目录

[课程内容 2](#_Toc523297246)

[课程目标 3](#_Toc523297247)

[课程准备 3](#_Toc523297248)

[课程知识点 3](#_Toc523297249)

[阿里云物联网平台架构介绍 3](#_Toc523297250)

[阿里云物联网平台功能介绍 5](#_Toc523297251)

[阿里云物联网平台优势介绍 6](#_Toc523297252)

[阿里云物联网平台规格介绍 6](#_Toc523297253)

[物联网平台侧开发 7](#_Toc523297254)

[开通物联网平台 7](#_Toc523297255)

[新建产品与设备 8](#_Toc523297256)

[定义产品功能 10](#_Toc523297257)

[node.js虚拟设备开发 11](#_Toc523297258)

[新建项目 11](#_Toc523297259)

[编码 13](#_Toc523297260)

[运行调试 15](#_Toc523297261)

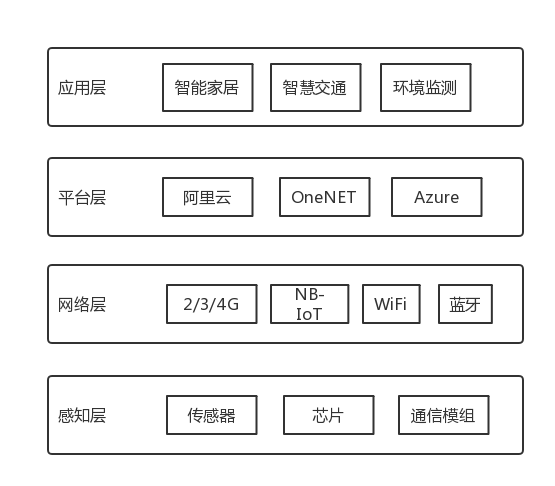
[设备上线 16](#_Toc523297262)

[消息上下行 16](#_Toc523297263)

[课程小结 18](#_Toc523297264)

**LP简介与快速上手**

在物联网的体系架构中，物联网平台处于软硬结合的枢纽位置，向下接入分散的物联网感知层，汇集传感数据；向上提供面向应用开发的基础性平台和面向底层网络的统一数据接口，支撑物联网应用的开发。物联网平台也被称为物联网的战略要塞。



阿里云物联网平台Link Platform是阿里云专门推出的面向物联网开发人员的一站式设备管理平台，可以实现海量设备的接入与管理，同时提供规则引擎与众多的阿里云*产品*打通，使开发者能够灵活快速地实现数据采集+计算+存储的全栈服务，构建物联网应用。

本次课程将系统地介绍阿里云物联网平台的功能优势、组成架构、以及不同规格的对比，让读者对阿里云物联网平台在开发中的定位以及功能特色有所了解。在此基础上，我们将以Node.js版本的设备端SDK为例，介绍物联网平台的接入和上下行通信，让读者在手边没有开发板的场景下也能体验整个设备接入的流程，实现快速上手。

**课程内容**

* 介绍阿里云物联网平台的功能、架构与优势
* 对比介绍基础版与高级版阿里云物联网平台
* 开发node.js虚拟设备接入物联网平台

**课程目标**

* 掌握阿里云物联网平台的功能和架构，了解其优势
* 熟悉基础版与高级版物联网平台的区别
* 熟悉物联网平台的基本操作
* 使用node.js虚拟设备成功接入物联网平台，并实现双向通信

**课程准备**

* 运行Windows系统的 PC机一台
* Node.js LTS版本（>=8.9）
* 文本编辑器VSCode
* 开通阿里云物联网平台*产品*

**课程知识点**

本次课程主要向读者介绍阿里云物联网平台的*产品*功能、*产品*架构、*产品*优势与*产品*规格，对于其中涉及到的各个组件例如IoT Hub、规则引擎，各个功能例如物模型、脚本解析等都将在后续课程中进行详细的讲解。

**阿里云物联网平台功能介绍**

阿里云物联网平台的主要功能包括：

* 设备接入、管理与维护

阿里云物联网平台提供不同网络的设备接入方案，例如2/3/4G、NB-IoT、LoRa等，解决异构网络设备接入管理的痛点。物联网平台还提供多种协议的设备端SDK，例如MQTT、CoAP等。同时阿里云还开源多种平台设备端代码，并且提供跨平台移植手册，让企业有能力基于不同平台将设备接入。

