IoT Device Access

FAQs

Issue 1.0

Date 2024-10-31

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Top FAQs

- What Are the Changes Brought by the Integration of IoT Device Management and IoTDA?
- How Do I Obtain the Access Addresses and Certificates of the Old and New Domain Names?
- What Are the Differences in Access Authentication If I Use the New or Old Domain Names?
- Why Does a Command or Properties Delivery Always Time Out?
- How Do I Obtain the ID and Secret of an Application?
- Which Java SDK Demo Should I Refer to?
- Which C SDK Demo Should I Refer to?
- Why Can't I See the Data Reported by a Device on the IoTDA Console?
- Why Did an Application Fail to Call an API?
- How Does an Application Obtain Data Reported by Devices to the IoT Platform?
- How Do I Obtain the Callback URL When Calling the Subscription API?
- What Do I Do If Message Push Fails After Subscription?

2 Solution Consulting

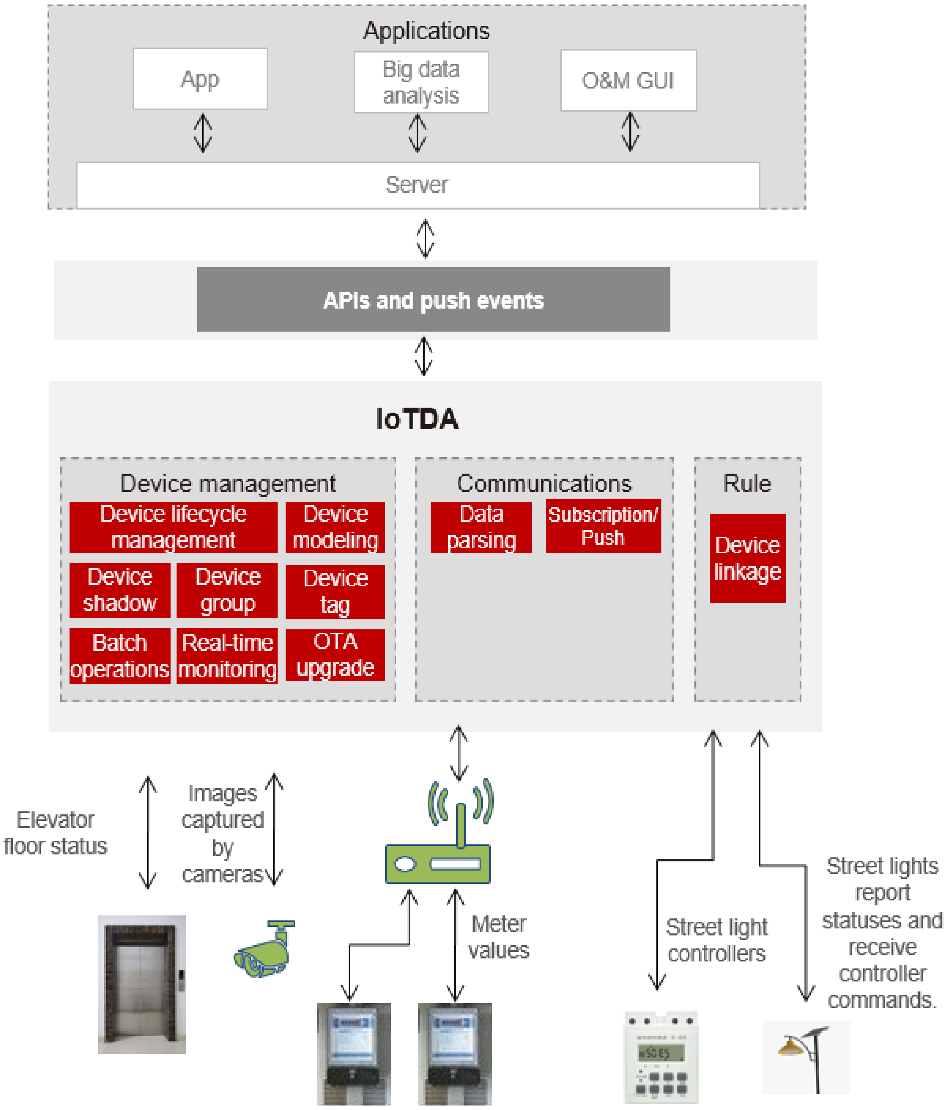
2.1 In What Scenarios Can the IoT Platform Be Applied?

There are four scenarios:

1. Devices + platform + applications

   Devices report data to the platform, and the platform manages the devices and pushes data to applications based on event types. The applications can deliver commands to the platform, which immediately delivers the commands to the devices or caches the commands until the devices go online.

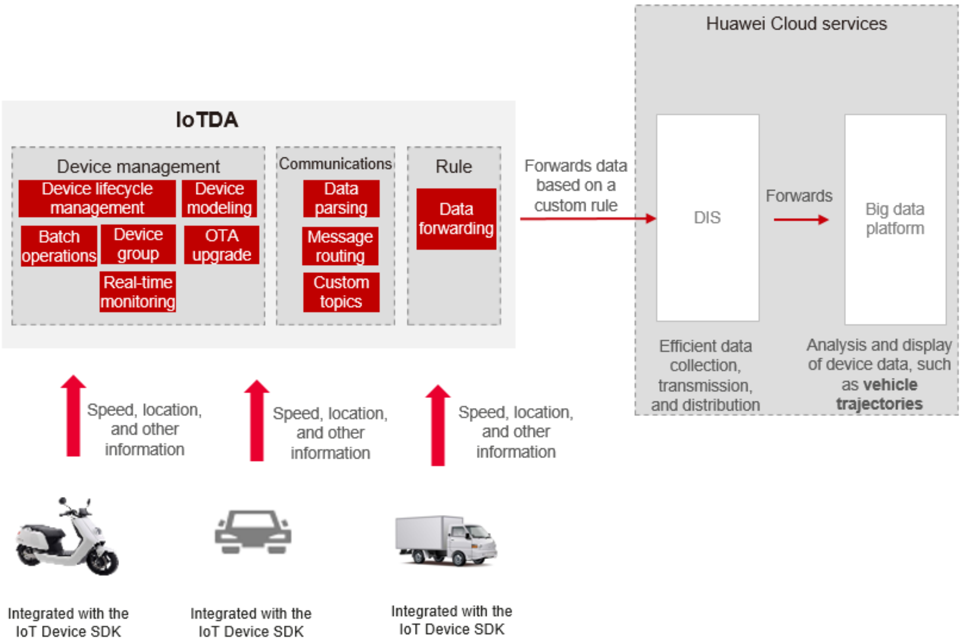
Figure 2-1 Devices + platform + applications



2. Device + platform + Other Huawei Cloud services

   Devices report data to the platform. The platform manages devices and forwards device data to other Huawei Cloud services based on custom data forwarding rules for cross-service processing.

