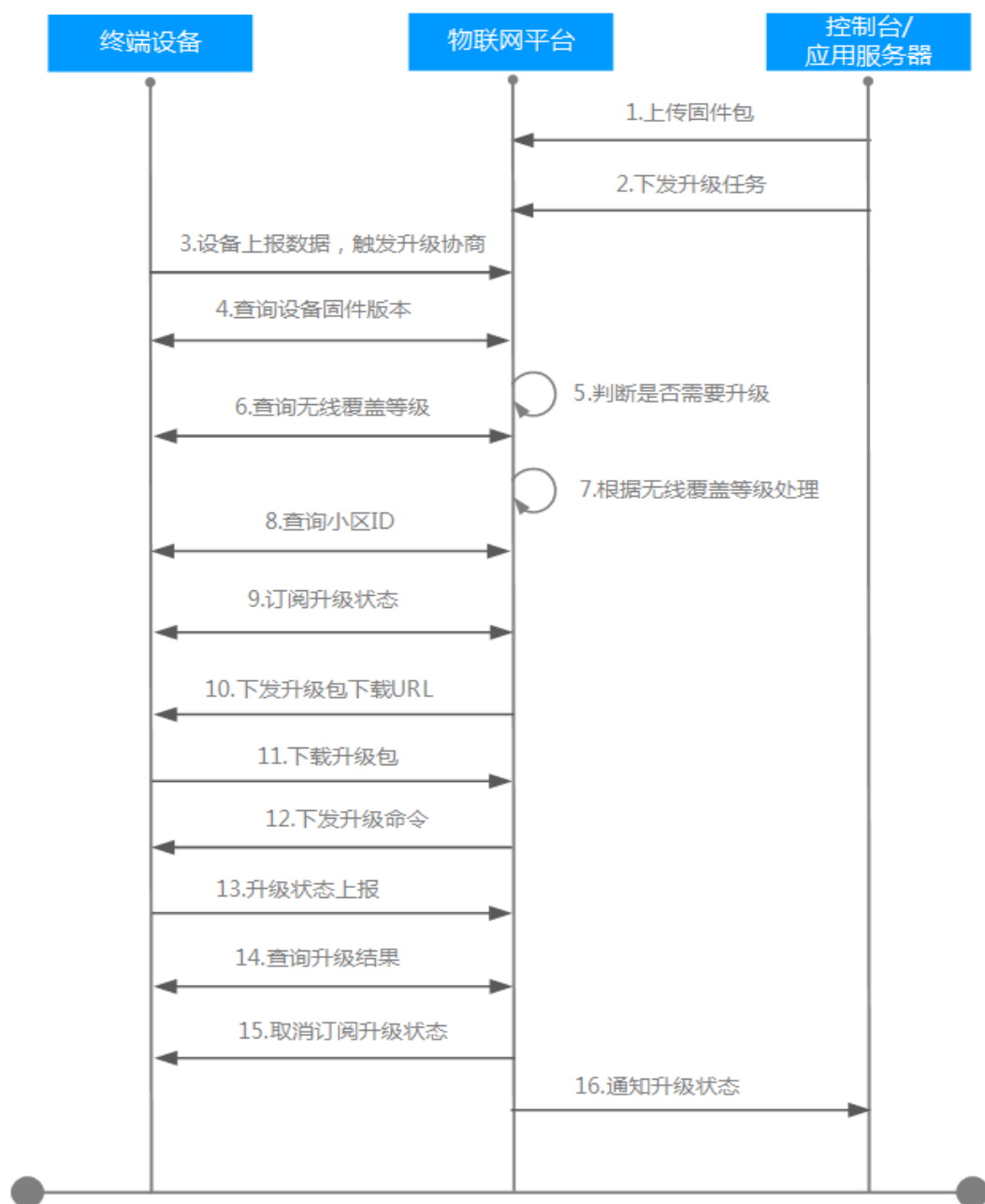
14 固件升级（联通用户专用）

概述

固件（Firmware）一般是指设备硬件的底层“驱动程序”，承担着一个系统最基础最底层工作的软件，比如计算机主板上的基本输入/输出系统BIOS（Basic Input/output System）。

固件升级又称为FOTA（Firmware Over The Air），是指用户可以通过OTA的方式对支持LWM2M协议和MQTT协议的设备进行固件升级。物联网平台支持LWM2M协议和MQTT协议的产品模型的固件升级。

LWM2M 协议固件升级流程



LWM2M协议FOTA升级流程的详细说明：

1~2. 用户在设备管理服务的控制台上传固件包，并在控制台或者应用服务器上创建固件升级任务。

3. NB-IoT设备上报数据，平台感知设备上线，触发升级协商流程。

4~5. 物联网平台向设备下发查询设备固件版本的命令，查询成功后，物联网平台根据升级的目标版本判断设备是否需要升级。

- 如果返回的固件版本信息与升级的目标版本信息相同，则升级流程结束，不做升级处理。
- 如果返回的固件版本信息与升级的目标版本信息不同，则继续进行下一步的升级处理。

6~7. 物联网平台查询终端设备所在的无线信号覆盖情况，获取小区ID、RSRP（Reference Signal Received Power，参考信号接收功率）和SINR（Signal to Interference Plus Noise Ratio，信号干扰噪声比）信息。

- 查询成功：则根据如下方式计算可同时升级的并发数计算，并按照[步骤9](#)进行处理。
 - 如下图所示，如果设备的RSRP强度和SINR强度均落在等级“0”中，则同时可以对小区的50个相同信号覆盖区间的设备进行同时升级。
 - 如果设备的RSRP强度和SINR强度分别落在等级“0”和“1”中，则以信号较弱的等级“1”为准，则只能同时对小区的10个设备进行升级。
 - 如果设备的RSRP强度和SINR强度分别落在等级“1”和“2”中，则以信号较弱的等级“2”为准，则只能同时对小区的1个设备进行升级。
 - 如果设备的RSRP强度和SINR强度不在该3个等级范围内，且均可以查询到，则按照信号最弱覆盖等级“2”处理，则只能同时对1个设备进行升级。

无线覆盖等级	RSRP门限范围 (dBm)	SINR门限范围 (dB)
0	$-105 \leq \text{RSRP}$	$7 \leq \text{SINR}$
1	$-115 \leq \text{RSRP} < -105$	$-3 \leq \text{SINR} < 7$
2	$-125 \leq \text{RSRP} < -115$	$-8 \leq \text{SINR} < -3$

说明

如果用户在固件升级中发现同时进行升级的设备数较少，则可以联系当地运营商检查和优化设备所在小区的无线覆盖情况。

- 查询失败：则按照[流程步骤8](#)进行处理。
8. 物联网平台继续下发查询小区ID信息的命令，获取终端设备所在的小区ID信息。
- 如果查询成功：物联网平台支持同时对该小区的10个相同情况的设备进行固件升级。
 - 如果查询失败：则升级失败。

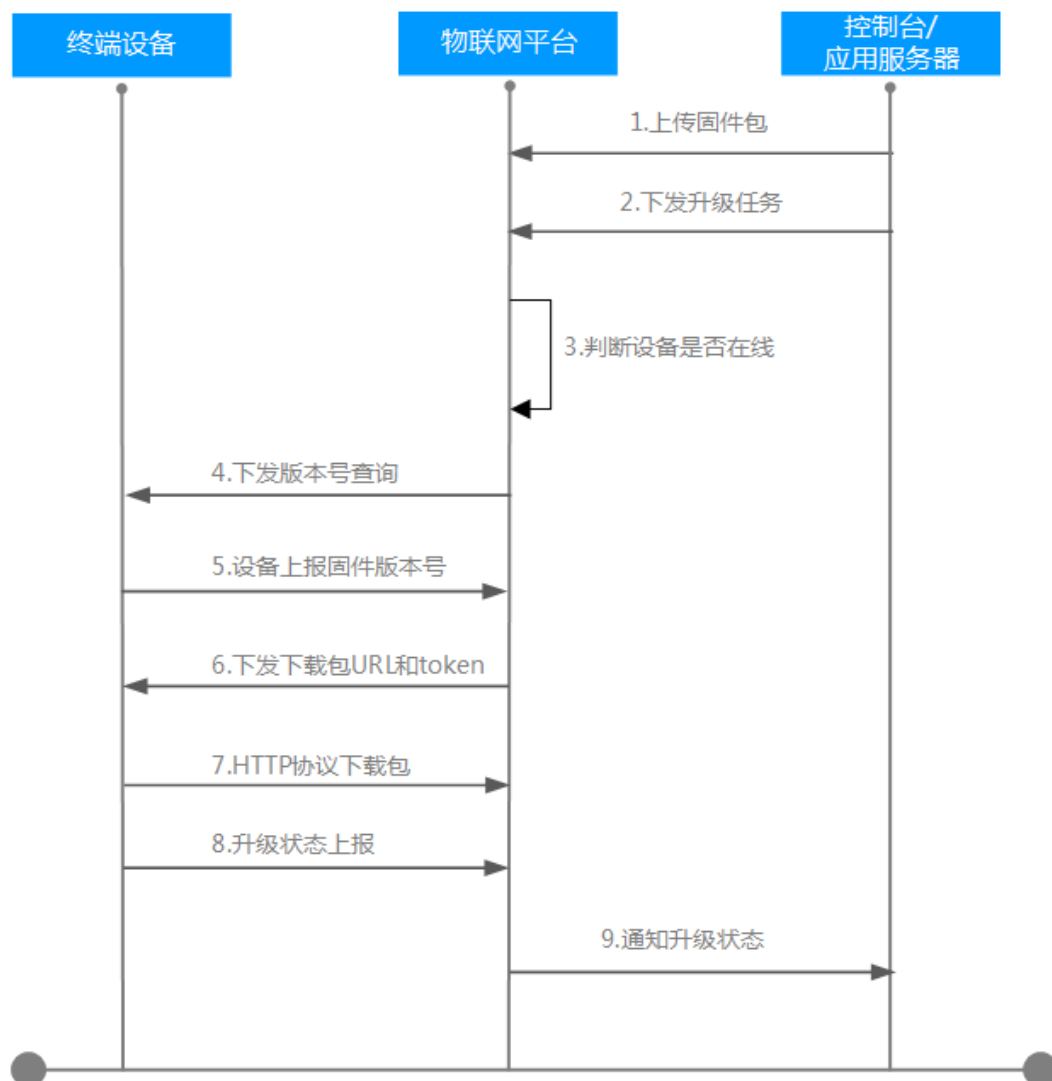
9. 物联网平台向设备订阅固件升级的状态。

10~11. 物联网平台向设备下发下载固件包的URL地址，通知设备下载固件包。终端设备根据该URL地址下载固件包，固件包的下载支持分片下载，下载完成后，设备知会物联网平台固件包已下载完毕。

12~13. 物联网平台向设备下发升级的命令，终端设备进行升级操作，升级完成后终端设备向物联网平台反馈升级结束。

14~16. 物联网平台下发命令查询固件升级的结果，获取升级结果后，向终端设备取消订阅升级状态通知，并向控制台应用服务器通知升级的结果。

MQTT 协议固件升级流程



MQTT协议FOTA升级流程的详细说明：

1~2. 用户在设备管理服务的控制台上传软件包，并在控制台或者应用服务器上创建软件升级任务。

3. 平台感知设备是否在线，当设备在线时立即触发升级协商流程。当设备不在线时，等待设备上线订阅升级topic，平台感知设备上线，触发升级协商流程。

4~5. 平台向设备下发查询设备固件版本号的命令，查询成功后，物联网平台根据升级的目标版本判断设备是否需要升级。如果返回的固件版本信息与升级的目标版本信息相同，则升级流程结束，不做升级处理，升级任务置为成功。如果返回的固件版本信息与升级的目标版本信息不同，且该版本号支持升级，则继续进行下一步的升级处理。

6~7. 物联网平台下发下载包URL、token及包的相关信息，用户根据下载包URL和token通过HTTP协议来下载软件包，24小时后token无效。

8. 终端设备进行下载包升级操作，升级完成后终端设备向物联网平台反馈升级的结果。


9. 物联网平台向控制台/应用服务器通知升级的结果。

检查固件升级能力

对终端设备进行固件升级，需要检查设备对应的产品模型中，是否支持设备的固件升级能力。您可以通过如下方式检查和设置设备固件升级的能力。

说明

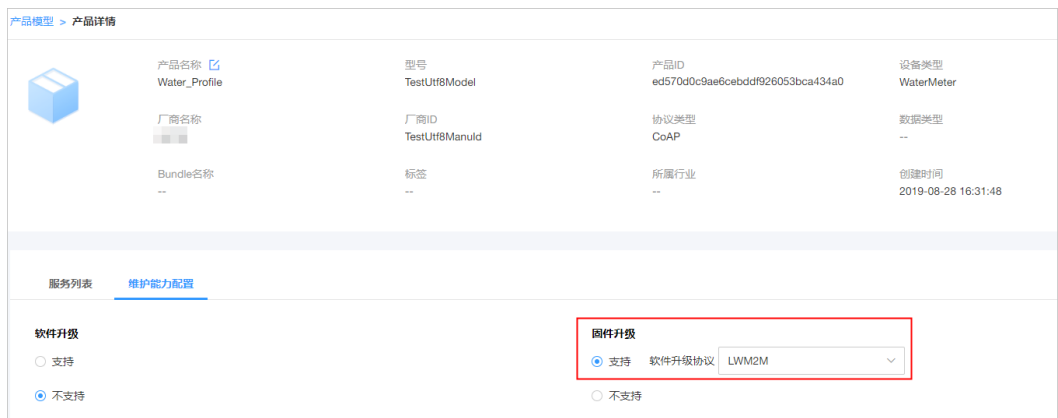
该操作只是修改了产品模型中定义的固件升级能力，该产品模型对应的终端设备也需要支持固件升级的能力。否则，当升级任务中选择了不支持固件升级的终端设备，会导致固件升级失败。

步骤1 登录**物联网平台控制台**，点击右上角“进入设备管理服务”。单击左下角，切换到设备管理页面。

步骤2 在“产品模型”界面点击“详情”，进入“产品详情”界面。



步骤3 在“维护能力配置”界面中，检查固件升级的能力。协议类型为LWM2M的产品模型，将固件升级设置为“支持”，协议为LWM2M。协议类型为MQTT的产品模型，将固件升级设置为“支持”，不校验固件升级协议类型。