设备接入云端之后，物联网平台还提供完整的设备生命周期管理能力，并支持设备功能定义、脚本解析、在线调试、远程配置、固件升级、设备影子、实时监控、数据存储等功能。

* 安全与权限管理

阿里云物联网平台为所有节点（包括设备端和各个云服务）提供身份验证服务，如果没有身份验证，将不能够进行节点间的数据交换，即意味着设备无法与物联网平台通信。一机一密的设备认证机制降低设备被攻破的安全风险，设备秘钥的安全管理机制和芯片级的安全存储方案防止设备秘钥被泄露和破解。

物联网平台还提供设备级的权限粒度服务，保证设备或者应用程序只有在具有相应的访问权限时才能操作资源，保证数据的安全性。

同时，物联网平台还提供端到端TLS加密的数据传输通道，保证数据的机密性和完整性。

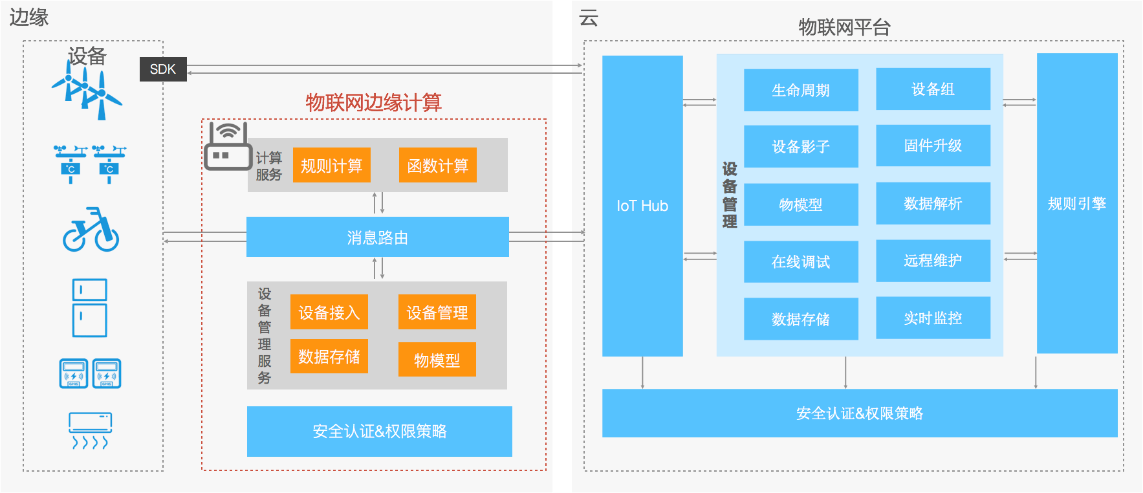
* 设备数据计算与存储

物联网平台基础版不会做数据计算和存储工作，但是用户可以通过规则引擎将平台与阿里云的计算和存储*产品*打通，通过对数据的解析与转发实现采集+计算+存储的完整解决方案。

物联网平台高级版会提供稳定可靠的结构化数据存储能力，客户无需购买存储*产品*即可将设备结构化的数据存储下来，并且能够实时访问。同时，用户依然可以通过规则引擎将数据转发到存储*产品*中，实现更加丰富的应用扩展。