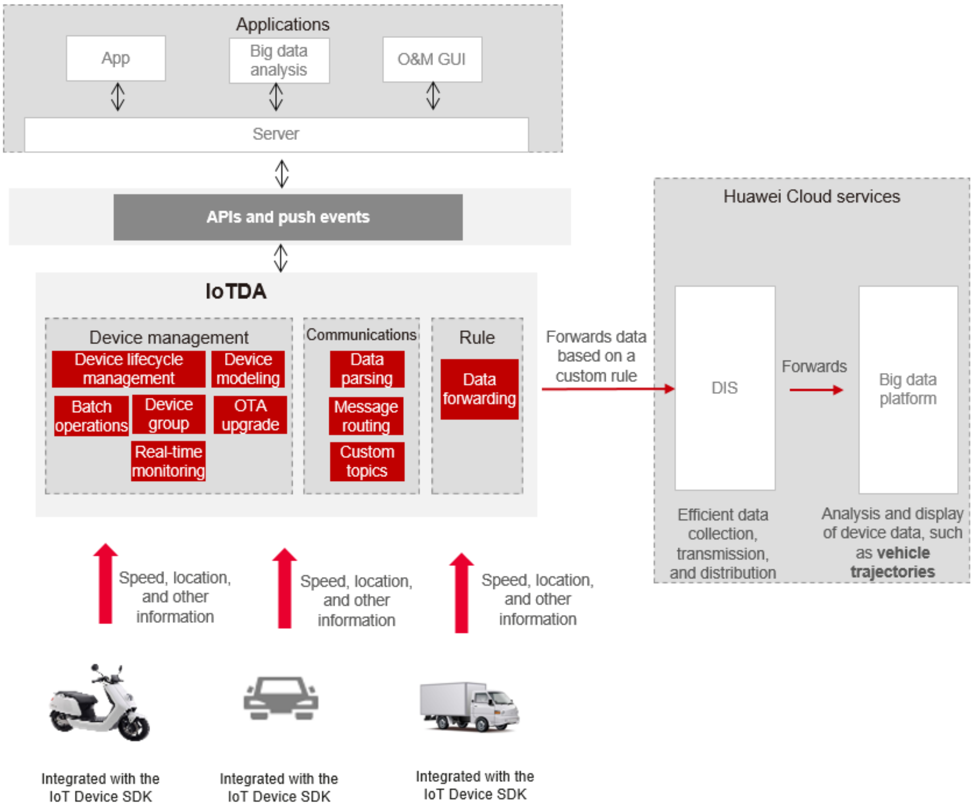
Figure 2-2 Devices + platform + third-party cloud services


Huawei Cloud services
IoTDA
Device management
Device lifecycle management
Device modeling
Batch operations
Device group
OTA upgrade
Real-time monitoring
Communications
Data parsing
Message routing
Custom topics
Rule
Data forwarding
Forwards data based on a custom rule
DIS
Forwards
Big data platform
Efficient data collection, transmission, and distribution
Analysis and display of device data, such as vehicle trajectories
Speed, location, and other information
Speed, location, and other information
Speed, location, and other information
Integrated with the IoT Device SDK
Integrated with the IoT Device SDK
Integrated with the IoT Device SDK

3. Devices + platform + third-party cloud services + applications

   Devices report data to the platform, and the platform manages the devices. Users can obtain data and deliver commands through applications, and set rules to transfer data to third-party cloud services for processing.

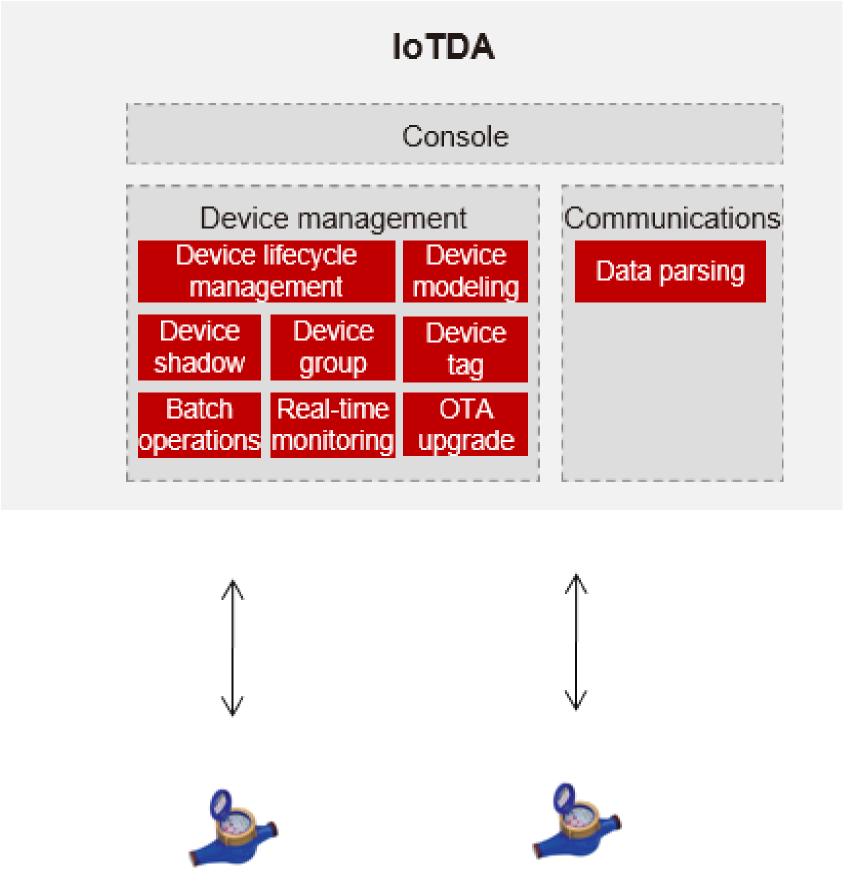
Figure 2-3 Devices + platform + third-party cloud services + applications


Applications
App
Big data analysis
O&M GUI
Server
APIs and push events
Huawei Cloud services
IoTDA
Device management
Device lifecycle management
Device modeling
Batch operations
Device group
OTA upgrade
Real-time monitoring
Communications
Data parsing
Message routing
Custom topics
Rule
Data forwarding
Forwards data based on a custom rule
DIS
Forwards
Big data platform
Efficient data collection, transmission, and distribution
Analysis and display of device data, such as vehicle trajectories
Speed, location, and other information
Speed, location, and other information
Speed, location, and other information
Integrated with the IoT Device SDK
Integrated with the IoT Device SDK
Integrated with the IoT Device SDK

4. Devices + platform

   Devices report data to the platform. Users can view device data and monitor device status in the IoTDA console. No further data processing is required.

Figure 2-4 Devices + platform



2.2 What Are the Changes Brought by the Integration of IoT Device Management and IoTDA?

The changes are as follows:

- **Functions**: IoTDA has integrated the functions of IoT Device Management while still keeping the original IoTDA functions.
  - **Billing**: If you have enabled IoTDA before, you can now use both the IoTDA and IoT Device Management functions, and you will be billed based on the number of messages. If you have enabled both IoTDA and IoT Device Management before, you can use the services in the same way. However, you will be billed based on **Billing Overview** since 00:00 on March 26, 2020 (GMT+08:00).

2.3 Can I Enable IoTDA for IAM Users or Sub-Projects?

1. You can enable IoTDA as an Identity and Access Management (IAM) user. However, the enabled IoTDA service belongs to the IAM master account. That is, the master account is the payment entity.

2. IoTDA cannot be enabled for sub-projects created using IAM.

2.4 Which Regions of Huawei Cloud Are Supported by the IoT Platform?

The original IoTDA and IoT Device Management can be used only in **CN North-Beijing1** and are not open to new users.

2.5 Does Huawei Provide Modules, Hardware Devices, and Application Software?

Huawei provides chips for module manufacturers and access solutions for device manufacturers to integrate.

Multiple partners have completed integration certification with the Huawei IoT platform. We recommend that you find suitable suppliers in the **Huawei Cloud KooGallery**.

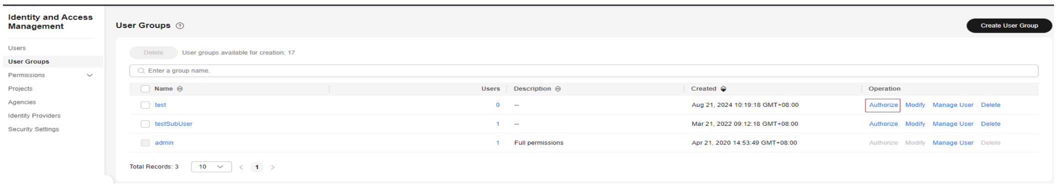
2.6 What Should I Do If I Want to Call an API But Have No Permissions to Do So as an IAM User? (Is It Edition-specific?)

Policy-based permission control is available in IoTDA since April 23, 2020. If an IAM user is not granted the permissions on IoTDA, the message "Operation not allowed. The user does not have the permission" will be displayed when you use the IAM user to access IoTDA resources. To assign permissions of the **IoTDA FullAccess** policy to the user group that the IAM user belongs to, do as follows:

Procedure:

Step 1 Visit **IAM** and click **Try Free** to access the IAM console.