----结束

上传固件包

设备管理服务的控制台，提供固件包的管理能力，您可以将需要升级的固件包统一上传到“固件包管理”界面上进行管理。在上传固件包到控制台前，需要对固件包进行签名。

说明

固件升级包的格式为.bin格式，不支持其它格式。上传的固件包需要使用压缩工具打包为.zip包。

固件包签名：

- 步骤1** 登录**物联网平台控制台**，点击右上角“进入设备管理服务”。在“工具”页面，下载“离线签名工具”。
- 步骤2** 解压“signtool.zip”，双击“signtool.exe”，运行离线签名工具。
- 步骤3** 在“生成数字签名公私钥对”区域，选择“签名算法”，设置“私钥加密口令”，点击“生成公私密钥”，在弹出的窗口中选择需要保存的目录，点击“确定”，保存生成的公钥文件和私钥文件。



- 步骤4** 在“软件包数字签名”区域导入私钥文件并输入私钥加密口令（私钥加密口令为在**步骤3**中设置的“私钥加密口令”）后，点击“确定”。并选择需要签名的固件包（固件包格式只能为.zip格式），点击“进行数字签名”。

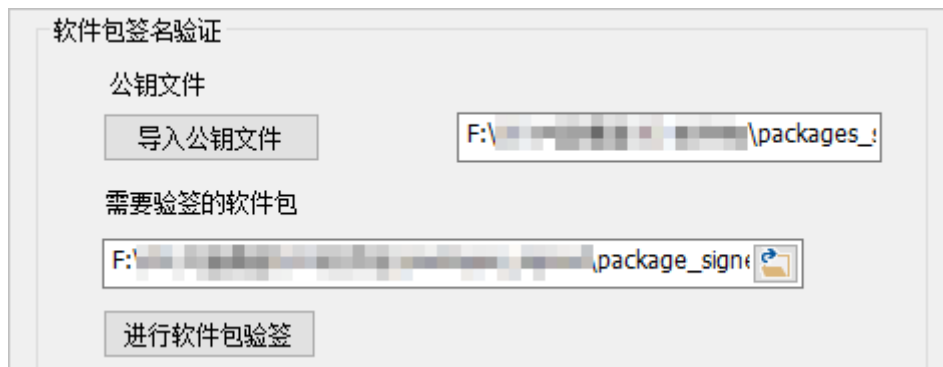
数字签名成功后，将会在原软件包所在目录下生成名称为“xxx_signed.zip”的带签名软件包。



- 步骤5** 验证固件包数字签名是否成功，在“软件包签名验证”区域导入公钥文件后，并选择**步骤4**中签名的固件包，点击“进行软件包验签”。
- 验证成功则弹出“验证签名成功！”提示框。
 - 验证失败则弹出“验签异常！”提示框。


📖 说明

在进行固件包验签时，带签名固件包的存放路径不能包含中文字符。



---结束

上传固件包：

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，在“软件库 > 公钥管理”中，上传固件包签名生成的“公钥文件（public.pem）”，上传公钥时，需选择厂商名称，该名称为上传的产品模型中定义的厂商名称。



步骤2 在“软件库 > 固件包管理”界面中，选择“添加固件包”，将已经签名的固件包进行上传。

上传固件包

请先完成固件包数字签名，并上传公钥文件，才能上传固件包。

*固件名称

*版本

*设备类型

*厂商名称

*型号

*协议

支持设备源版本

描述

在上传固件包时，需要填写如下信息：

- 固件名称：选择已经签名的固件包。
- 版本：固件包的版本，不能带中文字符，建议与固件包的名称保持一致。
- 设备类型：选择对应设备的[产品模型](#)。
- 厂商名称：需要与产品模型中定义的厂商名称保持一致。
- 型号：需要与产品模型中定义的型号保持一致。
- 协议：需要与产品模型中定义的协议保持一致。LWM2M协议的产品模型，协议填写LWM2M；MQTT协议的产品模型，协议填写MQTT。均需区分大小写。
- 支持设备源版本：手动输入，如需输入多个，可以在输入完一个版本后，点击“回车”按键，再输入下一个。

----结束

单个设备固件升级


用户对单个设备进行固件升级有两种方式：

- 应用服务器通过调用的“创建固件升级任务”API接口，创建单个设备的升级任务，详情请参考[创建固件升级任务](#)。
- 通过控制台，创建单个设备的固件升级任务。

📖 说明

- 设备升级前，请在“设备详情 > 命令”中，检查是否存在还在执行的命令，若存在，需要先等待所有命令执行后或是取消等待执行的命令，才可以创建设备升级任务。
- 在设备升级期间，应用侧不要向设备下发其他命令消息，否则可能会导致设备升级失败。

下面将重点介绍通过控制台创建单个设备的固件升级任务。

步骤1 登录**物联网平台控制台**，点击右上角“进入设备管理服务”。单击左下角，进入“设备 > 所有设备”页面。

步骤2 在设备列表中选择需要升级的设备，进入“设备详情 > 软件”页面，点击当前固件版本右侧的“升级”按钮。

📖 说明

需要在**检查固件升级能力**中，打开固件升级支持的开关，才会在“软件”界面显示固件信息和“升级”按钮。



步骤3 在弹出的界面中，选择需要升级的固件版本对应的固件包，单击“确定”，提示创建固件升级任务成功。通过该方式进行固件升级，如果升级失败，是不会进行重试的。



步骤4 在“设备详情 > 操作记录中”，可以查看固件升级任务的状态。操作类型需要选择“固件升级”。



固件升级任务的状态说明：

- 等待：等待任务的执行，创建完升级任务后，在10秒内为该状态，10秒后变更为“运行中”。
- 运行中：表示正在进行升级操作，包括等待设备上线、设备升级操作期间均为该状态。

- 停止：手动停止正在“运行中”的升级任务。
- 成功：终端设备反馈升级任务执行成功。
- 失败：设备升级失败，如下载升级包超时、升级响应超时、设备反馈升级结果失败等原因。

----结束

批量设备固件升级


用户对批量设备进行固件升级有两种方式：

1. 应用服务器通过调用的“创建固件升级任务”API接口，创建批量设备的升级任务，详情请参考[创建固件升级任务](#)。
2. 通过控制台，创建批量设备的固件升级任务。

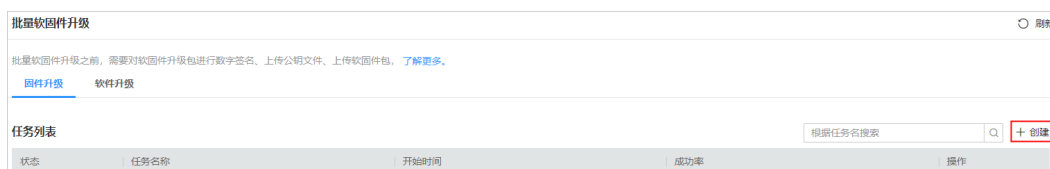
下面将重点介绍通过控制台创建单个设备的固件升级任务。

说明

- 设备升级前，请在“设备详情 > 命令”中，检查是否存在还在执行的命令，若存在，需要先等待所有命令执行后或是取消等待执行的命令，才可以创建设备升级任务。
- 在设备升级期间，应用侧不要向设备下发其他命令消息，否则可能会导致设备升级失败。

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，进入“批量任务 > 批量软固件升级”页面。

步骤2 在“固件升级”页面，点击“创建”按钮，创建批量固件升级任务。



步骤3 设置“任务信息”，填写任务名称和重试策略。

重试策略可以选择是否重试，如果选择了“自定义”，可以设置重试的次数。默认的重试时间为120秒，即设备升级失败后，隔120秒后会进行升级重试。

步骤4 选择需要升级的固件包。**步骤5** 选择需要升级的设备群组，然后点击“提交”。

设备群组可以参考**11 群组与标签（联通用户专用）**创建需要升级的设备群组，并绑定对应的设备。

**步骤6** 创建完批量升级任务后，可以在任务列表中查看批量任务的执行结果。点击任务名称，可以在“执行详情”界面查看每个设备的升级结果。**说明**

如果升级任务正在执行中，是不允许删除任务的，如需删除，请先在任务列表中，手动停止任务后，再删除升级任务。

状态	设备ID	描述	执行次数
运行中	4a7f31d5-b028-4f90-8fb1-7a5dd4458a4c	WaitingDeviceOnline	0
失败	e444e336-11c7-4833-9b18-a00431e4f1bb	Task Conflict	1

----结束**常见问题**

软/固件升级业务热点咨询问题如下，更多咨询问题请点击[查看更多](#)。

- [目标版本可以比当前版本低吗？](#)
- [软/固件包及其版本号如何获取？](#)
- [在软/固件升级任务中，业务处理是否会中断？](#)
- [为什么创建了软固件升级任务后，任务立即结束？](#)
- [常见的软/固件升级错误有哪些？](#)

相关 API 接口

- [查询版本包列表](#)
- [查询指定版本包](#)
- [删除指定版本包](#)
- [创建固件升级任务](#)
- [查询指定任务信息](#)
- [查询指定任务的子任务详情](#)
- [查询任务列表](#)

15 软件升级

- 15.1 升级指导（联通用户专用）
- 15.2 设备侧适配开发指导（联通用户专用）
- 15.3 PCP协议介绍（联通用户专用）

15.1 升级指导（联通用户专用）

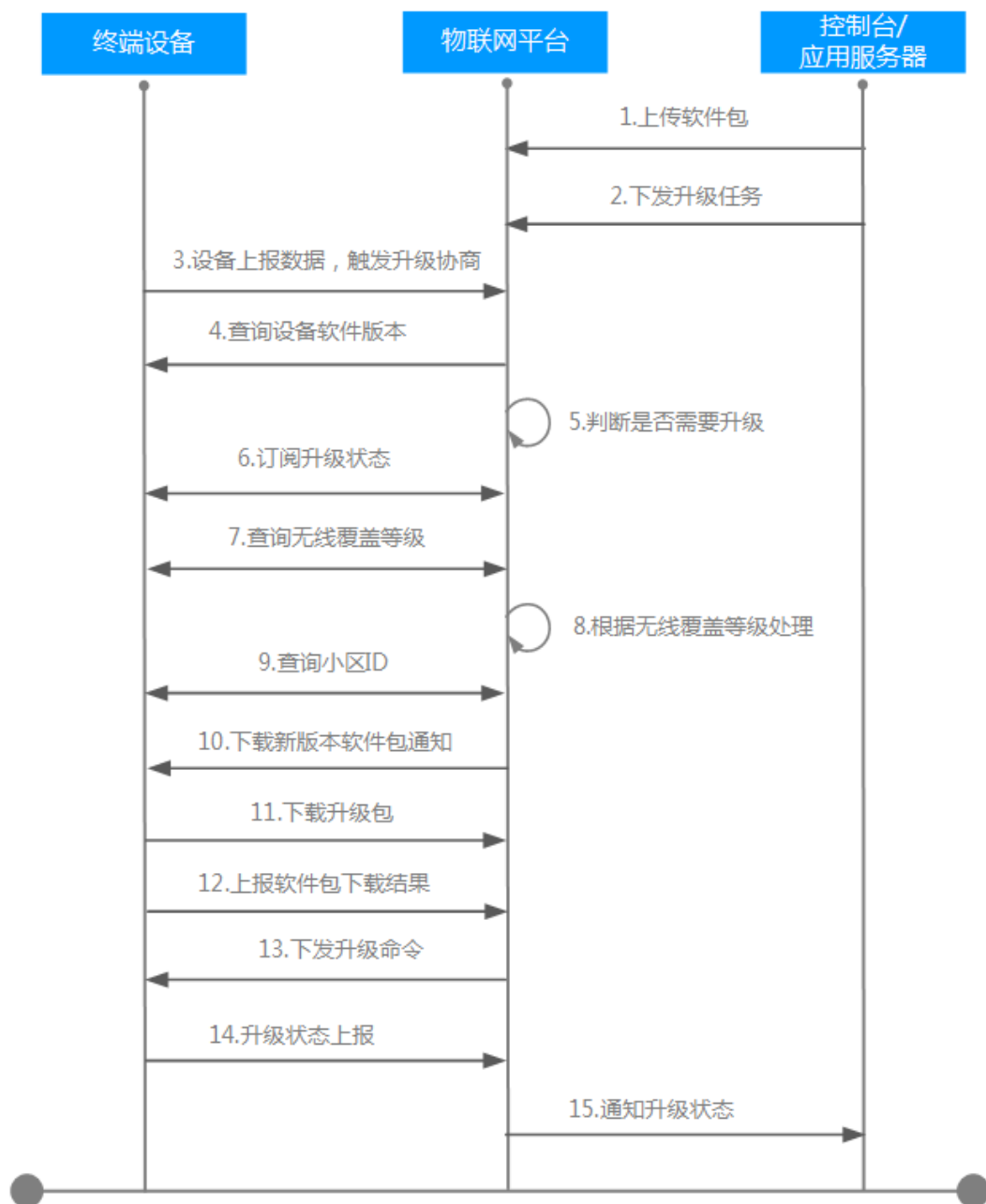
概述

软件（Software）一般分为系统软件和应用软件，系统软件实现设备最基本的功能，比如编译工具、系统文件管理等；应用软件可以根据设备的特点，提供不同的功能，比如采集数据、数据分析处理等。

软件升级又称为SOTA（SoftWare Over The Air），是指用户可以通过OTA的方式支持对LWM2M协议和MQTT协议的设备进行软件升级。基于LWM2M协议的产品模型，软件升级遵循的协议为PCP协议（[查看PCP协议介绍](#)），设备侧需要遵循PCP协议进行软件升级的适配开发，适配方法请参考[15.2 设备侧适配开发指导（联通用户专用）](#)；基于MQTT协议的产品模型，不校验软件升级协议类型。