**阿里云物联网平台架构介绍**

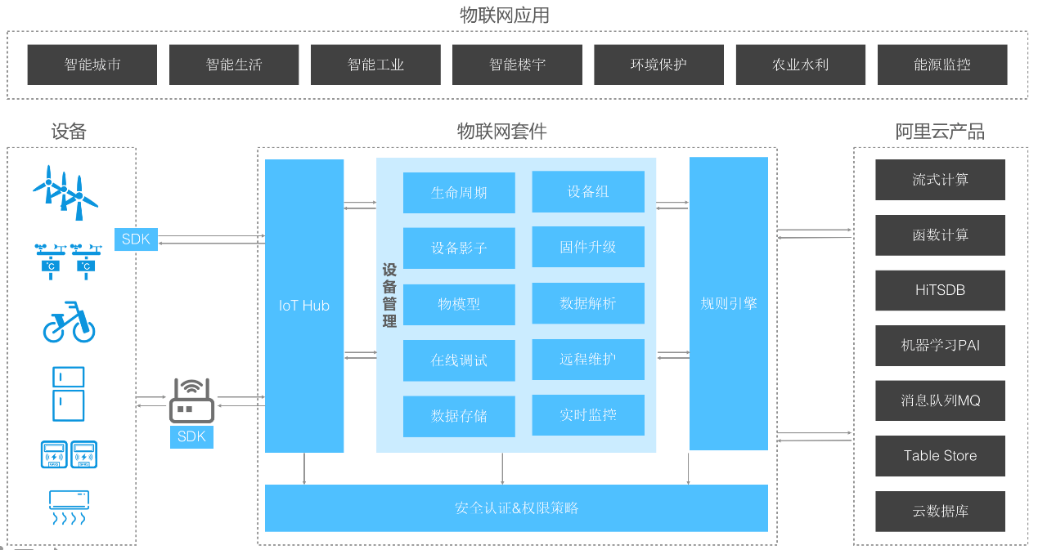
目前，阿里云物联网平台提供的能力分为边缘端和云端两大部分，整体架构图如下：



用户可以使用边缘计算能力，将计算扩展到边缘端，在离设备最近的位置构建边缘计算节点，有关物联网边缘计算（Link Edge）的更多介绍可参阅文档：

https://help.aliyun.com/document\_detail/85155.html。

本课程主要介绍阿里云物联网平台的云端能力，云端部分由四个组件构成：



* IoT Hub

IoT Hub组件帮助设备连接到LP，为设备和物联网应用程序提供发布和接收消息的安全通道，目前支持设备以CoAP协议、MQTT协议和HTTP协议接入。

* 安全认证和权限策略

LP提供多重防护来保障设备云端安全：

物联网平台为每个设备颁发唯一证书，设备使用证书进行身份验证后才能连接到IoT Hub。

针对不同安全等级和产线烧录的要求，物联网平台为开发者提供了多种设备认证方式。

授权粒度精确到设备级别，任何设备只能对自己所属的Topic发布订阅消息，服务端必须凭借阿里云AK才能对账号下所属的Topic进行操作。

* 规则引擎

规则引擎让用户能够通过编写SQL语句，过滤数据、处理数据，并将数据发送到阿里云的消息队列等其他*产品*中去。

* 设备管理