Step 2 In the navigation pane, choose **User Groups**. Click **Authorize** in the row of the target user group.



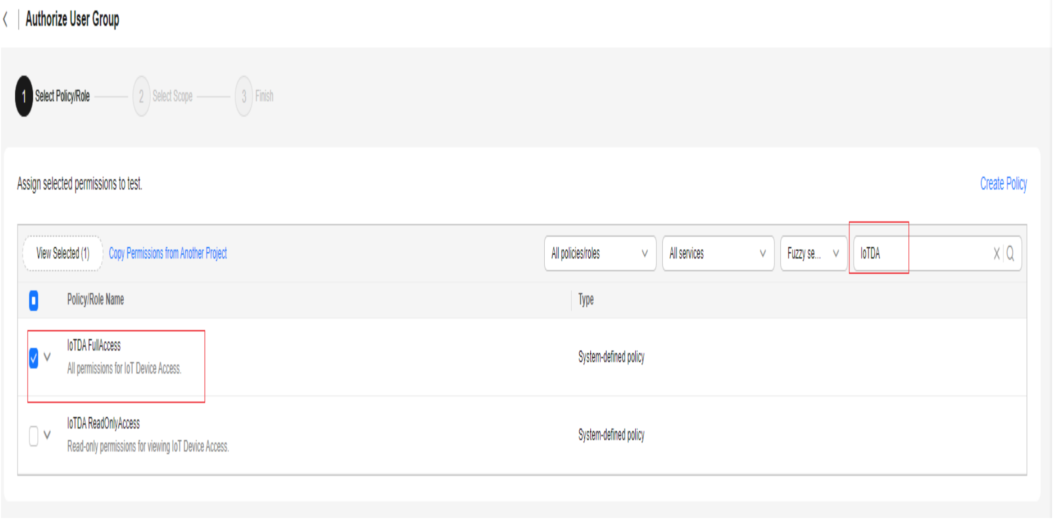
User groups available for creation: 17

Enter a group name.

Name	Users	Description	Created	Operation
test	0	--	Aug 21, 2024 10:19:18 GMT+08:00	Authorize Modify Manage User Delete
testSubUser	1	--	Mar 21, 2022 09:12:18 GMT+08:00	Authorize Modify Manage User Delete
admin	1	Full permissions	Apr 21, 2020 14:53:49 GMT+08:00	Authorize Modify Manage User Delete

Total Records: 3 10 < 1 >

Step 3 Search by IoTDA, select **IoTDA FullAccess**, and click **Next** in the lower right corner.



Authorize User Group

1 Select Policy/Role — 2 Select Scope — 3 Finish

Assign selected permissions to test.

Create Policy

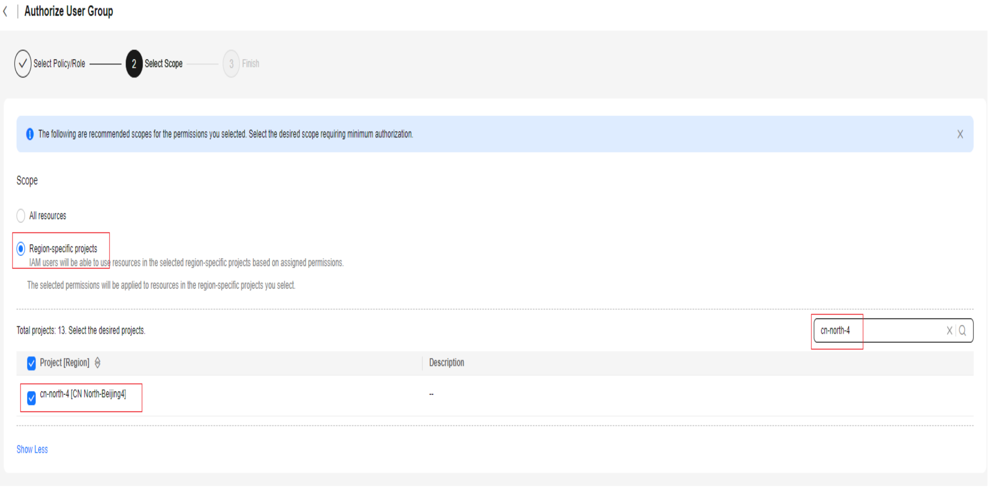
View Selected (1) Copy Permissions from Another Project

All policies/roles ∨ All services ∨ Fuzzy se... ∨ IoTDA

☑	Policy/Role Name	Type
☑	IoTDA FullAccess All permissions for IoT Device Access.	System-defined policy
□	IoTDA ReadOnlyAccess Read-only permissions for viewing IoT Device Access.	System-defined policy

Step 4 Set the scope.

1. Select **Region-specific projects**.
2. Enter search criteria in the search box and select the region to be authorized from the drop-down list box.



Authorize User Group

✓ Select Policy/Role — 2 Select Scope — 3 Finish

The following are recommended scopes for the permissions you selected. Select the desired scope requiring minimum authorization.

Scope

○ All resources

● Region-specific projects
IAM users will be able to use resources in the selected region-specific projects based on assigned permissions.

The selected permissions will be applied to resources in the region-specific projects you select.

Total projects: 13. Select the desired projects.

cn-north-4

☑ Project [Region]	Description
☑ cn-north-4 [CN North-Beijing4]	--

Show Less

NOTE

If the token is domain-level, select all projects. If the token is project-level, select one or more specific projects.

3. Click **OK** in the lower right corner to complete the authorization.

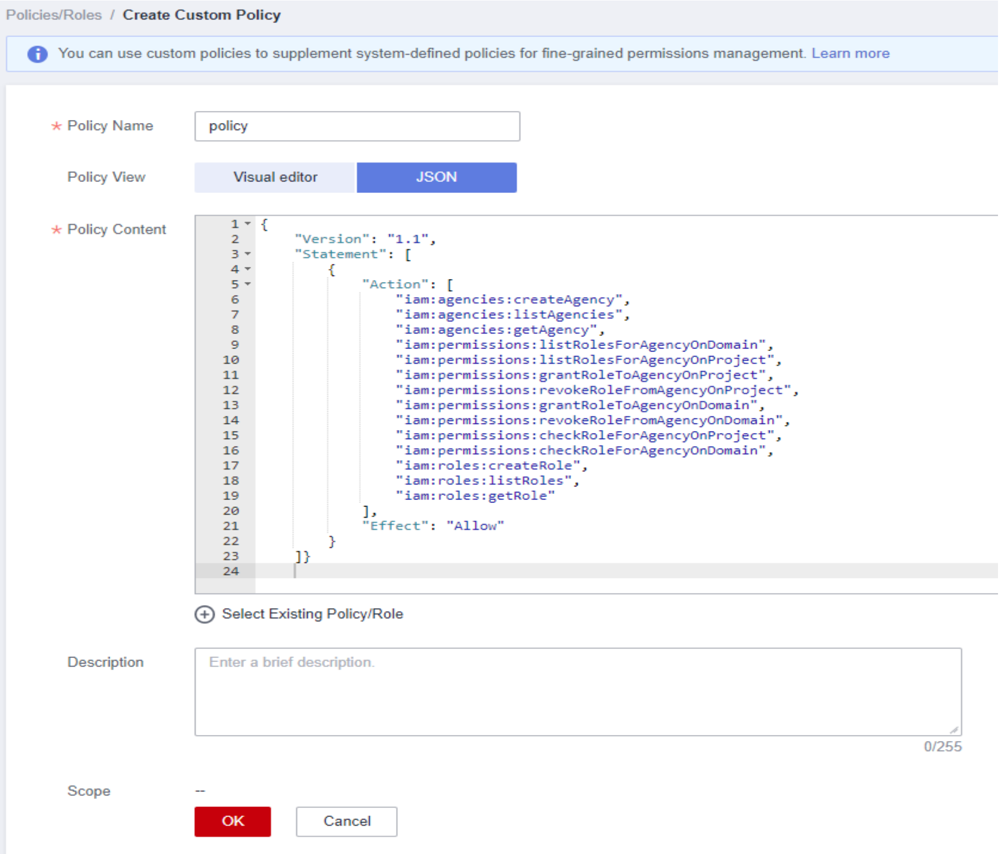
----End

2.7 Why Was I Prompted to Grant Security Administrator Permissions When I Create a Rule or Set Resource File Storage?

This is because the Identity and Access Management (IAM) user that you are using is not granted with the required management permissions. Use the administrator account to perform one of the following methods to grant permissions to the IAM user.