物联网平台支持LWM2M协议和MQTT协议的产品模型的软件升级。

LWM2M 协议软件升级流程



LWM2M协议SOTA升级流程的详细说明：

1~2. 用户在设备管理服务的控制台上上传软件包，并在控制台或者应用服务器上创建软件升级任务。

3. NB-IoT设备上报数据，平台感知设备上线，触发升级协商流程。

4~5. 物联网平台向设备下发查询设备软件版本的命令，查询成功后，物联网平台根据升级的目标版本判断设备是否需要升级。

- 如果返回的软件版本信息与升级的目标版本信息相同，则升级流程结束，不做升级处理。
- 如果返回的软件版本信息与升级的目标版本信息不同，则继续进行下一步的升级处理。

6. 物联网平台向设备订阅软件升级的状态。

7~8. 物联网平台查询终端设备所在的无线信号覆盖情况，获取小区ID、RSRP（Reference Signal Received Power，参考信号接收功率）和SINR（Signal to Interference Plus Noise Ratio，信号干扰噪声比）信息。

- 查询成功：则根据如下方式计算可同时升级的并发数计算，并按照**步骤10**进行处理。
 - 如下图所示，如果设备的RSRP强度和SINR强度均落在等级“0”中，则同时可以对小区的50个相同信号覆盖区间的设备进行同时升级。
 - 如果设备的RSRP强度和SINR强度分别落在等级“0”和“1”中，则以信号较弱的等级“1”为准，则只能同时对小区的10个设备进行升级。
 - 如果设备的RSRP强度和SINR强度分别落在等级“1”和“2”中，则以信号较弱的等级“2”为准，则只能同时对小区的1个设备进行升级。
 - 如果设备的RSRP强度和SINR强度不在该3个等级范围内，且均可以查询到，则按照信号最弱覆盖等级“2”处理，则只能同时对1个设备进行升级。

无线覆盖等级	RSRP门限范围 (dBm)	SINR门限范围 (dB)
0	$-105 \leq \text{RSRP}$	$7 \leq \text{SINR}$
1	$-115 \leq \text{RSRP} < -105$	$-3 \leq \text{SINR} < 7$
2	$-125 \leq \text{RSRP} < -115$	$-8 \leq \text{SINR} < -3$

说明

如果用户在软件升级中发现同时进行升级的设备数较少，则可以联系当地运营商检查和优化设备所在小区的无线覆盖情况。

- 查询失败：则按照**流程9**进行处理。

9. 物联网平台继续下发查询小区ID信息的命令，获取终端设备所在的小区ID信息。

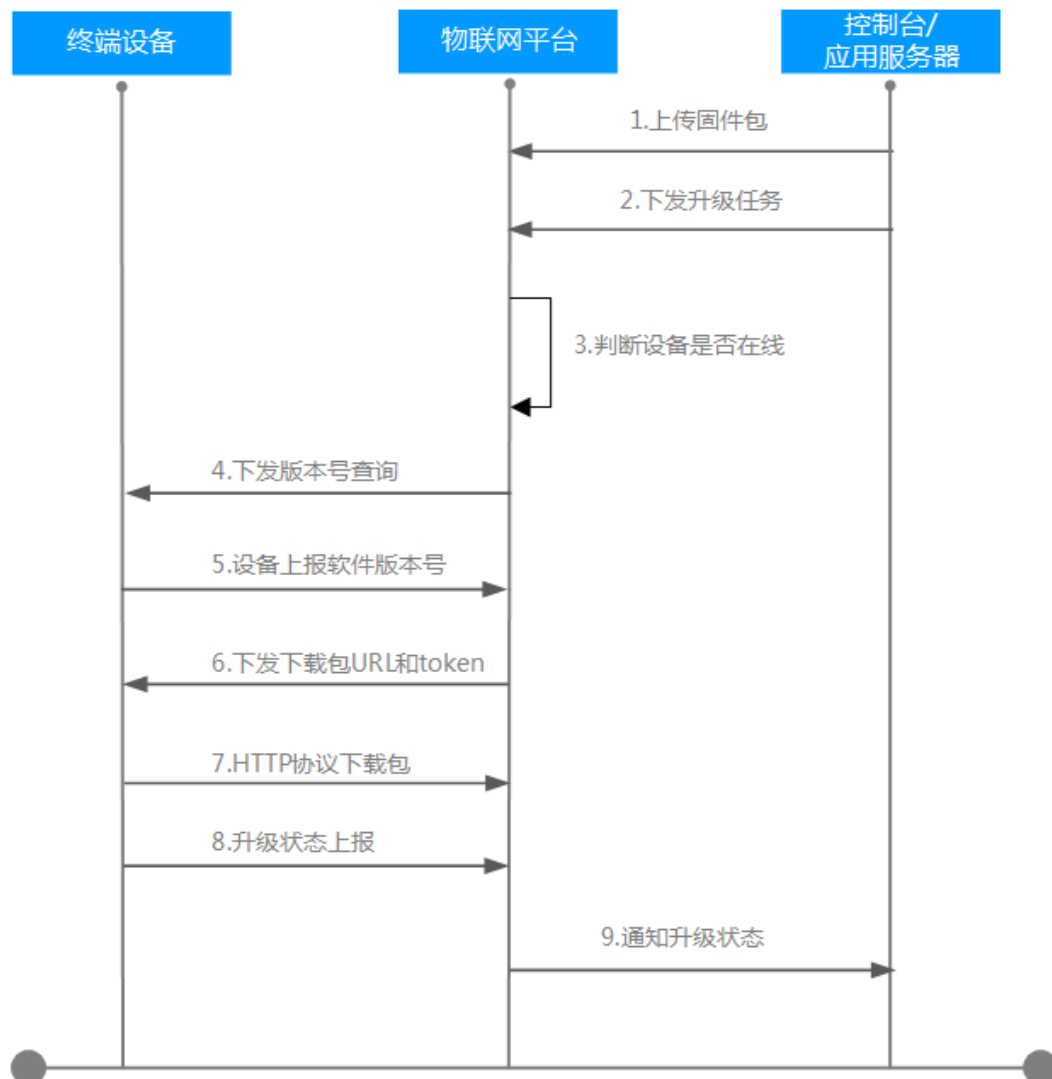
- 如果查询成功：物联网平台支持同时对该小区的10个相同情况的设备进行软件升级。
- 如果查询失败：则升级失败。

10~12. 物联网平台通知设备有新的软件包版本，设备启动软件包的下载。软件包的下载按照分片的方式进行下载，支持断点续传功能，通过软件包分片中携带的“versionCheckCode”确定是否属于同一个软件包。下载完成后，设备知会物联网平台软件包已下载完毕。

13~14. 物联网平台向设备下发升级的命令，终端设备进行升级操作，升级完成后终端设备向物联网平台反馈升级的结果。

15. 物联网平台向控制台/应用服务器通知升级的结果。

MQTT 协议软件升级流程



MQTT协议SOTA升级流程的详细说明：

1~2. 用户在设备管理服务的控制台上传软件包，并在控制台或者应用服务器上创建软件升级任务。

3. 平台感知设备是否在线，当设备在线时立即触发升级协商流程。当设备不在线时，等待设备上线订阅升级topic，平台感知设备上线，触发升级协商流程。

4~5. 平台向设备下发查询设备软件版本号的命令，查询成功后，物联网平台根据升级的目标版本判断设备是否需要升级。如果返回的软件版本信息与升级的目标版本信息相同，则升级流程结束，不做升级处理，升级任务置为成功。如果返回的软件版本信息与升级的目标版本信息不同，且该版本号支持升级，则继续进行下一步的升级处理。


6~7. 物联网平台下发下载包URL、token及包的相关信息，用户根据下载包URL和token通过HTTP协议来下载软件包，24小时后token无效。

8. 终端设备进行下载包升级操作，升级完成后终端设备向物联网平台反馈升级的结果。

9. 物联网平台向控制台/应用服务器通知升级的结果。

检查软件升级能力

对终端设备进行软件升级，需要检查设备对应的产品模型中，是否支持设备的软件升级能力。您可以通过如下方式检查和设置设备软件升级的能力

步骤1 登录设备管理服务的控制台，单击左下角 ，切换到设备管理页面。

步骤2 在“产品模型”界面点击“详情”，进入“产品详情”界面。



步骤3 在“维护能力配置”界面中，检查软件升级的能力。协议为LWM2M的产品模型，将软件升级设置为“支持”，协议为PCP。协议为MQTT的产品模型，将软件升级设置为“支持”，不校验软件升级协议类型。

说明

该操作只是修改了产品模型的能力，该产品模型对应的终端设备也需要支持软件升级的能力。否则，当升级任务中选择了不支持软件升级的终端设备，会导致软件升级失败。



----结束

制作软件升级版本包

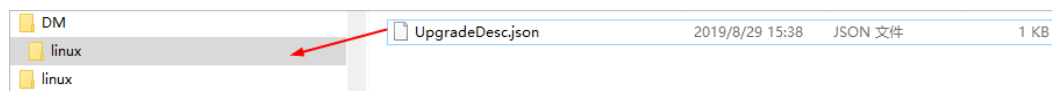
设备升级的软件包文件由各设备厂商提供，在物联网平台上传设备的软件升级包前，需要制作软件升级的版本包，用于修改软件包的描述文件，如软件版本、厂商名称、设备类型、产品模型等信息。下面将详细介绍版本包的制作方法。

步骤1 新建文件夹命名为“DM”，在DM文件夹下新建文件夹，命名为“linux”。

步骤2 使用Notepad++文本工具新建一个文本文件，拷贝如下内容到文本中，在Notepad++工具的“编码”菜单中选择“使用UTF-8编码”，然后将文本进行存储，存储路径选择步骤1中的“linux”文件夹，文件名称命名为“UpgradeDesc”，保存类型选择“json”。

```
{  
  "specVersion": "",
```

```
"fileName": "",
"packageType": "",
"version": "",
"deviceType": "",
"manufacturerName": "",
"model": "",
"protocolType": "",
"description": "",
"versionCheckCode": "",
"deviceShard": "",
"platform": "",
"supportSourceVersionList": [],
"date": ""
}
```

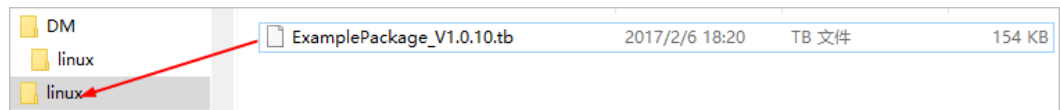


步骤3 打开创建的“UpgradeDesc.json”文件，修改软件升级描述文件，相关字段如下表所示。

字段名	字段描述	是否必填
specVersion	描述文件版本号，固定值：“1.0”。	是
fileName	软件包文件名，例如：“ExamplePackage_V1.0.10.xx”。	是
packageType	软件包类型，必须设置为：“softwarePackage”。	是
version	软件包版本号长度不超过16个字节，例如：“V1.0.10”	是
deviceType	设备类型，需要与产品模型保持一致。例如：“WaterMeter”。	是
manufacturerName	制造商名称，需要与产品模型保持一致。例如：“TestUtf8ManuName”。	是
model	产品型号，需要与产品模型保持一致。例如：“TestUtf8ModelM2M”。	是
protocolType	设备接入协议类型，需要与产品模型保持一致。例如：“CoAP”。 说明 设备接入的协议类型有三种：“CoAP”、“LWM2M”、“MQTT”。	是
description	对软件包的自定义描述。	是
versionCheckCode	软件升级包校验码，长度为4个字符。软件包下载时支持断点续传，根据该字段标识前后两次下载的软件包分片是否为同一个软件包。	否
deviceShard	终端下载软件包的每个分片的大小，单位为byte，如果不设置默认为500byte。大小为32~500之间。	否
platform	标识设备的操作系统，如linux。	否

字段名	字段描述	是否必填
supportSourceVersionList	支持用于升级此版本包的设备源版本列表。 支持通配符配置，*代表匹配任意0~n个字符，?代表匹配单个任意字符，如果存在多个版本，请使用英文“;”隔开。	否
date	出包时间，格式为：“yyyy-MM-dd”。	否

步骤4 在与“DM”同级目录下创建文件夹，命名为“linux”，该文件夹名称必须同**步骤1**中的文件夹命令保持一致，将厂商软件包（软件包格式无限制）置于该文件中。



步骤5 选中“DM”和“linux”文件夹，使用压缩工具打包成ZIP格式的压缩包，建议命令为“xx_package.zip”。

须知

- 文件“DM”和“linux”的命名是固定的。
- “xx_package.zip”下不能包含package这层目录。
- 仅支持ZIP格式的压缩包，不能压缩成其他格式后，例如rar，再手动修改文件类型为zip。

----结束

上传软件包

设备管理服务的控制台，提供软件包的管理能力，您可以将需要升级的软件包统一上传到“软件包管理”界面上进行管理。在上传软件包到控制台前，需要对软件包进行签名。

软件包签名：

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。在“工具”页面，下载“离线签名工具”。

步骤2 解压“signtool.zip”，双击“signtool.exe”，运行离线签名工具。

步骤3 在“生成数字签名公私钥对”区域，选择“签名算法”，设置“私钥加密口令”，点击“生成公私密钥”，在弹出的窗口中选择需要保存的目录，点击“确定”，保存生成的公钥文件和私钥文件。



步骤4 在“软件包数字签名”区域导入私钥文件并输入私钥加密口令（私钥加密口令为在步骤3中设置的“私钥加密口令”）后，点击“确定”。并选择需要签名的软件包（软件包格式只能为.zip格式），点击“进行数字签名”。