物联网平台为用户提供功能丰富的设备管理服务，例如设备生命周期管理，状态管理，拓扑关系管理，设备影子等等。

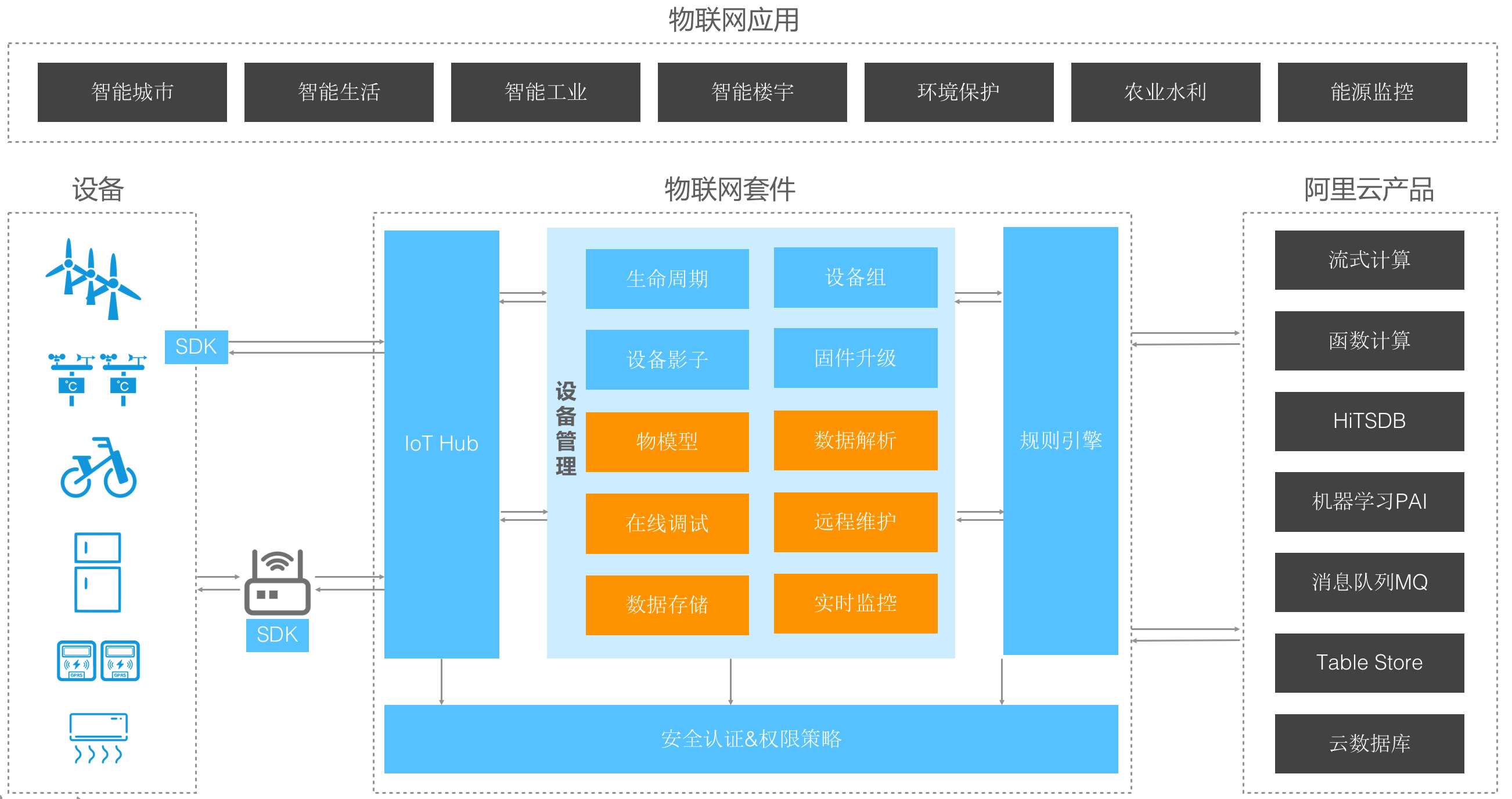
**阿里云物联网平台优势介绍**

相较于传统开发，基于阿里云物联网平台进行开发具有以下特点及优势：

|  |  |  |
| --- | --- | --- |
|  | 基于阿里云物联网平台开发 | 传统开发 |
| 设备接入 | 提供不同环境下设备端SDK，帮助设备快速接入云端；支持全球设备接入，支持异构网络设备接入，支持多协议设备接入 | 不仅需要搭建基础设施，而且需要自行寻找嵌入式开发人员与云端开发人员联合开发，工作量大，效率低 |
| 性能 | 具备亿级设备的长连接能力，百万级并发的能力，并且架构支撑水平性扩展 | 需要自行实现扩展性架构，极难做到从设备粒度调度服务器与负载均衡等基础设施 |
| 安全 | 提供多重防护保障设备云端安全 | 需要额外开发和部署各种安全措施 |
| 稳定 | 服务可用性99.9%，单点故障，自动迁移 | 需要自行发现宕机并完成迁移，迁移过程中服务会中断 |
| 简单易用 | 提供一站式设备管理服务，用户几乎无开发成本即可实时监控设备；提供规则引擎与阿里云*产品*无缝打通，方便用户灵活搭建物联网复杂应用 | 不仅需要购买服务器搭建负载均衡分布式架构接入设备，而且还需要花费大量的人力物力去开发接入+计算+存储这一整套物联网系统 |