Method 1 (recommended): Create a custom policy (that allows agency creation, role query, and more) and attach it to the user group to which the IAM user belongs.

1. Log in to the IAM console, choose **Permissions** > **Policies/Roles** in the navigation pane, and click **Create Custom Policy** in the upper right corner.



Policies/Roles / Create Custom Policy

You can use custom policies to supplement system-defined policies for fine-grained permissions management. Learn more

* Policy Name: policy

Policy View: Visual editor | JSON

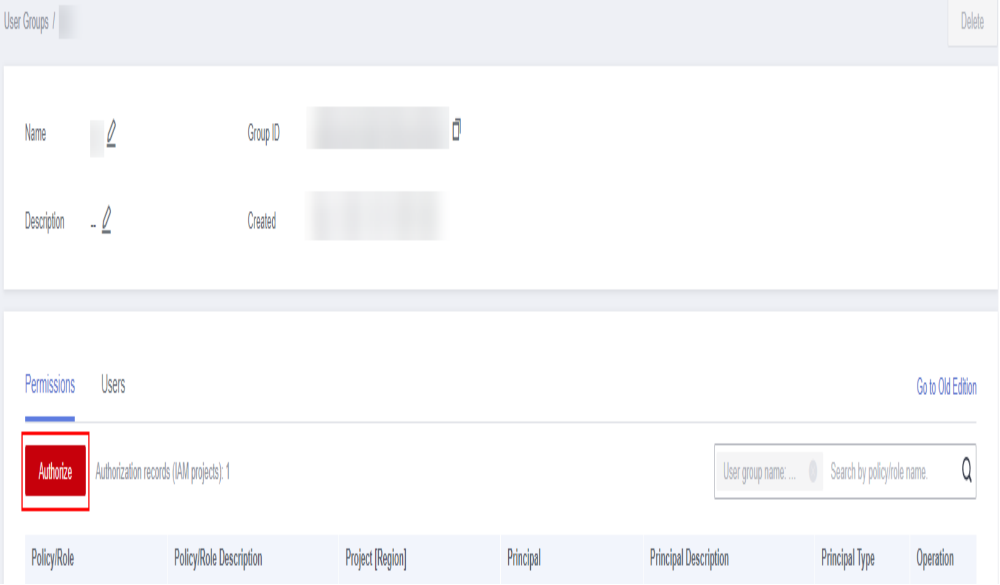
* Policy Content

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "iam:agencies:createAgency",
                "iam:agencies:listAgencies",
                "iam:agencies:getAgency",
                "iam:permissions:listRolesForAgencyOnDomain",
                "iam:permissions:listRolesForAgencyOnProject",
                "iam:permissions:grantRoleToAgencyOnProject",
                "iam:permissions:revokeRoleFromAgencyOnProject",
                "iam:permissions:grantRoleToAgencyOnDomain",
                "iam:permissions:revokeRoleFromAgencyOnDomain",
                "iam:permissions:checkRoleForAgencyOnProject",
                "iam:permissions:checkRoleForAgencyOnDomain",
                "iam:roles:createRole",
                "iam:roles:listRoles",
                "iam:roles:getRole"
            ],
            "Effect": "Allow"
        }
    ]}
```

   - **Policy Name**: Enter a policy name.
   - **Policy View**: Select **JSON**.
   - **Policy Content**: Set this parameter as follows:

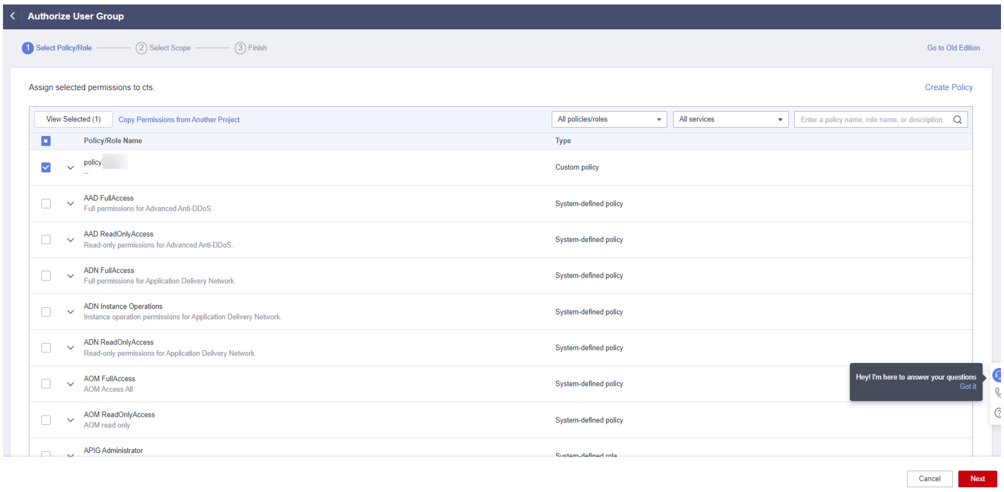
```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "iam:agencies:createAgency",
                "iam:agencies:listAgencies",
                "iam:agencies:getAgency",
                "iam:permissions:listRolesForAgencyOnDomain",
                "iam:permissions:listRolesForAgencyOnProject",
                "iam:permissions:grantRoleToAgencyOnProject",
                "iam:permissions:revokeRoleFromAgencyOnProject",
                "iam:permissions:grantRoleToAgencyOnDomain",
                "iam:permissions:revokeRoleFromAgencyOnDomain",
                "iam:permissions:checkRoleForAgencyOnProject",
                "iam:permissions:checkRoleForAgencyOnDomain",
                "iam:roles:createRole",
                "iam:roles:listRoles",
                "iam:roles:getRole"
            ],
            "Effect": "Allow"
        }
    ]
}
```

2. Click **OK**. The custom policy is created.
3. Choose **User Groups** in the navigation pane and click the target user group.
4. Click the **Permissions** tab and click **Authorize**.



Policy/Role	Policy/Role Description	Project [Region]	Principal	Principal Description	Principal Type	Operation

5. On the **Select Policy/Role** page, select the policy created in **1**, and click **Next**.



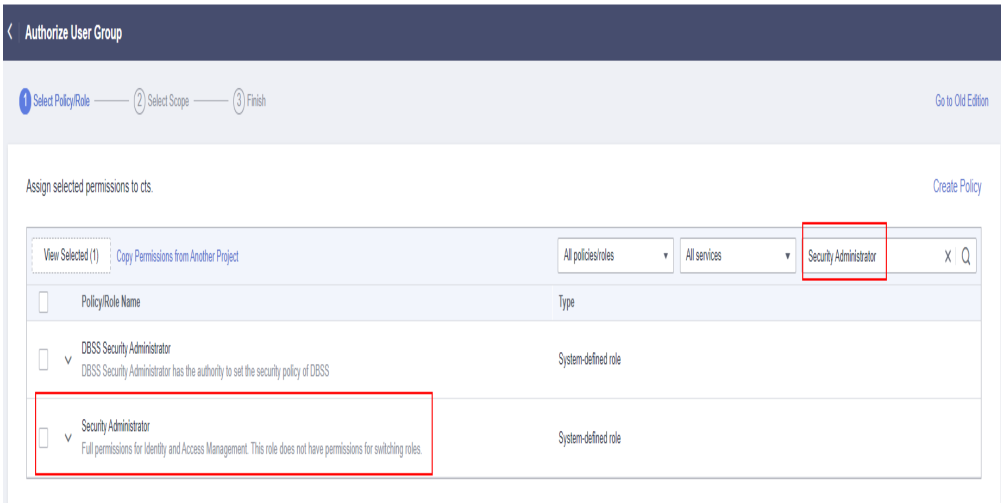
Assign selected permissions to cts.

		Policy/Role Name	Type
☑	⌄	policy: --	Custom policy
□	⌄	AAD FullAccess Full permissions for Advanced Anti-DDoS.	System-defined policy
□	⌄	AAD ReadOnlyAccess Read-only permissions for Advanced Anti-DDoS.	System-defined policy
□	⌄	ADN FullAccess Full permissions for Application Delivery Network.	System-defined policy
□	⌄	ADN Instance Operations Instance operation permissions for Application Delivery Network.	System-defined policy
□	⌄	ADN ReadOnlyAccess Read-only permissions for Application Delivery Network.	System-defined policy
□	⌄	AOM FullAccess AOM Access All	System-defined policy
□	⌄	AOM ReadOnlyAccess AOM read only	System-defined policy
□	⌄	APIG Administrator	System-defined role

6. On the **Select Scope** page, select **All Resources**, and click **OK**.

Method 2: Grant the Security Administrator permissions to the user group to which the IAM user belongs.

1. Log in to the **IAM console**, choose **User Groups** in the navigation pane, and click the target user group.
2. Click the **Permissions** tab, search for **Security Administrator**, select **Security Administrator**, and click **Next**.



Assign selected permissions to cts.

□	Policy/Role Name	Type
□	DBSS Security Administrator DBSS Security Administrator has the authority to set the security policy of DBSS	System-defined role
□	Security Administrator Full permissions for Identity and Access Management. This role does not have permissions for switching roles.	System-defined role

Note: Users with the Security Administrator permissions can perform all actions such as agency, role, and user management. Exercise caution when assigning the permissions.

3. On the **Select Scope** page, select **All Resources**, and click **OK**.