数字签名成功后，将会在原软件包所在目录下生成名称为“xxx_signed.zip”的带签名软件包。

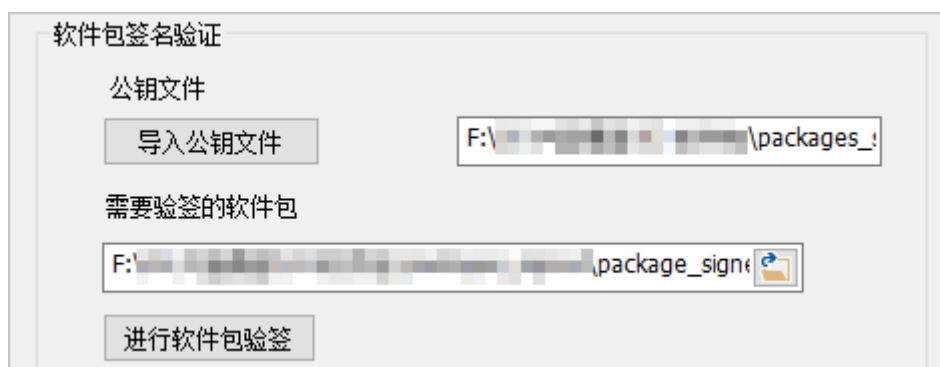


步骤5 验证软件包数字签名是否成功，在“软件包签名验证”区域导入公钥文件后，并选择步骤4中签名的软件包，点击“进行软件包验签”。

- 验证成功则弹出“验证签名成功！”提示框。
- 验证失败则弹出“验签异常！”提示框。


说明

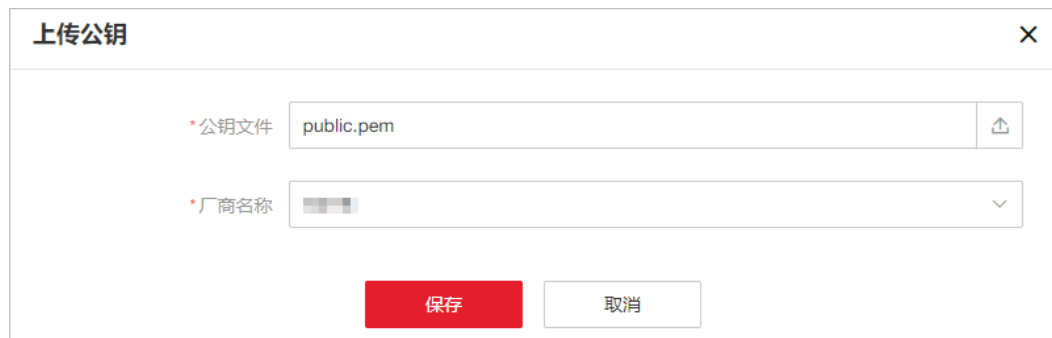
在进行软件包验签时，带签名软件包的存放路径不能包含中文字符。




----结束


上传软件包：

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，在“软件库 > 公钥管理”中，上传软件包签名生成的“公钥文件（public.pem）”，上传公钥时，需选择厂商名称，该名称为上传的产品模型中定义的厂商名称。



上传公钥

*公钥文件 

*厂商名称 

步骤2 在“软件库 > 软件包管理”界面中，选择“添加软件包”，将已经签名的软件包进行上传。

----结束

单个设备软件升级


用户对单个设备进行软件升级有两种方式：

1. 应用服务器通过调用的“创建软件升级任务”API接口，创建单个设备的升级任务，详情请参考[创建软件升级任务](#)。
2. 通过控制台，创建单个设备的软件升级任务。

说明

- 设备升级前，请在“设备详情 > 命令”中，检查是否存在还在执行的命令，若存在，需要先等待所有命令执行后或是取消等待执行的命令，才可以创建设备升级任务。
- 在设备软件升级期间，应用侧不要向设备下发其他命令消息，否则可能会导致设备升级失败。

下面将重点介绍通过控制台创建单个设备的软件升级任务。

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，进入“设备 > 所有设备”页面。

步骤2 在设备列表中选择需要升级的设备，进入“设备详情 > 软件”页面，点击当前软件版本右侧的“升级”按钮。

说明

需要在[检查软件升级能力](#)中，打开软件升级支持的开关，才会在“软件”界面显示软件信息和“升级”按钮。



步骤3 在弹出的界面中，选择需要升级的软件版本对应的软件包，单击“确定”，提示创建软件升级任务成功。通过该方式进行软件升级，如果升级失败，是不会进行重试的。



步骤4 在“设备详情 > 操作记录中”，可以查看软件升级任务的状态。操作类型需要选择“软件升级”。



软件升级任务的状态说明：

- 等待：等待任务的执行，创建完升级任务后，在10秒内为该状态，10秒后变更为“运行中”。
- 运行中：表示正在进行升级操作，包括等待设备上线、设备升级操作期间均为该状态。
- 停止：手动停止正在“运行中”的升级任务。
- 成功：终端设备反馈升级任务执行成功。
- 失败：设备升级失败，如下载升级包超时、升级响应超时、设备反馈升级结果失败等原因。

----结束

批量设备软件升级


用户对批量设备进行软件升级有两种方式：

1. 应用服务器通过调用的“创建软件升级任务”API接口，创建批量设备的升级任务，详情请参考[创建软件升级任务](#)。
2. 通过控制台，创建批量设备的软件升级任务。

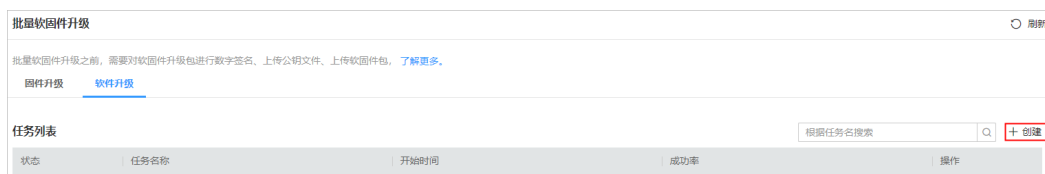
下面将重点介绍通过控制台创建单个设备的软件升级任务。

📖 说明

- 设备升级前，请在“设备详情 > 命令”中，检查是否存在还在执行的命令，若存在，需要先等待所有命令执行后或是取消等待执行的命令，才可以创建设备升级任务。
- 在设备升级期间，应用侧不要向设备下发其他命令消息，否则可能会导致设备升级失败。

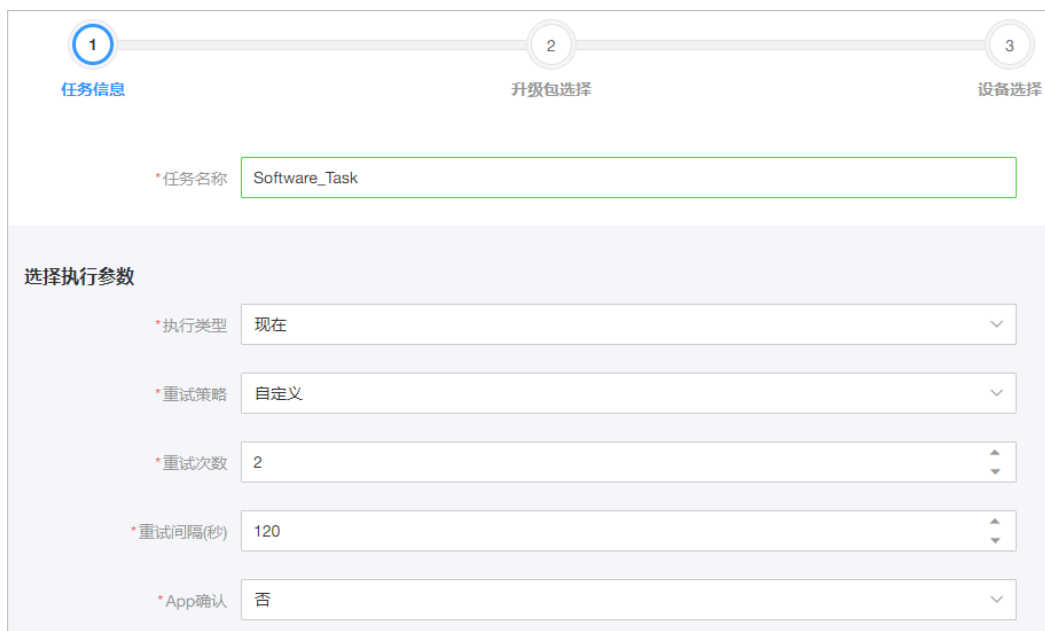
步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，进入“批量任务 > 批量软固件升级”页面。

步骤2 在“软件升级”页面，点击“创建”按钮，创建批量软件升级任务。



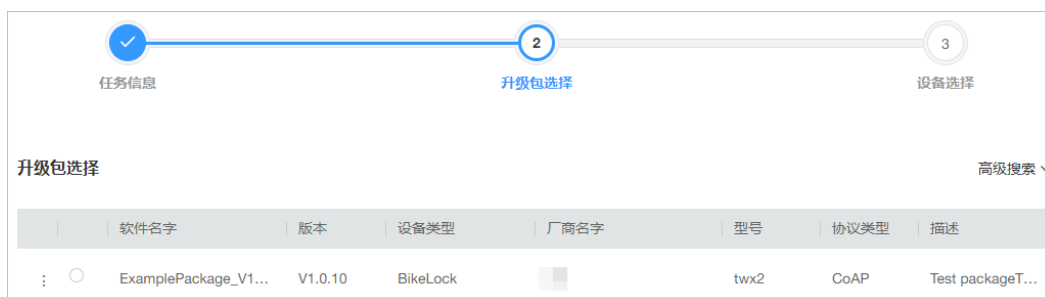
步骤3 设置“任务信息”，填写执行类型、任务名称、重试策略、App确认等信息。

- 重试策略：如果选择了“自定义”，可以设置重试的次数和重试间隔，重试间隔表示升级的设备在升级失败后，隔多长时间进行重试。
- APP确认：是否需要应用服务器确认，默认为否。

该截图展示了任务配置的第一步：“任务信息”。顶部有一个进度条，包含三个步骤：1. 任务信息（当前选中）、2. 升级包选择、3. 设备选择。配置项如下：

- *任务名称：Software_Task
- *执行类型：现在
- *重试策略：自定义
- *重试次数：2
- *重试间隔(秒)：120
- *App确认：否

步骤4 选择需要升级的软件包。

该截图展示了任务配置的第二步：“升级包选择”。顶部进度条显示第二步“升级包选择”为当前步骤。下方有一个“高级搜索”按钮。主要部分是一个表格，列出了可供选择的软件包：

软件名字	版本	设备类型	厂商名字	型号	协议类型	描述
ExamplePackage_V1...	V1.0.10	BikeLock		twx2	CoAP	Test packageT...

步骤5 选择需要升级的设备或者设备群组，然后点击“提交”。

设备群组可以参考[11 群组与标签（联通用户专用）](#)创建需要升级的设备群组，并绑定对应的设备。



步骤6 创建完批量升级任务后，可以在任务列表中查看批量任务的执行结果。点击任务名称，可以在“执行详情”界面查看每个设备的升级结果。

说明

如果升级任务正在执行中，是不允许删除任务的，如需删除，请先在任务列表中，手动停止任务后，再删除升级任务。



----结束

常见问题

软/固件升级业务热点咨询问题如下，更多咨询问题请点击[查看更多](#)。

- [目标版本可以比当前版本低吗？](#)
- [软/固件包及其版本号如何获取？](#)
- [在软/固件升级任务中，业务处理是否会中断？](#)
- [为什么创建了软固件升级任务后，任务立即结束？](#)
- [常见的软/固件升级错误有哪些？](#)

相关 API 接口

- [查询版本包列表](#)
- [查询指定版本包](#)
- [删除指定版本包](#)
- [创建软件升级任务](#)
- [查询指定任务信息](#)
- [查询指定任务的子任务详情](#)
- [查询任务列表](#)

15.2 设备侧适配开发指导（联通用户专用）

概述

设备的OTA软件升级是基于华为定义的PCP协议进行的，设备侧需根据PCP协议定义的交互流程进行适配开发。下面我们将结合物联网平台与设备的软件升级交互流程，介绍终端设备将如何基于PCP协议构建交互过程中的请求消息和应答消息，帮助您更好的根据PCP协议进行终端侧的软件升级功能开发。

下面我们先了解下PCP消息的结构，PCP协议的请求消息和应答消息都遵循相同的消息结构，主要由这几部分组成：



PCP协议消息由：起始标识位、版本号、消息码、校验码、数据区长度和数据区组成，各字段的要求和描述如下表所示。

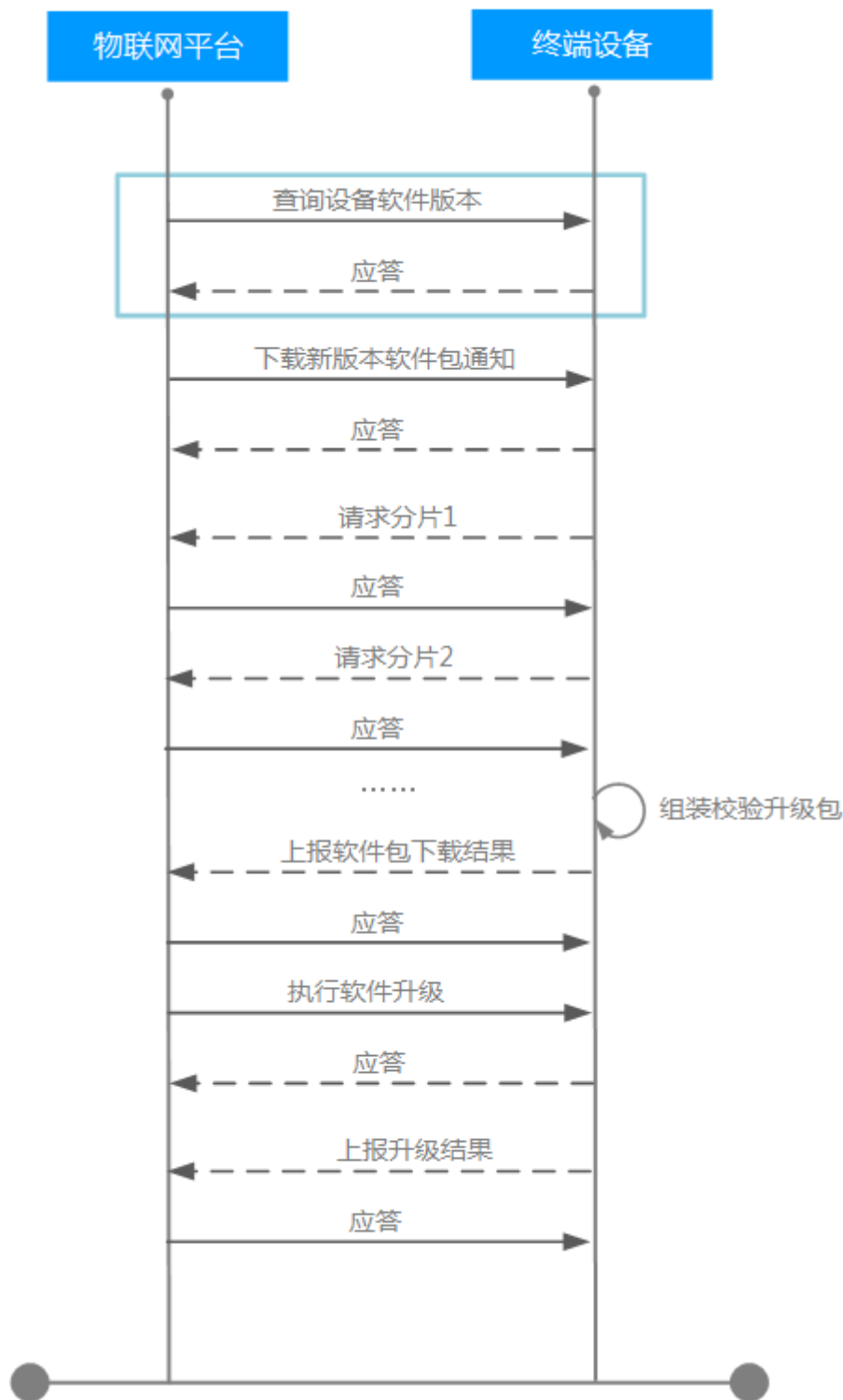
字段名	字段类型	描述和要求
起始标识	WORD	起始标识，固定为0XFFFE。
版本号	BYTE	高四位预留；低四位表示协议版本号，当前为1。
消息码	BYTE	标识物联网平台与设备之间的请求消息类型，应答消息的消息码和请求消息相同。消息码的定义为： <ul style="list-style-type: none">● 0-18：预留消息码，暂未使用。● 19：查询设备版本。● 20：下载新版本软件包通知。● 21：请求下载升级包。● 22：上报升级包下载结果。● 23：执行软件升级。● 24：上报升级结果。● 25-127：预留消息码，暂未使用。
校验码	WORD	从起始标识到数据区的最后一个字节的CRC16校验值，计算前先把校验码字段置为0，计算完成后把结果写到校验码字段。