**阿里云物联网平台规格介绍**

目前，物联网平台包括两种版本：基础版和高级版。与基础版相比，高级版物联网平台具备一些特有的设备管理功能，如图中的黄色模块所示。



基础版适合具有较强软硬件实力，希望灵活组合阿里云各种云*产品*搭建业务系统的开发者。而高级版则适用于需要较小的机器智能化周期，需要同时解决设备的数据格式和存储问题，需要开发者聚焦于垂直业务系统搭建等快速实现智能转型的场景。不同版本物联网平台的功能详细对比列表可参见：https://help.aliyun.com/document\_detail/50080.html。

**基于物联网平台的开发流程**



使用阿里云物联网平台进行开发的第一步，是在物联网平台控制台上添加*产品*和设备，并获取设备的认证三元组信息（包括*productKey*、*deviceName*与deviceSecret信息）。完成*产品*与设备的创建后，用户还可以根据需要进行物模型定义或解析脚本配置等操作。

平台侧操作完毕后，真实物理设备端即可凭借三元组信息被物联网平台识别，进而连接到平台，并基于Topic实现数据在设备端与云端间的传输。

当设备将数据上报至物联网平台后，通过配置服务端订阅，或使用消息队列等其他*产品*，便可以让平台上的数据流转至您的服务器，用于后续的用户应用开发。

**物联网平台侧开发**

**开通物联网平台**

本课程借助Node.js虚拟设备SDK介绍物联网平台的相关操作和设备接入，帮助大家快速上手。

首先，进入阿里云官方网站https://www.aliyun.com/，使用个人淘宝账号或手机号开通阿里云账号，并通过实名认证。在官方首页点击“*产品*”→“物联网”→“物联网平台”，立即开通“物联网平台”*产品*。



开通成功后点击“管理控制台”进入LP控制台。



**新建产品与设备**

进入控制台后，选择区域为“华东2（上海）”。



在“*产品*管理”页面，点击右侧“创建*产品*”按钮，选择版本为高级版，输入*产品*名称，选择节点类型、设备类型和数据格式后，点击确定。此处创建的示例*产品*名称为Example1，节点类型为“设备”，设备类型为“水表”，数据格式为“Alink JSON”。



* *产品*名称：需要在账号内唯一，可以填写*产品*型号等信息。
* 节点类型：“设备”节点类型指不能挂载子设备的设备，这种设备可以直连IoT Hub，也可以作为网关的子设备连接IoT Hub；“网关”节点类型指可以挂载子设备的直连设备，网关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。
* 设备类型：设备类型是一组预定义的标准功能模板。物联网平台为用户提供了多种*产品*功能模板，并且已经预定义好了相关的标准功能。例如，在“电表”类型中预定义了用电量、电压、电流、总累计量等标准功能；在“水表”类型中预定义了用水量标准功能。在创建*产品*时，我们可以选择某种定义好的*产品*功能模板，在标准功能的基础上编辑、修改功能，还可以添加更多自定义功能。如果不选择功能模板，平台将不会默认创建任何标准功能，该*产品*的功能完全需要用户自己定义，这部分内容在后续教程中会详细介绍。
* 数据格式：设备上下行的数据格式，可以选择为Alink JSON或透传/自定义。Alink JSON是物联网平台高级版为开发者提供的设备与云端的数据交换协议，采用JSON格式；如果用户希望使用自定义的串口数据格式，可以选择数据格式为透传/自定义，用户需要在云端配置数据解析脚本，将自定义格式的数据转换为Alink JSON格式。

*产品*创建成功后，进入“设备管理”页面，选择已创建的高级版*产品*，点击“添加设备”，输入设备名称后确定。



设备名称需要在*产品*内唯一，因为设备名称作为设备的唯一标识符被用于与物联网平台进行通信。

设备添加成功后，平台会生成设备的三元组信息，即*ProductKey*，*DeviceName*，和 DeviceSecret。*ProductKey*是物联网平台为我们创建的*产品*颁发的全局唯一标识符；*DeviceName*是自定义的设备唯一标识符，用于设备认证和通信；DeviceSecret是物联网平台为设备颁发的设备秘钥，用于认证加密，需与DeiceName成对使用。三元组信息与设备一一对应，十分重要。