2.8 Which Resource Space Will Be Set As Default on the IoT Platform?

If you enabled IoTDA before 00:00 on April 27, 2020 and your account has multiple resource spaces, the platform sets the following resource space as default:

1. Resource space whose name contains the **Default** word.
2. Resource space with the largest number of devices if there is no resource space whose name contains the **Default** word.

2.9 How Does IoTDA Obtain Device Data?

IoTDA obtains device data and forwards the data to other services through rules. For example, it can forward data to OBS for storage, or to DIS or DMS, which then further forwards the data to an application.

2.10 Is There Any Limitation on the Number of Resource Spaces and Devices I Can Add on the IoT Platform?

Yes, there is. For details about the limitations, see **Limitations**.

2.11 Does the IoTDA Support Device Registration in Batches?

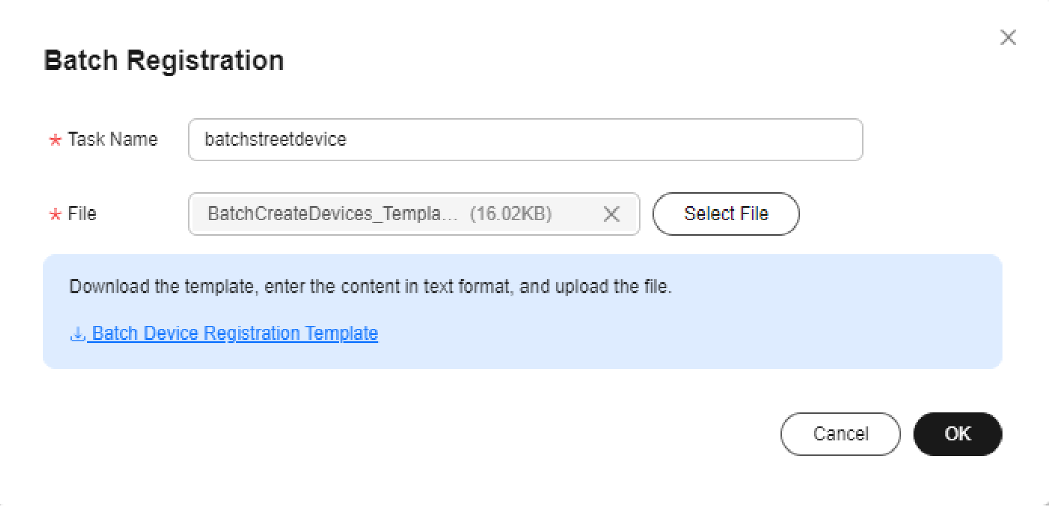
IoTDA allows an application to call the API **Creating a Batch Task** to register a batch of devices. Alternatively, you can perform batch registration on the IoTDA console. This topic describes how to use the IoTDA console to register a batch of devices.

Step 1 Visit the **IoTDA** product page and click **Access Console**.

Step 2 Click the target instance. In the navigation pane, choose **Devices** > **All Devices**.

Step 3 Click the **Batch Registration** tab, click **Batch Register**, specify **Task Name**, add a file, and click **OK**.

Figure 2-5 Device - Registering devices in batches



Batch Registration

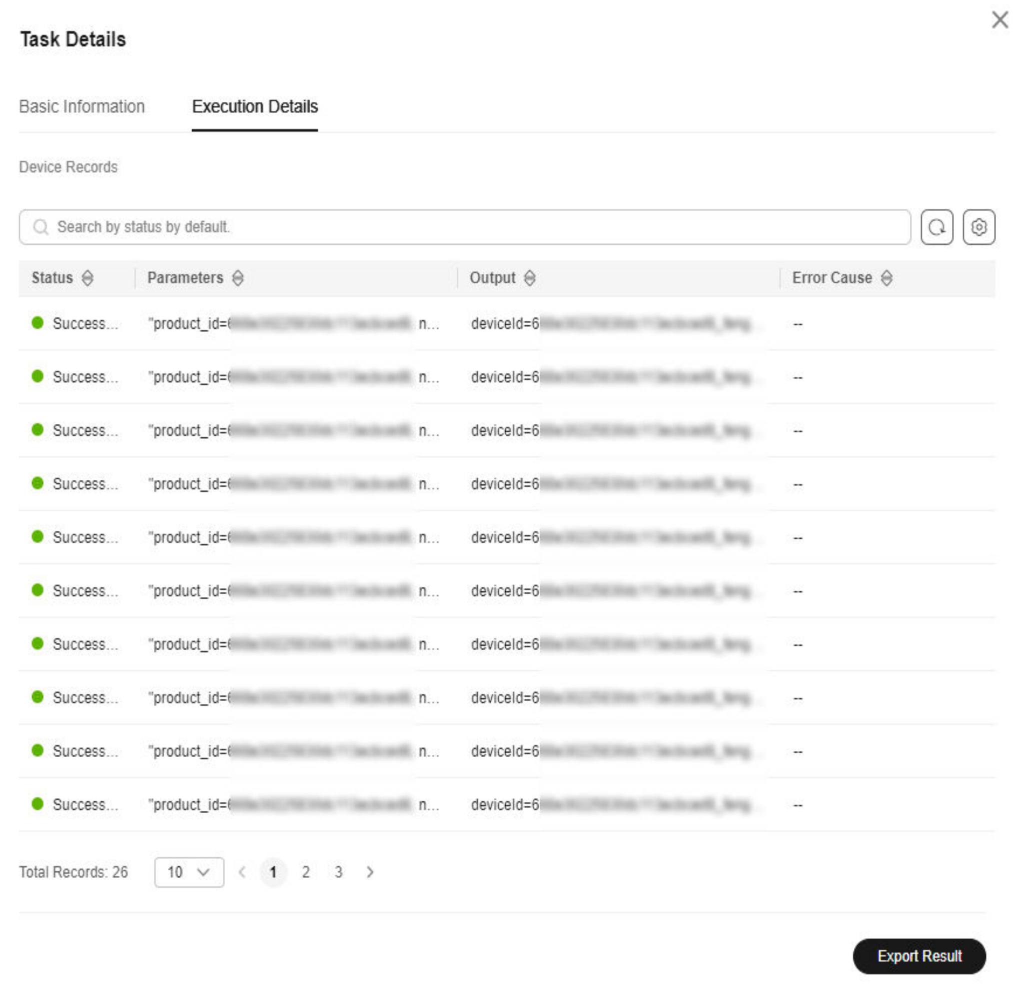
* Task Name: batchstreetdevice

* File: BatchCreateDevices_Templa... (16.02KB)

Download the template, enter the content in text format, and upload the file.

Step 4 If the devices use the native MQTT protocol, click the batch task registration record to open the task execution details, and save the device IDs and secrets generated, which will be used for device access.

Figure 2-6 Batch device registering - Execution details



Task Details

Device Records

Status	Parameters	Output	Error Cause
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--
● Success...	"product_id=[redacted] n...	deviceId=6[redacted]	--

Total Records: 26

----End

2.12 Are There Any Limitations on the Use of the IoT Platform?

Yes. For details about the limitations, see **Limitations**.

2.13 What DTLS Encryption Algorithms Are Supported by the IoT Platform?

The platform allows the use of DTLS with pre-shared keys. The following encryption suites are supported:

- TLS_PSK_WITH_AES_128_CCM_8, as defined in [RFC6655]
- TLS_PSK_WITH_AES_128_CBC_SHA256, as defined in [RFC5487]

2.14 Does the IoT Platform Support Conversion Between Big Endian and Little Endian for Binary Data?

No. The platform supports only big endian.

2.15 What Is NB-IoT?

Narrowband Internet of Things (NB-IoT) is an important part of the IoT. NB-IoT enables a wide range of devices and services to be connected using cellular telecommunications bands. It only consumes about 180 kHz of bandwidth, hence the name, "narrowband". NB-IoT technology can be deployed on GSM, UMTS, or LTE networks, which keeps costs down and makes it easier to upgrade smoothly.

NB-IoT is a new technology in the IoT field. It supports cellular data connection of low-power devices in a wide area network (WAN), so it is also called low-power wide area network (LPWAN). NB-IoT supports efficient connection of devices with long standby time and high network connection requirements. The battery of an NB-IoT device is expected to last at least 10 years. NB-IoT devices are also capable of providing complete indoor coverage of cellular networks.

Among the use cases of NB-IoT are smart locks, smart cities, smart water and gas meters, smart trackers, smart warehousing, and smart street lamps. In these scenarios, raw data is sent to the platform, which then uses NB-IoT modules to process the data and send it along for analysis and other uses.

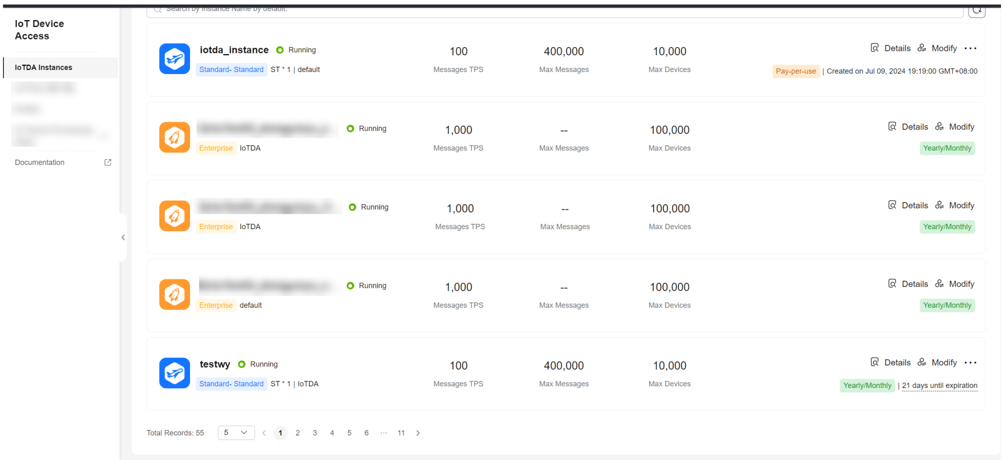
2.16 What Are the Components of the IoT Platform and What Hardware Architectures Does It Support?

The platform is built on Huawei EulerOS, Huawei-developed Java compilation and running environment, Huawei GaussDB, and open source MongoDB. It supports multiple hardware architectures, including x86 and ARM, to accommodate different scenarios.

2.17 How Do I Obtain the Platform Access Address?

1. Log in to the IoTDA console, choose **IoTDA Instances** in the navigation pane, and click the target instance card.

Figure 2-7 Instance management - Changing instance