字段名	字段类型	描述和要求
数据区长度	WORD	数据区的长度。
数据区	BYTE[n]	可变长度，具体由各个指令定义，可参考下面介绍的各个指令对应的请求消息和应答消息定义。

数据类型	描述
BYTE	无符号一字节整数
WORD	无符号二字节整数
DWORD	无符号四字节整数
BYTE[n]	n字节的十六进制数
STRING	字符串

查询设备版本号

从设备的软件升级流程（本流程只描述物联网平台与设备基于PCP协议交互的流程，完整的软件升级流程请参考[LWM2M协议软件升级流程](#)）可以看到，首先物联网平台向设备下发查询版本号信息，设备进行应答。



物联网平台发送消息

根据**PCP消息结构的定义**可以得出，物联网平台向设备下发查询版本号时，各消息字段的填写如下：

- 起始标识：固定为消息流的前2个字节，固定为FFFE。
- 版本号：数据类型为1个字节整数，且固定为1，即在消息流中为01。

- 消息码：数据类型为1个字节整数，查询设备版本的消息码为19，转换为十六进制为13。
- 校验码：数据类型为2个字节整数，先将校验码置为0000，然后将完整的消息码流进行CRC16的算法计算得到校验码，再将得到的校验码替换原消息中的0000。
- 数据区长度：数据类型为2个字节整数，代表数据区的消息长度，根据数据区的数据结构可以得出该条消息无数据区，即数据区长度为0000。
- 数据区：数据区代表要真正发送给设备的数据，根据查询版本信息的数据区定义，该条消息是没有实际要传送的数据的，即无需数据区字段。

字段	数据类型	描述及要求
无数据区		

因此将查询版本消息的码流组合起来得到：FFFE 01 13 0000 0000。前面的校验码时讲了，需要将组合后的消息码流进行CRC16算法（物联网平台提供了基于JAVA和C语言的[CRC16算法代码样例](#)，您可以直接使用）得到校验码4C9A，然后将该校验码替换原码流中的0000后得到FFFE01134C9A0000，该消息码流即为物联网平台发送给设备的查询版本信息的消息码流。

设备返回的应答消息

设备收到物联网平台要查询设备的软件版本号消息，设备要向物联网平台反馈查询的结果，各消息字段的填写如下。

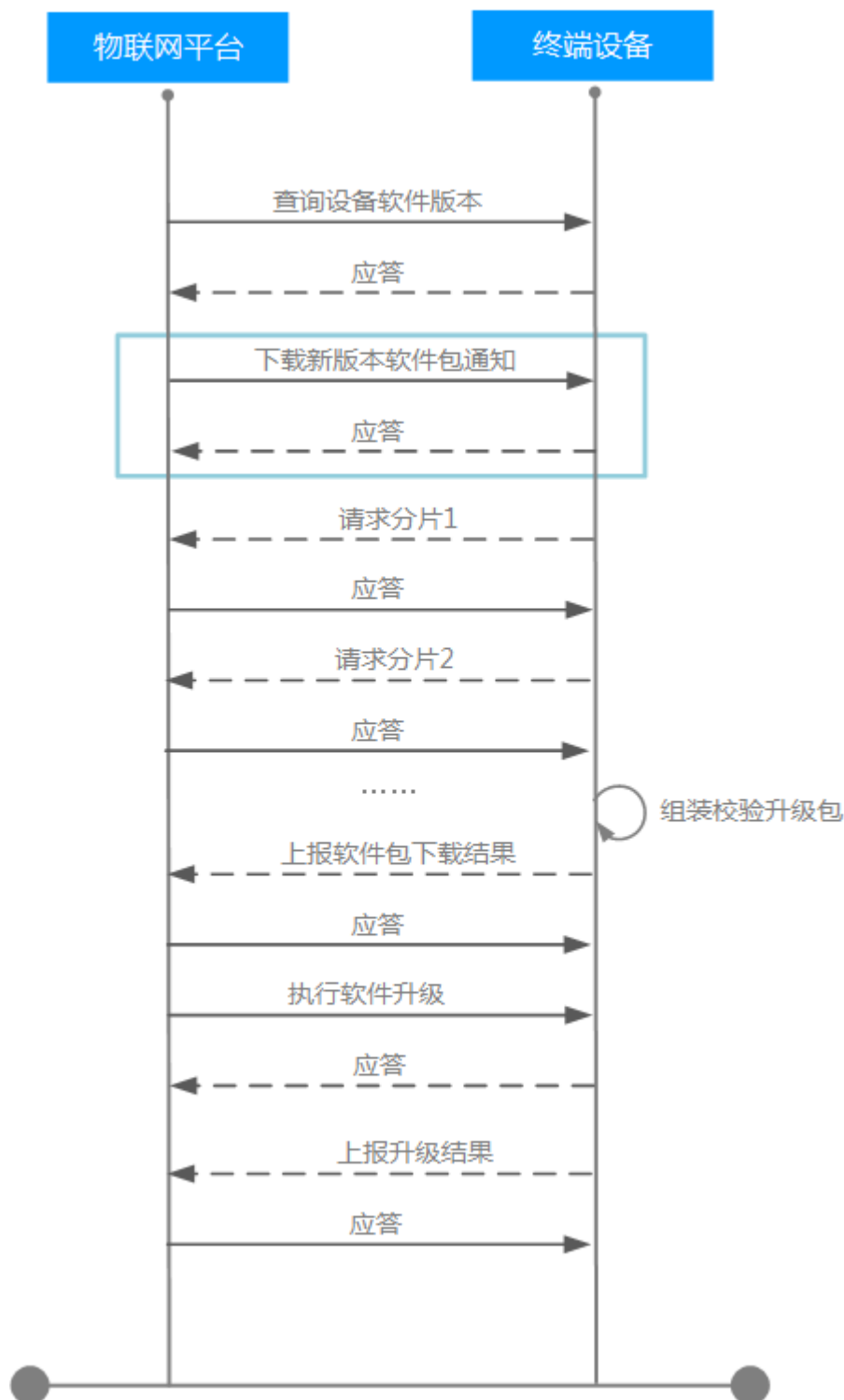
- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：与请求的消息码一致，为13。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段的数据类型得出数据区长度为17个字节，转换为十六进制为：0011。
- 数据区：根据数据区的定义可知，处理成功的结果码为00，版本号信息假设为V0.9，将V0.9进行ASCII转码得到56302E39，由于版本号的数据类型为BYTE[16]，即16个字节，当前只有4个字节，因此需要在版本号数据后面补0，得到56302E39000000000000000000000000。因此，数据区合并后为0056302E390000000000000000000000。

字段	数据类型	描述及要求
结果码	BYTE	“0X00” 处理成功
当前版本号	BYTE[16]	当前版本号，由ASCII字符组成，位数不足时，后补“0X00”。

将查询版本信息的消息流组合起来得到：FFFE 01 13 0000 0011 0056302E390000000000000000000000000000。前面讲到，还要将消息流进行CRC16算法计算得到校验码为8DE3。因此，物联网平台向设备查询版本号信息，设备向平台返回的消息流为FFFE01138DE300110056302E390000000000000000000000000000。

下载新版本软件包通知

根据PCP协议约定的交互流程，查询完版本号后，物联网平台下发指令让设备下载新版本的软件包。



物联网平台发送消息

根据PCP消息结构的定义可以得出，物联网平台向设备下发下载新版本软件包通知时，各消息字段的填写如下。

- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：此处为新版本通知，查询消息码表可以知道新版本通知为20，转换为十六进制为14。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的消息字段可以得出，数据区长度为22个字节，转换为十六进制为：0016。
- 数据区：根据数据区的定义可知。
 - 目标版本号：由16个字节组成，假设升级的目标版本号为v1.0版本，转换为十六进制并在后面14个字节补充0后得到：
56312E30000000000000000000000000。
 - 升级包分片大小：由2个字节组成，升级包分片大小由制作软件升级版本包中的“deviceShard”字段定义，假设为500byte，转换为十六进制后为：
01F4。
 - 升级包分片总数：由2个字节组成，由软件包大小除以每个分片的大小并向上取整获得。假设软件包大小为500byte，则分片数量为1，转换为十六进制后为：0001。
 - 检验码：由2个字节组成，由制作软件升级版本包中的“versionCheckCode”字段定义，假设为1234。

字段	数据类型	描述及要求
目的版本号	BYTE[16]	目的版本号，由ASCII字符组成，位数不足时，后补“0X00”。
升级包分片大小	WORD	每个分片的大小
升级包分片总数	WORD	升级包分片总数
升级包校验码	WORD	升级包校验码。用户上传升级包时，需要在升级包描述文件里填写校验码

将下载新版本软件包通知的消息流组合起来得到：FFFE 01 14 0000 0016 56312E30000000000000000000000000 01F4 0001 0000。前面说了，还要将消息流进行CRC16算法计算得到校验码为02F7。因此，物联网平台向设备通知下载新版本软件包的信息，物联网平台向设备发送的消息流为
FFFE011402F7001656312E30000000000000000000000000000001F400011234。

设备返回的应答消息

设备收到下载新版本软件包通知后，设备向物联网平台返回应答消息，是否允许设备进行升级，各消息字段的填写如下。

- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：与请求的消息码一致，为14。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段的数据类型得出数据区长度为1个字节，转换为十六进制为：0001。

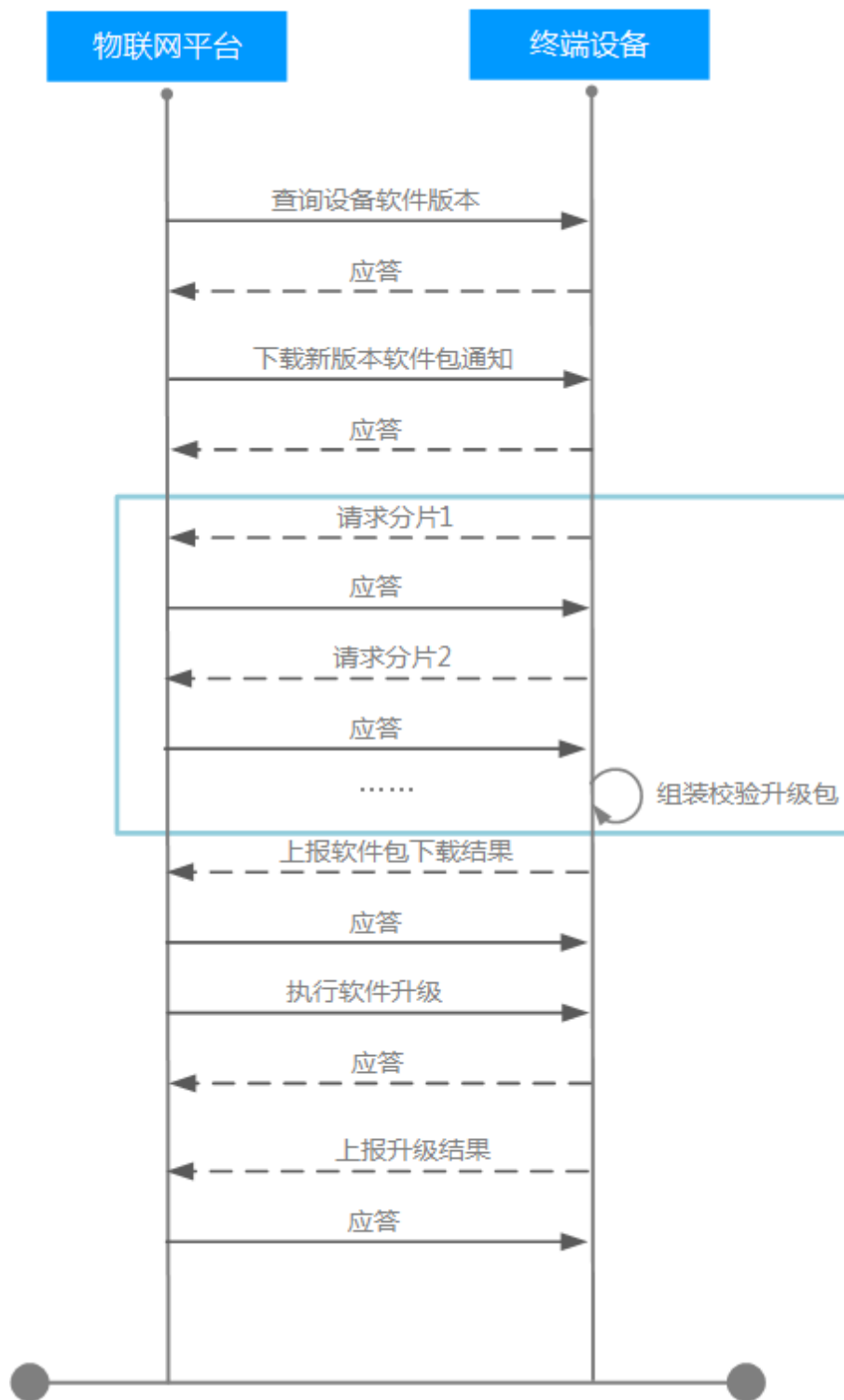
- 数据区：设备根据自身的情况对平台下发的新版本通知进行响应，本示例以设备应答“允许升级”为例进行介绍，得出数据区为：00。其它应答消息请根据应答消息字段进行适配。