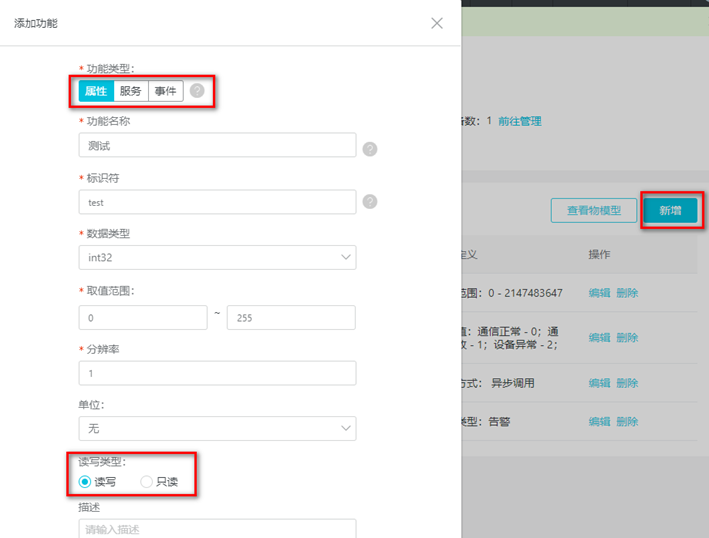
**定义产品功能**

本课程中将用到的Node.js虚拟设备SDK的逻辑是：成功连接到物联网平台后一直监听云端下发的“属性设置”消息，当收到“设置属性”的消息时，通过串口打印在控制台上，并且将属性内容原样上报到物联网平台，因此我们需要在当前*产品*下创建一个“属性”。

从“*产品*管理”栏点击刚才创建的高级版*产品*，进入*产品*详情界面，点击“功能定义”栏目。



点击右侧的“新增”按钮，选择功能类型为“属性”，填写必填项后确认即可。注意读写类型一定要选择为“读写”，因为只读类型的属性不具备“设置”方法，只具有“获取”方法。此处的示例中新增的功能名称为“测试”，标识符为“test”，数据类型为“int32”，取值范围为0~255，分辨率为1，单位无。



进行到这里，物联网平台侧的所需开发就基本完成了，下面我们进行Node.js虚拟设备的开发。

**node.js虚拟设备开发**

**新建项目**

打开VSCode软件，通过点击上方功能栏中的“查看→调试控制台”打开控制台界面。



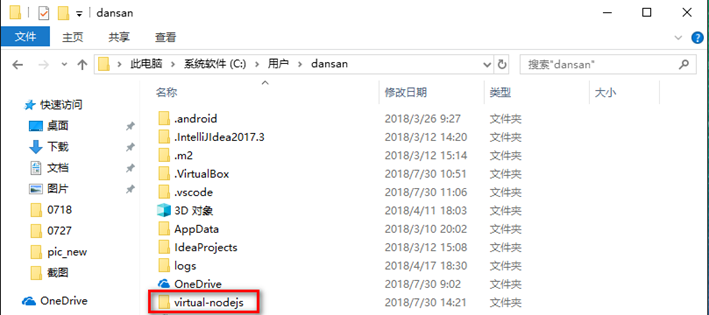
在命令中执行以下两条指令，新建项目目录。

mkdir virtual-nodejs

cd virtual-nodejs



指令成功执行后，我们进入本机的对应目录就可以看到新建的virtual-nodejs文件夹。



然后我们执行npm install --save aliyun-iot-device-sdk命令，安装nodejs设备的SDK。命令成功执行后，我们可以看到刚刚新建的文件夹内容不再为空。

**编码**

点击“文件→打开文件夹”。



选择之前新建的virtual nodejs文件夹位置打开该项目。



成功打开该项目后，点击资源管理器下的virtual-nodejs右侧的“新建文件”按钮，在项目下新建index.js文件。



在index.js文件中编辑写入内容：

|  |
| --- |
| //加载SDK  const aliyunIot = require('aliyun-iot-device-sdk');  // 创建设备实例  const device = aliyunIot.device({  // 激活凭证，这里替换成在平台申请到的真实设备凭证  *productKey*: '<*productKey*>',  *deviceName*: '<*deviceName*>',  deviceSecret: '<deviceSecret>',  });  device.on('connect', () => {  // 连接成功  console.log('connect successfully');  // 监听云端消息  device.serve('property/set', params => {  console.log('receieve params:', params);  // 原样上报  console.log('post props:', params);  device.postProps(params, err => {  if (err) {  return console.log('post error:', err);  }  console.log('post successfully!');  });  });  });  device.on('error', (err) =>{  console.log('error', err);  }); |