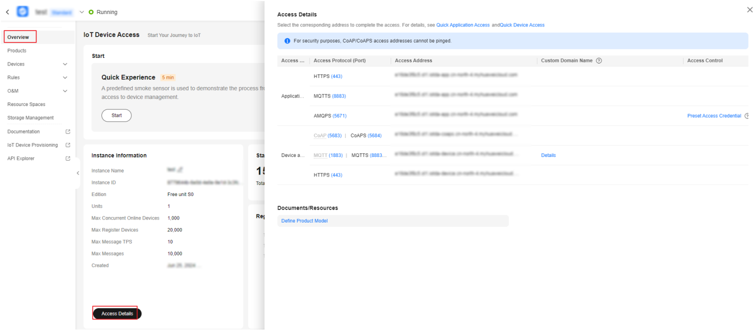


Instance	Messages TPS	Max Messages	Max Devices	
iotda_instance Running Standard- Standard ST * 1 | default	100	400,000	10,000	Details Modify Pay-per-use | Created on Jul 09, 2024 19:19:00 GMT+08:00
[redacted] Running Enterprise IoTDA	1,000	--	100,000	Details Modify Yearly/Monthly
[redacted] Running Enterprise IoTDA	1,000	--	100,000	Details Modify Yearly/Monthly
[redacted] Running Enterprise default	1,000	--	100,000	Details Modify Yearly/Monthly
testwy Running Standard- Standard ST * 1 | IoTDA	100	400,000	10,000	Details Modify Yearly/Monthly | 21 days until expiration

Total Records: 55 5 < 1 2 3 4 5 6 ··· 11 >

2. In the navigation pane, choose **Overview**. Click **Access Details** to check your platform access address.

Figure 2-8 Obtaining access information



A predefined smoke sensor is used to demonstrate the process fr
access to device management.

Instance Information

Instance Name	[redacted]
Instance ID	[redacted]
Edition	Free unit S0
Units	1
Max Concurrent Online Devices	1,000
Max Register Devices	20,000
Max Message TPS	10
Max Messages	10,000
Created	[redacted]

Access Details

Select the corresponding address to complete the access. For details, see Quick Application Access andQuick Device Access

For security purposes, CoAP/CoAPS access addresses cannot be pinged.

Access ...	Access Protocol (Port)	Access Address	Custom Domain Name	Access Control
Applicati...	HTTPS (443)	[redacted]		
	MQTTS (8883)	[redacted]		
	AMQPS (5671)	[redacted]		Preset Access Credential
Device a...	CoAP (5683) | CoAPS (5684)	[redacted]		
	MQTT (1883) | MQTTS (8883...	[redacted]	Details	
	HTTPS (443)	[redacted]		

Documents/Resources

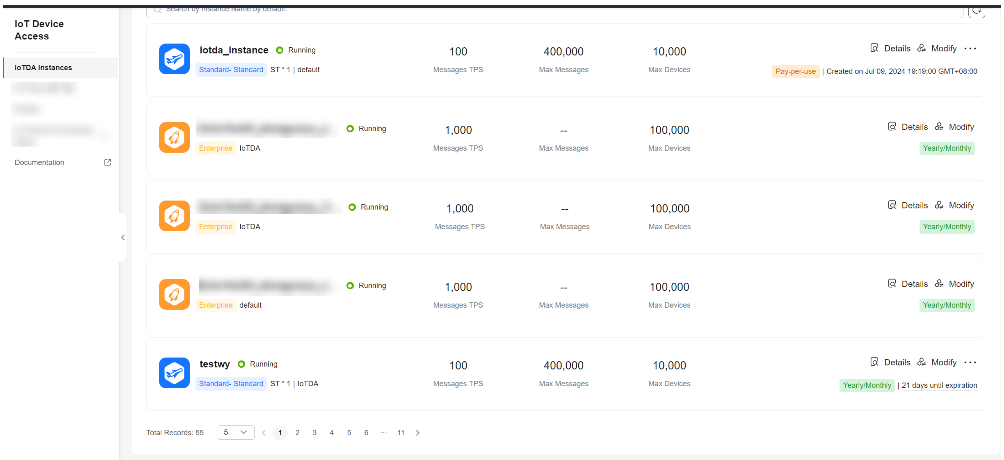
Define Product Model

3 Device Integration

How Do I Obtain the Device Access Address?

1. Log in to the **IoTDA console**. In the navigation pane, choose **IoTDA Instances**. Click the target instance card.

Figure 3-1 Instance management - Changing instance

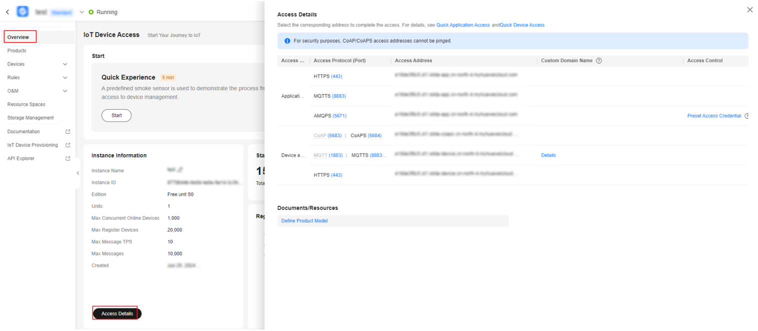


Instance	Messages TPS	Max Messages	Max Devices	
iotda_instance ○ Running Standard- Standard ST * 1 | default	100	400,000	10,000	Details Modify ··· Pay-per-use | Created on Jul 09, 2024 19:19:00 GMT+08:00
[redacted] ○ Running Enterprise IoTDA	1,000	--	100,000	Details Modify Yearly/Monthly
[redacted] ○ Running Enterprise IoTDA	1,000	--	100,000	Details Modify Yearly/Monthly
[redacted] ○ Running Enterprise default	1,000	--	100,000	Details Modify Yearly/Monthly
testwy ○ Running Standard- Standard ST * 1 | IoTDA	100	400,000	10,000	Details Modify ··· Yearly/Monthly | 21 days until expiration

Total Records: 55 5 < 1 2 3 4 5 6 ··· 11 >

2. In the navigation pane, choose **Overview**. Click **Access Details** to check your platform access address.

Figure 3-2 Obtaining access information



A predefined smoke sensor is used to demonstrate the process fr access to device management.

Instance Information

Instance Name	[redacted]
Instance ID	[redacted]
Edition	Free unit S0
Units	1
Max Concurrent Online Devices	1,000
Max Register Devices	20,000
Max Message TPS	10
Max Messages	10,000
Created	[redacted]

Access Details

Select the corresponding address to complete the access. For details, see Quick Application Access andQuick Device Access

For security purposes, CoAP/CoAPS access addresses cannot be pinged.

Access ...	Access Protocol (Port)	Access Address	Custom Domain Name	Access Control
Applicati...	HTTPS (443)	[redacted]		
	MQTTS (8883)	[redacted]		
	AMQPS (5671)	[redacted]		Preset Access Credential
Device a...	CoAP (5683) | CoAPS (5684)	[redacted]		
	MQTT (1883) | MQTTS (8883...	[redacted]	Details	
	HTTPS (443)	[redacted]		

Documents/Resources

Define Product Model

Must the X.509 Certificate I Use for Device Connection Be Issued by an Authoritative Organization?

Not necessary. A custom certificate is also supported, but you are advised to use a certificate issued by an authoritative organization. For details, see **Uploading a Device CA Certificate**.

How Do I Locate the Cause of the Certificate-based Device Connection Failure?

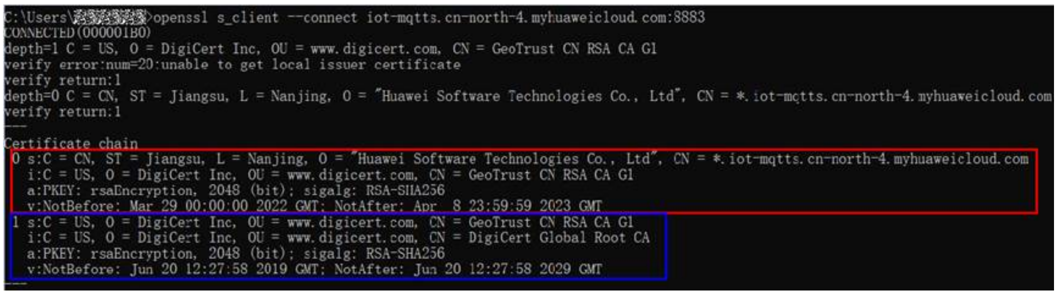
Step 1 Check whether the platform CA entered on the device side is correct.

Platform certificate verification fails on the device side if the following stack information is displayed: **Caused by: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target**. In this case, check whether the platform CA entered on the device side is correct.

Use OpenSSL to run the following command to obtain server certificate information:

openssl s_client --connect <brokerAddress:brokerPort>

Figure 3-3 OpenSSL execution