字段	数据类型	描述及要求
结果码	BYTE	“0X00” 允许升级 “0X01” 设备使用中 “0X02” 信号质量差 “0X03” 已经是最新版本 “0X04” 电量不足 “0X05” 剩余空间不足 “0X09” 内存不足 “0X7F” 内部异常

将设备给物联网平台的应答消息流组合起来得到：FFFE 01 14 0000 0001 00。前面说了，还要将消息流进行CRC16算法计算得到校验码为D768。因此，设备向平台返回的应答消息流为FFFE0114D768000100。

下载升级包

根据PCP协议约定的交互流程，物联网平台通知设备有新的软件版本时，设备向物联网平台请求下载软件包，按照分片的序号进行下载。



设备发送的请求消息

根据PCP消息结构的定义可以得出，设备向物联网平台发送的请求软件包分片的第一条消息，各消息字段的填写如下。

- 起始标识固定为：FFFE。
- 版本号固定为：01。

- 消息码：查询消息码表可知请求升级包的消息码为21，转换为十六进制为：15。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段的数据类型得出数据区长度为18个字节，转换为十六进制为：0012。
- 数据区：目标版本号为平台下发的新版本通知版本号，即v1.0，转换为十六进制为56312E30000000000000000000000000，分片序号为第0个分片，即0000。

字段	数据类型	描述及要求
目的版本号	BYTE[16]	目的版本号，由ASCII字符组成，位数不足时，后补“0X00”。
分片序号	WORD	表示请求获取的分片序号，从0开始计算，分片的总数为软件包大小除以每个分片的大小并向上取整获得。设备可以保存已经收到的分片，下次直接从缺失的分片开始请求，达到断点续传的效果。

设备向物联网平台发送请求软件包分片的第一条消息为：FFFE 01 15 0000 0012 56312E30000000000000000000000000 0000（CRC16校验前），经CRC16计算得到校验码为：5618。则替换校验码后设备发送的第一条请求分片消息为：FFFE01155618001256312E3000。

其它分片请求的消息流只需要替换分片序号后，重新计算并替换CRC16校验码即可，此处就不再展开。

物联网平台的应答消息

物联网平台收到设备的请求软件包分片消息后，将会给设备下发分片的数据。物联网平台向设备响应的第一条请求分片的消息，各消息字段的填写如下。

- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：与请求的消息码一致：15。
- 校验码：CRC16计算前先用0000替代。
- 数据区：先讲数据区再讲数据区长度。结果码：00，分片序号：第0个分片：0000，分片数据：跟软件包定义的内容有关，我们假设软件包内容为HELLO, IoT SOTA!，经ASCII码转换为十六进制为：48454C4C4F2C20496F5420534F544121，共16字节。由于制作软件升级版本包中的“deviceShard”字段定义分片大小为500byte，即最大长度为500字节。这种情况下，无需在数据的后面补充0。

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X80升级任务不存在。 0X81指定的分片不存在。
分片序号	WORD	表示返回的分片序号。
分片数据	BYTE[n]	分片的内容，n为实际的分片大小。如果结果码不为0，则不带此字段。

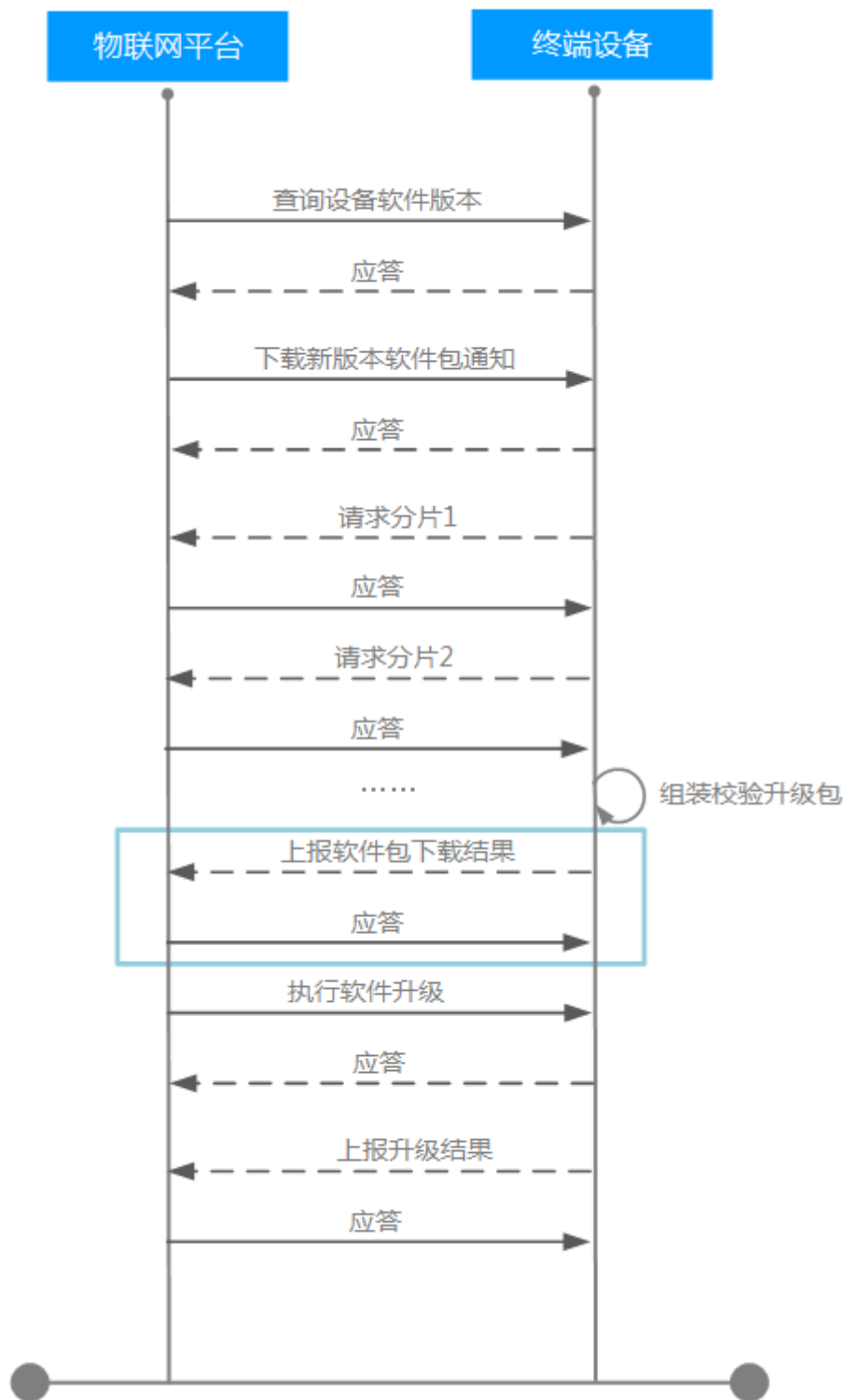
- 数据区长度：根据数据区的字段定义得出该数据长度为19，转换为十六进制为：0013。

物联网平台向设备发送的第一个软件包分片消息为：FFFE 01 15 0000 0013 00 0000 48454C4C4F2C20496F5420534F544121（CRC16校验前），经CRC16计算得到校验码为：E107。则替换校验码后物联网平台向设备发送的第一个软件包分片消息为：FFFE0115E107001300000048454C4C4F2C20496F5420534F544121。

其它软件包分片的消息流只需要替换分片序号和分片数据后，重新计算并替换CRC16校验码即可，此处就不再展开。

上报下载结果

根据PCP协议约定的交互流程，设备接收完所有分片数据并组装完软件包后，需要向物联网平台上报软件包的下载结果。



设备发送的请求消息

根据PCP消息结构的定义可以得出，设备向物联网平台发送的上报软件包下载结果消息，各个消息字段的填写如下：

- 起始标识固定为：FFFE。
- 版本号固定为：01。

- 消息码：与请求的消息码一致，为16。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段的数据类型得出数据区长度为1个字节，转换为十六进制为：0001。
- 数据区：上报软件包的下载结果，比如下载成功，设备侧上报00。

字段	数据类型	描述及要求
下载状态	BYTE	0X00下载成功。 0X05剩余空间不足。 0X06下载超时。 0X07升级包校验失败。 0X08升级包类型不支持。

设备向物联网平台发送升级包下载结果的消息为：FFFE 01 16 0000 0001 00（CRC16校验前），经CRC16计算得到校验码为：850E。则替换校验码后设备发送的升级包下载结果的消息为：FFFE0116850E000100。

物联网平台的应答消息

物联网平台收到设备上报的软件包下载结果后，将会向设备返回应答消息，各个消息字段的填写如下。

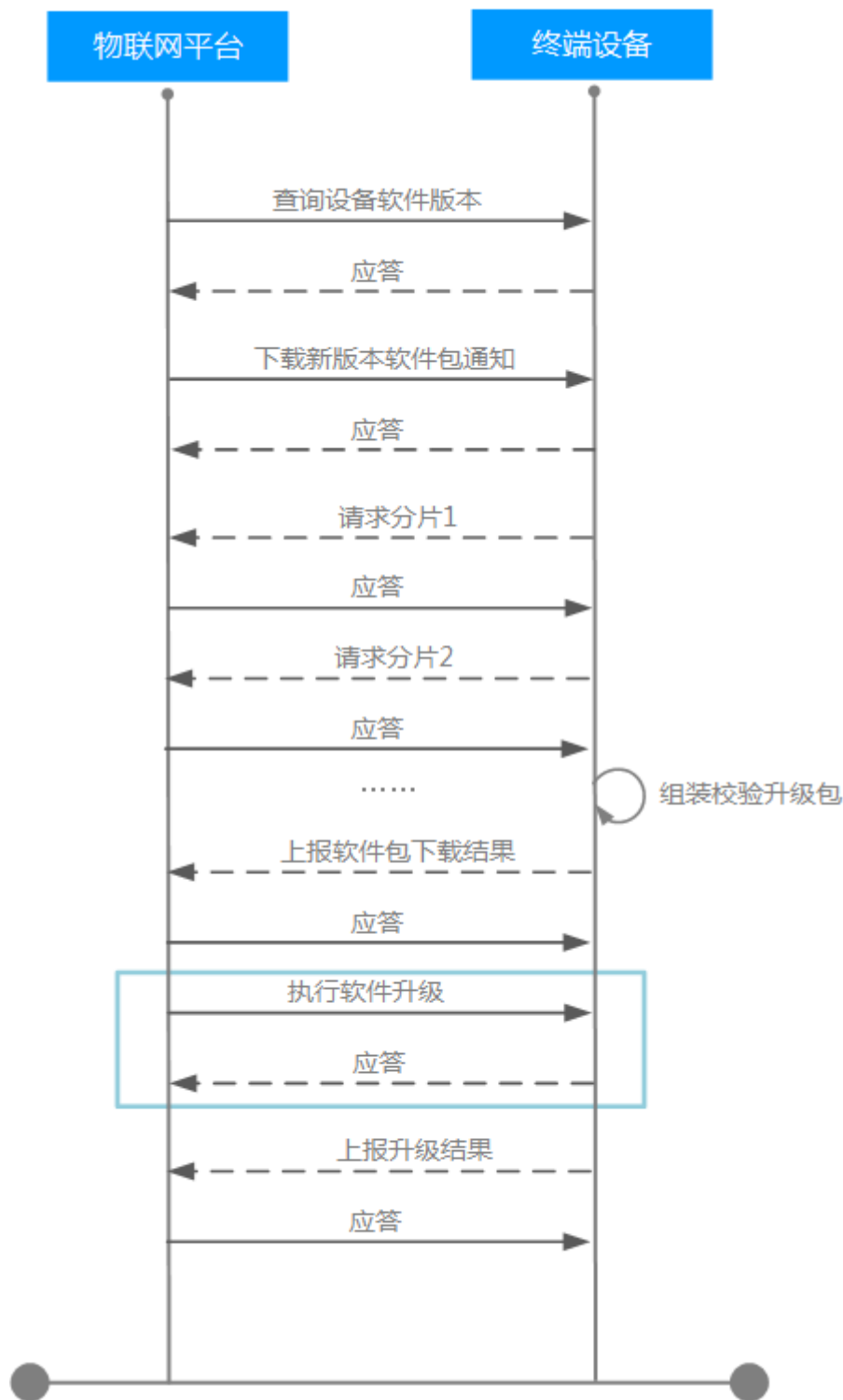
- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：与请求的消息码一致：16。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段定义得出该数据长度为1个字节，转换为十六进制为：0001。
- 数据区：处理成功，则返回00，处理失败返回80。本示例以返回00处理成功为例进行说明。

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X80升级任务不存在。

物联网平台向设备应答的消息为：FFFE 01 16 0000 0001 00（CRC16校验前），经CRC16计算得到校验码为：850E。则替换校验码后物联网平台向设备应答的消息为：FFFE0116850E000100。