上述代码中调用的API来自于上一步执行“npm install --save aliyun-iot-device-sdk”命令时安装的nodejs设备SDK。

* aliyunIot.device(options)

与云端建立连接，返回一个Device连接实例。入参options包括*productKey*、*deviceName*、deviceSecret、regionId和tls信息。

* Event ‘connect’

当连接到云端成功时触发。

* Event ‘error’

当设备不能连接到云端的时候触发。

在上述代码中，当设备成功连接到云端时，将触发监听云端的属性设置消息以及后续逻辑；而不能连接到云端时触发在控制台上打印error内容的动作。

* aliyunIot.device#serve(serviceIdentifier,[callback])

监听物模型服务，其中，入参seviceIdentifier是String类型的服务id，callback的内容为params，即服务参数。

* aliyunIot.device#postProps(params,[callback])

上报物模型属性，其中，入参params为Object类型的属性参数，callback分别err和res两种：err表示错误，比如超时或res.code!=200；res表示服务端回复的消息内容。

设备端SDK提供的更多API内容可参见链接：https://www.npmjs.com/package/aliyun-iot-device-sdk

**运行调试**

**设备上线**

在LP控制台的“设备管理”栏点击刚刚创建好的设备，查看设备详情，获取*productKey*，*deviceName*和deviceSecret信息，并据此修改index.js文件中的“激活凭证”信息。



修改完毕后，用终端在项目目录下执行“node index.js”命令，看到“connect successfully!”，就说明当前node.js虚拟设备已经成功连接到云端了。



此时刷新物联网平台页面，我们可以看到test设备处于在线状态。



**消息上下行**

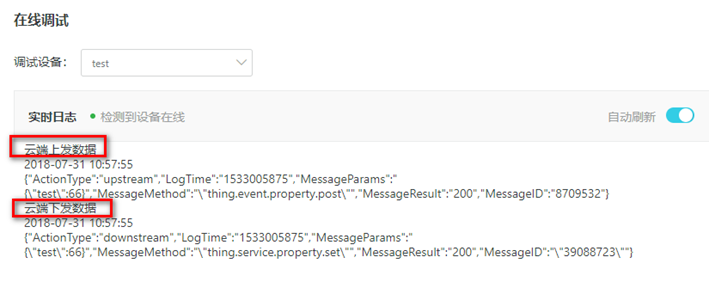
进入*产品*详情界面，点击“在线调试”栏，以下发对设备属性的设置命令。



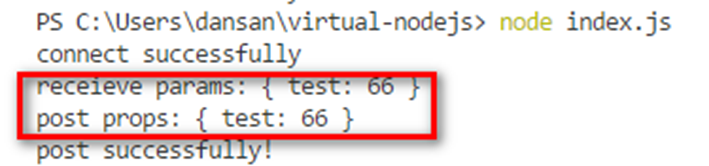
选择调试设备为刚刚创建的test，调试功能为“测试（test）”，方法选择为“设置（即set）”，然后编辑下方JSON编辑器中的内容，例如替换为{“test”:66}。



点击“发送指令”按钮，我们可以在“实时日志”栏中看到，云端下发了一条内容为{“test”:66}的属性设置(set)指令，之后收到了一条内容为{“test”:66}的属性上报(post)消息。



在VSCode终端控制台上查看，发现终端也打出了“receieve params...”和“post successfully!”的日志，说明设备成功接收了数据并将其成功上报。



至此，基于Node.js的虚拟设备端就已经成功接入物联网平台了！

**课程小结**

本次课程学习了阿里云物联网平台Link Platform的架构，优势以及不同规格的平台*产品*的功能差异，同时还对平台的四大组件进行了简要的介绍。在此基础上，通过使用Node.js虚拟设备，让读者可以在手边没有开发板的场景下熟悉物联网平台的相关操作和设备接入流程。