```
C:\Users\[redacted]>openssl s_client --connect iot-mqtts.cn-north-4.myhuaweicloud.com:8883
CONNECTED(000001B0)
depth=1 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust CN RSA CA G1
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 C = CN, ST = Jiangsu, L = Nanjing, O = "Huawei Software Technologies Co., Ltd", CN = *.iot-mqtts.cn-north-4.myhuaweicloud.com
verify return:1
---
Certificate chain
 0 s:C = CN, ST = Jiangsu, L = Nanjing, O = "Huawei Software Technologies Co., Ltd", CN = *.iot-mqtts.cn-north-4.myhuaweicloud.com
   i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust CN RSA CA G1
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Mar 29 00:00:00 2022 GMT; NotAfter: Apr  8 23:59:59 2023 GMT
 1 s:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = GeoTrust CN RSA CA G1
   i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Jun 20 12:27:58 2019 GMT; NotAfter: Jun 20 12:27:58 2029 GMT
---
```

The obtained certificate chain contains the following certificates:

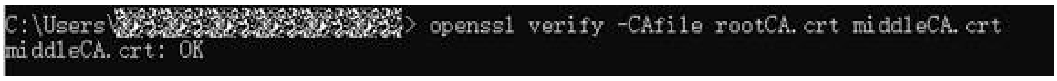
- Certificate 0 is the platform certificate that identifies the platform.
- Certificate 1 is the intermediate CA that issues the platform certificate.

When verifying the platform identity, a device needs to use the intermediate CA issuer to verify the certificate chain.

Check whether the user information of the platform root CA entered on the device side is consistent the issuer information of the intermediate CA. If consistent, save the intermediate CA displayed in the CLI as a file and run the following command to verify the issuing relationship between the root CA and the intermediate CA:

openssl verify -verbose -CAfile <CAFile> <middleCAFile>

Figure 3-4 OpenSSL certificate verification