执行软件升级

根据PCP协议约定的交互流程，物联网平台收到设备发送的软件包下载结果通知后，需要通知设备进行升级操作。



物联网平台发送的请求消息

根据PCP消息结构的定义可以得出，物联网平台向设备发送执行软件升级消息，各个消息字段的填写如下：

- 起始标识固定为：FFFE。
- 版本号固定为：01。

- 消息码：与请求的消息码一致，为17。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段的数据类型得出无数据区，即为0字节，转换为十六进制为：0000。
- 数据区：无数据区，无需携带该字段。

字段	数据类型	描述及要求
无数据区		

物联网平台向设备下发的执行软件升级的消息为：FFFE 01 17 0000 0000（CRC16校验前），经CRC16计算得到校验码为：CF90。则替换校验码后物联网平台向设备发送的消息为：FFFE0117CF900000。

设备发送的应答消息

设备收到物联网平台下发的执行升级消息后，将对收到消息后的执行动作进行应答，各消息字段的填写如下。

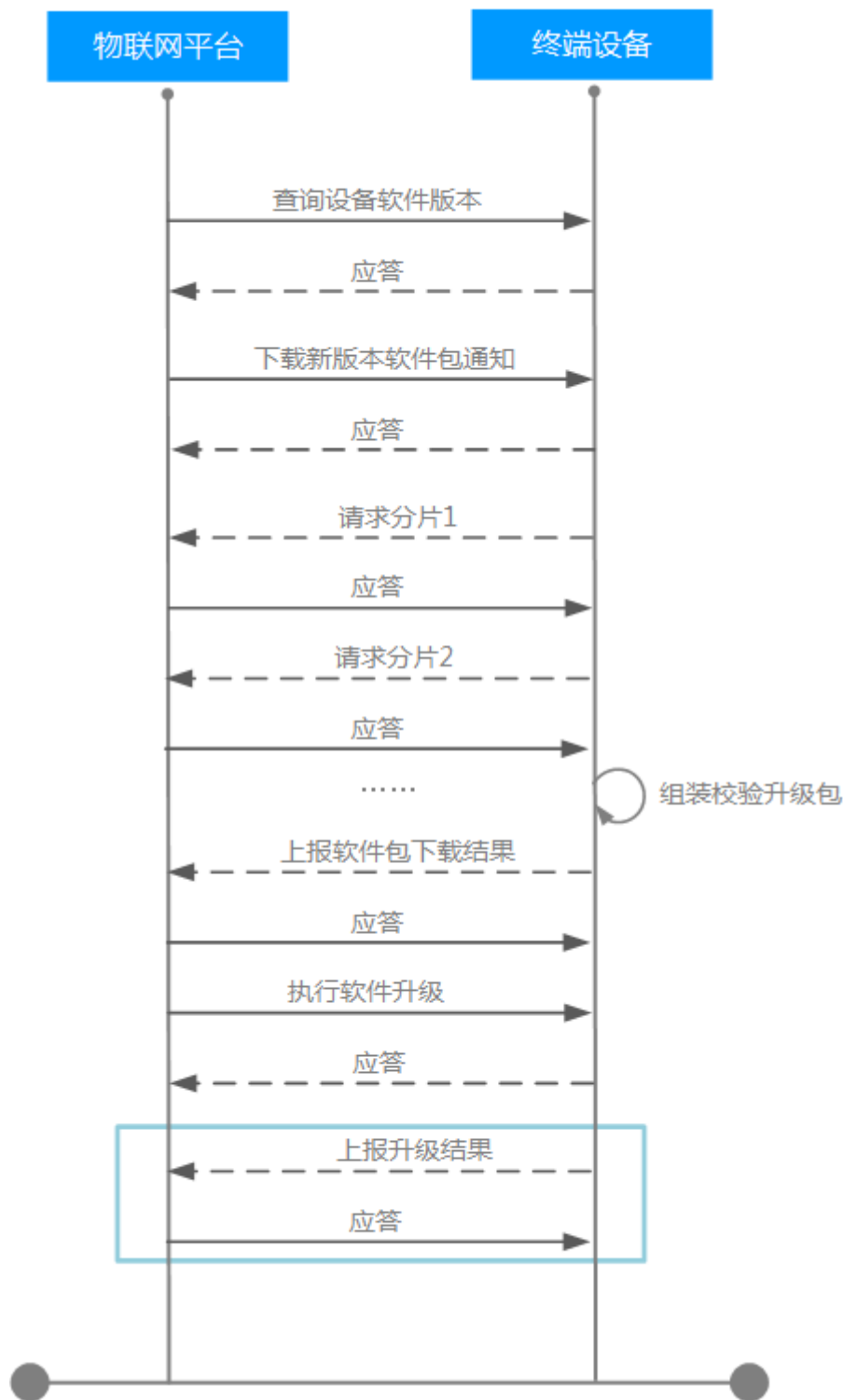
- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：与请求的消息码一致：17。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段定义得出该数据长度为1个字节，转换为十六进制为：0001。
- 数据区：处理成功，则返回00，其它处理结果请参考数据区定义。本示例以返回00处理成功为例进行说明。

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X01设备使用中。 0X04电量不足。 0X05剩余空间不足。 0X09内存不足。

设备向物联网平台应答的消息为：FFFE 01 17 0000 0001 00（CRC16校验前），经CRC16计算得到校验码为：B725。则替换校验码后设备返回的响应消息为：FFFE0117B725000100。

上报升级结果

根据PCP协议约定的交互流程，设备在执行完软件升级后，将会向物联网平台上报升级的结果。



设备发送的请求消息

根据PCP消息结构的定义可以得出，设备向物联网平台上报升级结果，各个消息字段的填写如下：

- 起始标识固定为：FFFE。
- 版本号固定为：01。

- 消息码：与请求的消息码一致，为18。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段的数据类型得出数据区长度为17字节，转换为十六进制为：0011。
- 数据区：结果码，以上报升级成功为例，结果码为00。当前版本号：升级完成后的版本号，与物联网平台下发的软件版本一致，即v1.0，转换为十六进制为：56312E30000000000000000000000000。

字段	数据类型	描述及要求
结果码	BYTE	0X00升级成功。 0X01设备使用中。 0X04电量不足。 0X05剩余空间不足。 0X09内存不足。 0X0A安装升级包失败。 0X7F内部异常。
当前版本号	BYTE[16]	设备当前版本号。

设备向物联网平台上报升级结果的消息为：FFFE 01 18 0000 0011 0056312E300000000000000000000000000000 (CRC16校验前)，经CRC16计算得到校验码为：C7D2。则替换校验码后设备向物联网平台上报升级结果码流为：FFFE0118C7D200110056312E300000000000000000000000000000。

物联网平台发送的应答消息

物联网平台收到设备上报的升级结果消息后，将对设备进行应答，各个消息字段的填写如下。

- 起始标识固定为：FFFE。
- 版本号固定为：01。
- 消息码：与请求的消息码一致：18。
- 校验码：CRC16计算前先用0000替代。
- 数据区长度：根据数据区的字段定义得出该数据长度为1个字节，转换为十六进制为：0001。
- 数据区：处理成功，则返回00，升级任务不存在80。本示例以返回00处理成功为例进行说明。

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X80升级任务不存在。

物联网平台向设备的应答消息为：FFFE 01 18 0000 0001 00 (CRC16校验前)，经CRC16计算得到校验码为：AFA1。则替换校验码后物联网平台返回的应答消息为：FFFE0118AFA1000100。

至此，设备的软件升级功能适配就完成了，赶紧动手试一试吧。

CRC16 算法代码样例

基于JAVA的CRC16算法样例:

```
public class CRC16 {  
  
    /*  
    * CCITT标准CRC16(1021)余数表 CRC16-CCITT ISO HDLC, ITU X.25,  $x^{16}+x^{12}+x^5+1$  多项式  
    * 高位在先时生成多项式  $Gm=0x11021$  低位在先时生成多项式,  $Gm=0x8408$  本例采用高位在先  
    */  
    private static int[] crc16_ccitt_table = { 0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,  
        0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef, 0x1231, 0x0210, 0x3273, 0x2252,  
        0x52b5, 0x4294, 0x72f7, 0x62d6, 0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,  
        0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485, 0xa56a, 0xb54b, 0x8528, 0x9509,  
        0xe5ee, 0xf5cf, 0xc5ac, 0xd58d, 0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,  
        0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc, 0x48c4, 0x58e5, 0x6886, 0x78a7,  
        0x0840, 0x1861, 0x2802, 0x3823, 0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,  
        0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12, 0xdbfd, 0xcdbc, 0xfbbf, 0xeb9e,  
        0x9b79, 0x8b58, 0xbb3b, 0xab1a, 0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,  
        0xedaе, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49, 0x7e97, 0x6eb6, 0x5ed5, 0x4ef4,  
        0x3e13, 0x2e32, 0x1e51, 0x0e70, 0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,  
        0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f, 0x1080, 0x00a1, 0x30c2, 0x20e3,  
        0x5004, 0x4025, 0x7046, 0x6067, 0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,  
        0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256, 0xb5ea, 0xa5cb, 0x95a8, 0x8589,  
        0xf56e, 0xe54f, 0xd52c, 0xc50d, 0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,  
        0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c, 0x26d3, 0x36f2, 0x0691, 0x16b0,  
        0x6657, 0x7676, 0x4615, 0x5634, 0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,  
        0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3, 0xcbb7, 0xdb5c, 0xeb3f, 0xfb1e,  
        0x8bf9, 0x9bd8, 0xabbb, 0xbb9a, 0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,  
        0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaа, 0xad8b, 0x9de8, 0x8dc9, 0x7c26, 0x6c07, 0x5c64, 0x4c45,  
        0x3ca2, 0x2c83, 0x1ce0, 0x0cc1, 0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,  
        0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0 };  
  
    /**  
    *  
    * @param reg_init  
    *     CRC校验时初值  
    * @param message  
    *     校验值  
    * @return  
    */  
    private static int do_crc(int reg_init, byte[] message) {  
        int crc_reg = reg_init;  
        for (int i = 0; i < message.length; i++) {  
            crc_reg = (crc_reg >> 8) ^ crc16_ccitt_table[(crc_reg ^ message[i]) & 0xff];  
        }  
        return crc_reg;  
    }  
  
    /**  
    * 根据数据生成CRC校验码  
    *  
    * @param message  
    *     byte数据  
    * @return int 返校码  
    */  
    public static int do_crc(byte[] message) {  
        // 计算CRC校验时初值从0x0000开始。  
        int crc_reg = 0x0000;  
        return do_crc(crc_reg, message);  
    }  
}
```

基于C的CRC16算法样例:

```
/**  
* CCITT标准CRC16(1021)余数表 CRC16-CCITT ISO HDLC, ITU X.25,  $x^{16}+x^{12}+x^5+1$  多项式  
* 高位在先时生成多项式  $Gm=0x11021$  低位在先时生成多项式,  $Gm=0x8408$  本例采用高位在先
```

```
*/
const unsigned short crc16_table[256] = {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
    0xdbfd, 0xcdbdc, 0xfdbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
    0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
    0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
    0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
    0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
    0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
    0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
    0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
    0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
    0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
    0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
    0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};

int do_crc(int reg_init, byte* data, int length)
{
    int cnt;
    int crc_reg = reg_init;
    for (cnt = 0; cnt < length; cnt++)
    {
        crc_reg = (crc_reg >> 8) ^ crc16_table[(crc_reg ^ *(data++)) & 0xFF];
    }
    return crc_reg;
}

int main(int argc, char **argv)
{
    //FFFE011300000000用byte数组表示:
    byte message[8] = {0xFF,0xFE,0x01,0x13,0x00,0x00,0x00,0x00};
    // 计算CRC校验时初值从0x0000开始
    int a = do_crc(0x0000, message, 8);
    printf("a ==> %x\n", a);
}
```

15.3 PCP 协议介绍（联通用户专用）

平台升级协议（PCP协议）规定了设备和平台之间升级的通信内容与格式。

本协议规定设备和IoT平台（以下简称“平台”）之间的应用层升级协议（简称“PCP协议”），用于实现设备的升级。

通讯方式

1. PCP协议运行在应用层，底层可以是LWM2M/CoAP/MQTT或者其他非流式协议。
2. 由于PCP协议消息没有使用单独的端口号，并且不依赖于底层协议，为了和设备业务消息区分，PCP协议固定以0xFFFE作为起始字节。因此要求设备的业务消息的前两个字节不能是0xFFFE，更多细节参考附录[PCP消息识别](#)。
3. 本协议消息采用一问一答模式，所有请求消息都有一个响应消息。

消息结构

字段名	字段类型	描述和要求
起始标识	WORD	起始标识，固定为0xFFFE。
版本号	BYTE	高四位预留；低四位表示协议版本号，当前为1。
消息码	BYTE	标识物联网平台与设备之间的请求消息类型，应答消息的消息码和请求消息相同。消息码的定义为： <ul style="list-style-type: none">● 0-18：预留消息码，暂未使用。● 19：查询设备版本。● 20：下载新版本软件包通知。● 21：请求下载升级包。● 22：上报升级包下载结果。● 23：执行软件升级。● 24：上报升级结果。● 25-127：预留消息码，暂未使用。
校验码	WORD	从起始标识到数据区的最后一个字节的CRC16校验值，计算前先把校验码字段置为0，计算完成后把结果写到校验码字段。
数据区长度	WORD	数据区的长度。
数据区	BYTE[n]	可变长度，具体由各个指令定义，可参考下面介绍的各个指令对应的请求消息和应答消息定义。

数据类型

数据类型	描述
BYTE	无符号一字节整数
WORD	无符号二字节整数
DWORD	无符号四字节整数

数据类型	描述
BYTE[n]	n字节的十六进制数
STRING	字符串

📖 说明

本协议采用网络序来传输WORD和DWORD。

查询设备版本消息

请求消息：

方向：平台->设备

字段	数据类型	描述及要求
无数据区		

响应消息：

方向：设备->平台

字段	数据类型	描述及要求
结果码	BYTE	“0X00” 处理成功
当前版本号	BYTE[16]	当前版本号，由ASCII字符组成，位数不足时，后补“0X00”。

📖 说明

- 正常处理：平台根据版本号判断设备是否需要升级，如果需要，下发请求升级。
- 异常处理：如果响应超时，平台中止升级任务。

新版本通知消息

请求消息：

方向：平台->设备

字段	数据类型	描述及要求
目的版本号	BYTE[16]	目的版本号，由ASCII字符组成，位数不足时，后补“0X00”。

字段	数据类型	描述及要求
升级包分片大小	WORD	每个分片的大小
升级包分片总数	WORD	升级包分片总数
升级包校验码	WORD	升级包校验码。用户上传升级包时，需要在升级包描述文件里填写校验码

应答消息：

方向：设备->平台

字段	数据类型	描述及要求
结果码	BYTE	“0X00” 允许升级 “0X01” 设备使用中 “0X02” 信号质量差 “0X03” 已经是最新版本 “0X04” 电量不足 “0X05” 剩余空间不足 “0X09” 内存不足 “0X7F” 内部异常

说明

- 正常处理：如果设备不允许升级，平台中止升级任务。
- 异常处理：如果响应超时，而且没收到请求升级包消息，平台中止升级任务。

请求消息包消息

请求消息：

方向：设备->平台

字段	数据类型	描述及要求
目的版本号	BYTE[16]	目的版本号，由ASCII字符组成，位数不足时，后补“0X00”。

字段	数据类型	描述及要求
分片序号	WORD	表示请求获取的分片序号，从0开始计算，分片的总数为软件包大小除以每个分片的大小并向上取整获得。设备可以保存已经收到的分片，下次直接从缺失的分片开始请求，达到断点续传的效果。

响应消息：

方向：平台->设备

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X80升级任务不存在。 0X81指定的分片不存在。
分片序号	WORD	表示返回的分片序号。
分片数据	BYTE[n]	分片的内容，n为实际的分片大小。如果结果码不为0，则不带此字段。

上报升级包下载状态消息

请求消息：

方向：设备->平台

字段	数据类型	描述及要求
下载状态	BYTE	0X00下载成功。 0X05剩余空间不足。 0X06下载超时。 0X07升级包校验失败。 0X08升级包类型不支持。

响应消息：

方向：平台->设备

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X80升级任务不存在。

执行升级消息

请求消息：

方向：平台->设备

字段	数据类型	描述及要求
无数据区		

响应消息：

方向：设备->平台

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X01设备使用中。 0X04电量不足。 0X05剩余空间不足。 0X09内存不足。

上报升级结果消息

请求消息：

方向：设备->平台

字段	数据类型	描述及要求
结果码	BYTE	0X00升级成功。 0X01设备使用中。 0X04电量不足。 0X05剩余空间不足。 0X09内存不足。 0X0A安装升级包失败。 0X7F内部异常。
当前版本号	BYTE[16]	设备当前版本号。

响应消息:

方向: 平台->设备

字段	数据类型	描述及要求
结果码	BYTE	0X00处理成功。 0X80升级任务不存在。

PCP 消息识别

由于PCP协议消息和设备业务消息共用一个端口和URL通讯，平台收到设备的消息时，按照如下步骤判断是PCP协议消息还是业务消息：

1. 检查设备是否支持软件升级（根据设备profile的omCapability.upgradeCapability定义），如果不支持，则认为是业务消息。
2. 检查设备软件升级协议是否是PCP，如果不是，则认为是业务消息。
3. 检查消息前两个字节是否为0XFFFE，如果不是，则认为是业务消息。
4. 检查版本号是否合法，如果不合法，则认为是业务消息。
5. 检查消息码是否合法，如果不合法，则认为是业务消息。
6. 检查校验码是否正确，如果不正确，则认为是业务消息。
7. 检查数据区长度是否正确，如果不正确，则认为是业务消息。
8. 如果以上检查都通过，认为是PCP协议消息。

说明

对设备的要求：需要设备保证业务消息的起始字节不是0XFFFE。

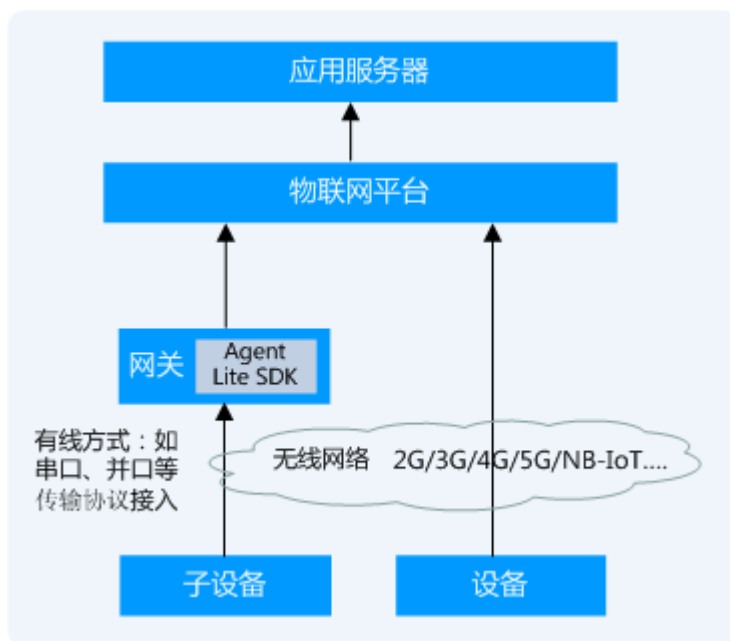
16 网关与子设备（联通用户专用）

非联通用户请查看[设备接入服务](#)。

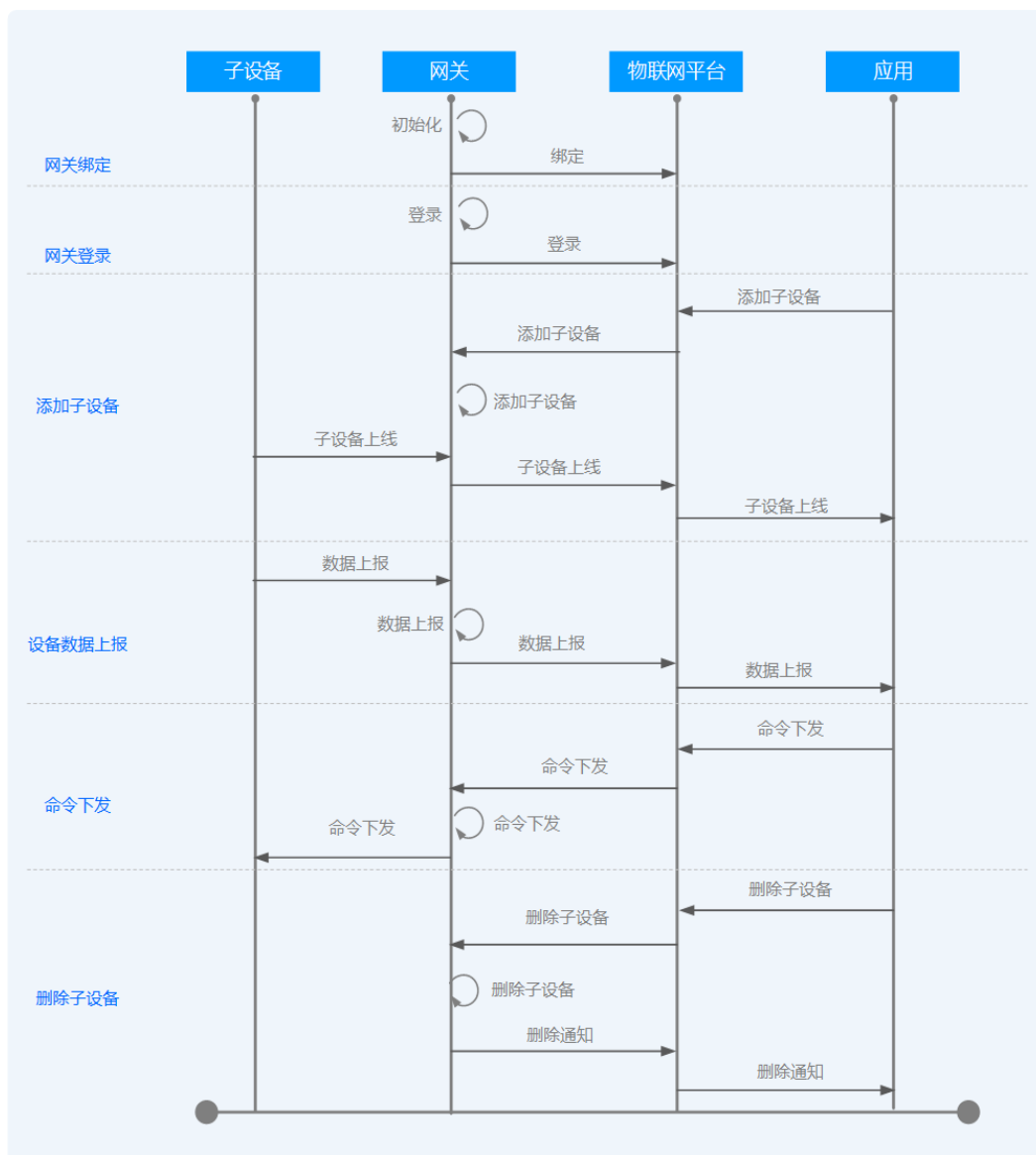
概述

物联网平台支持设备直连，也支持设备挂载在网关上，作为网关的子设备，由网关直连。

针对未实现TCP/IP协议栈的设备，由于无法直接同物联网平台通信，它需要通过网关进行数据转发，如下图所示。



接入流程



Agent Lite SDK提供相关的接口，开发者调用这些接口网关实现与物联网平台的对接。不同语言的SDK的接口名称不一样，每个接口功能请参见[Agent Lite SDK 使用指南（C）](#)、[Agent Lite SDK 使用指南（JAVA）](#)、[Agent Lite SDK 使用指南（Android）](#)。

- 1、网关调用Agent Lite SDK 的初始化接口，初始化Agent Lite资源。
- 2、在物联网平台的控制台上上传网关的产品模型，并注册网关。
- 3、网关第一次接入物联网平台时需要进行绑定操作，从而将网关与平台进行关联。需先修改配置文件里的设备固有信息（verifyCode和其他必要的设备信息），然后调用绑定接口设置绑定配置，并绑定网关。
- 4、网关绑定成功后，需要进行登录操作。网关先调用SDK接口配置所需的登录信息，然后调用登录接口进行网关登录。
- 5、在物联网平台的控制台上上传子设备的产品模型。

6、在网关登录成功后，网关调用添加非直连设备接口，填写设备相关信息（与产品模型定义一致），添加子设备到物联网平台（同时网关侧完成和子设备之间的对接，并上电）。添加成功后，在物联网平台上就能查看添加的子设备（[如何查看？](#)）。

用户也可以通过控制台添加子设备，[如何添加？](#)

7、子设备添加完后，一般情况下在物联网平台上显示是“离线”状态。所以在子设备添加成功后，或者在子设备上报数据前，网关要调用“设备状态更新”接口进行设备状态更新。

📖 说明

子设备的状态表示子设备接入网关的状态，由网关上报到物联网平台进行状态的刷新；如果网关不能正常上报子设备的状态信息到物联网平台，则展示的子设备状态不会刷新。例如：某子设备通过网关接入到物联网平台，子设备状态为在线状态，如果此时网关与物联网平台断开连接，则网关不能上报子设备的状态到物联网平台，该子设备的状态会一直显示在线。

8、网关调用“设备服务数据上报”接口上报网关和子设备的数据，接口里的参数填写网关和子设备的相关设备信息。

9、网关调用“设备命令接收”接口，接收并处理应用服务器或物联网平台下发的命令。

10、应用服务器调用[删除子设备](#)接口，给网关下发删除子设备命令，网关调用SDK上的“设备删除”接口删除在物联网平台上的子设备信息。并向平台返回删除结果。

网关如何接入物联网平台？


通过在网关上集成Agent Lite SDK, 设备将数据上报给网关，通过网关转发到华为物联网平台，网关接入物联网平台的开发指南请参见[设备通过网关接入平台](#)。

物联网平台上如何添加子设备？

方式1

当网关接入物联网平台后，在网关上通过调用Agent Lite的添加非直连设备接口，完成子设备接入到物联网平台。

方式2

登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。单击左下角，进入“设备 > 所有设备”页面，在设备列表中，单击具体的网关进入到网关的详情页面，进入“设置”页签。启用“添加设备模式（NWI）”。




📖 说明

只有在网关的产品模型中定义了“DISCOVERY”的命令，才会在控制台上显示该功能。

该功能用于在物联网平台上添加网关下接入的子设备的场景。子设备通过Z-wave、ZigBee协议接入网关的情况下，需要物联网平台向网关下发开启发现子设备的命令，然后网关与子设备进行匹配和添加。点击“启用”按钮，物联网平台向网关下发发现子设备的命令，如果在一段时间内（一般为180秒，具体请以网关的能力为准）没有发现子设备，网关会自动关闭发现子设备的开关；如果在打开发现子设备的时间窗内，发现并添加了一个子设备后，网关也会自动关闭发现子设备功能。如果需要添加多个子设备，可以在物联网平台上再次向网关下发打开发现子设备的命令。

查看子设备

步骤1 登录[物联网平台控制台](#)，点击右上角“进入设备管理服务”。

步骤2 单击左下角，进入“设备 > 所有设备”页面，在设备列表中，单击具体的网关进入到网关的详情页面，选择“子设备”页签。

步骤3 “子设备”页签呈现通过该网关方式接入到物联网平台的设备，可以查看子设备的状态、设备ID、设备类型等信息。



说明

子设备的状态表示子设备接入网关的状态，由网关上报到物联网平台进行状态的刷新；如果网关不能正常上报子设备的状态信息到物联网平台，则展示的子设备状态不会刷新。例如：某子设备通过网关接入到物联网平台，子设备状态为在线状态，如果此时网关与物联网平台断开连接，则网关不能上报子设备的状态到物联网平台，该子设备的状态会一直显示在线。

步骤4 在子设备页签中，单击具体的子设备，可以查看子设备的[设备详情](#)、[历史上报数据](#)、[查看操作记录](#)。

----结束