```
C:\Users\[redacted]> openssl verify -CAfile rootCA.crt middleCA.crt
middleCA.crt: OK
```

Step 2 Check whether the device certificate entered on the device side matches the private key of the device certificate.

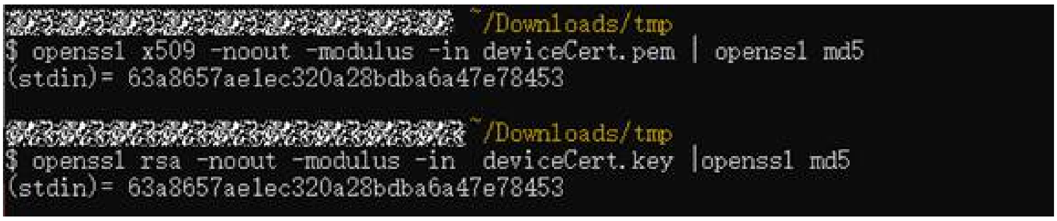
Run the following commands to extract the MD5 hash values of the certificate and private key file:

openssl x509 -noout -modulus -in *<certificate file>* **| openssl md5**

openssl rsa -noout -modulus -in *<private key file>* **| openssl md5**

If the two MD5 hash values are different, the certificate and private key do not match. You are advised to enter the correct device certificate and private key.

Figure 3-5 Comparing MD5 values



```
[redacted] ~/Downloads/tmp
$ openssl x509 -noout -modulus -in deviceCert.pem | openssl md5
(stdin)= 63a8657ae1ec320a28bdba6a47e78453

[redacted] ~/Downloads/tmp
$ openssl rsa -noout -modulus -in  deviceCert.key |openssl md5
(stdin)= 63a8657ae1ec320a28bdba6a47e78453
```

Step 3 Check the binding relationship between the device and the certificate fingerprint.

Enable message tracing for the device and locate the fault based on the traced messages.

----End

How Do I Set the Device Name?

- Set the name of a device when you add the device to the IoTDA console.
- Set the device name when you use the API for registering or creating a device.

How Do I Activate IoT Devices?

When a device is registered on the IoT platform, and gets connected to the platform or starts to report data to the platform, the device is considered to be activated. For details, see **Device Connection Authentication**.

How Is the Status of a Device Changed?

An NB-IoT device is considered online when it reports data to the IoT platform. If no data is reported within 25 hours since the last data reporting, the device status changes to abnormal. If no data is reported within 49 hours, the device status changes to offline.

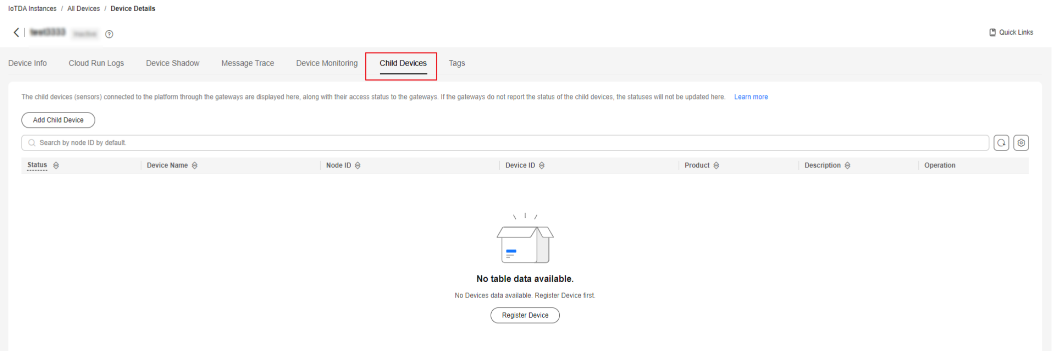
An MQTT device is considered online when it is connected to the platform. If the device is disconnected from the platform, it is considered to be offline and its status will be updated on the platform within a minute. You can click the status refresh button to update the status immediately.

For details, see **Device Management**.

How Can I Check the Details of Gateways and Child Devices?

Log in to the IoTDA console, choose **Devices** > **All Devices** in the navigation pane, and click **View** in the row where a gateway is located. Check the gateway details on the **Device Info** tab page, and check its child devices on the **Child Devices** tab page. For details, see **Gateways and Child Devices**.

Figure 3-6 Device - Viewing child devices



The child devices (sensors) connected to the platform through the gateways are displayed here, along with their access status to the gateways. If the gateways do not report the status of the child devices, the statuses will not be updated here. Learn more

Status	Device Name	Node ID	Device ID	Product	Description	Operation

No table data available.

No Devices data available. Register Device first.

Why Is the Status of Child Devices Displayed as Online When the Gateway Is Offline?

The status of child devices is managed by the gateway. When the gateway is offline, it can call the API for **Gateway Updating Child Device Status** to send the latest status of child devices to the IoT platform.

What Are the Differences Between Device ID, Node ID, and IMEI and What Are They Used For?

On the IoT platform, you need to enter a node ID (**nodeId**) when registering a device. A node ID is a physical identifier of a device. Generally, an IMEI or MAC address is used. A device ID (**deviceId**) is a logical identifier of a device on the IoT platform.

- NB-IoT devices: The node ID is carried for access authentication when a device connects to the platform.
- MQTT devices: A device uses the secret and device ID to connect to the platform. The access authentication is the one-device-one-secret mode.

For details, see **Device Authentication**.

Can a Device Send Files to the IoT Platform?

Yes. For details about the sending process, see **File Uploads**.

What Do I Do If the IoT Link Plug-in Download Fails When Using the BearPi Development Board?

Download Visual Studio Code 1.49 that matches your computer configuration and install it. Other versions do not support IoT Link.

What Do I Do If Device Activation Fails When Using the BearPi Development Board?

Enter **AT+CGATT?** and click **Send**. If **+CGATT:1** is returned, the network attach is successful, indicating that the NB-IoT network is normal. If **+CGATT:0** is returned, the network attach fails. Check whether the SIM card is correctly inserted, contact the carrier to check the network status, or check whether the IMEI is the same as that entered during device registration on the platform. You can obtain the IMEI by setting the dialing test switch to the **AT-PC** mode, selecting the STM port, setting the baud rate to 9600, and then running the **AT+CGSN=1** command.

What Do I Do If Data Reporting or Command Receiving Fails When Using the BearPi Development Board?

Obtain code from **Developing a Smart Street Light Using NB-IoT BearPi** or **Developing a Smart Smoke Detector Using NB-IoT BearPi**. If the code is obtained from other channels, contact technical support of the channel. You can also post your questions about BearPi on the forum.

What Do I Do If a Message is Displayed Indicating that the Device is Occupied During Device Registration?

For an MQTT device, check whether the device ID is unique under your account. For an LwM2M device, check whether the node ID is unique under your account. If the device/node ID is unique, submit a service ticket on the console.

4 IoT Device SDKs

Why Did the IoT Device SDK for C Fail to Be Started?

It is possible that the OpenSSL or Paho library files failed to be compiled or the **export LD_LIBRARY_PATH=./lib/** command was not used to load the library files. For details, see "Preparations" in **IoT Device SDK (C) Development Guide**.

Why Did the IoT Device SDK for Java Fail to Be Started?

It is because the JDK (1.8 or later) or Maven has not been installed.

Which Java SDK Demo Should I Refer to?

device_demo is recommended if you connect devices directly to the IoT platform and want to integrate the bootstrap (device provisioning) function, whereas **gateway_demo** is a better choice if you connect devices to the platform through common gateways or gateways that support generic protocols.

Which C SDK Demo Should I Refer to?

device demo is recommended if you connect devices directly to the IoT platform, whereas **bootstrap_demo** is recommended if you want to integrate the bootstrap function. **gateway demo** will be a better choice if you connect devices to the platform through common gateways or gateways that support generic protocols.

Why Was the Error Code 4 Returned When I Used the IoT Device SDK for C for Device Connection?

It is because the entered account name or password was incorrect. For details about the error codes returned upon connection failures, see the error code description in the **MQTTAsyn.h** file.

What Are the Differences Between the IoT Device SDK and IoT Device SDK Tiny?

IoT Device SDK Tiny is more lightweight than IoT Device SDK and suitable for devices with smaller memory and disk space and few child devices mounted. The dynamic link library is not used during compilation. The code provides the OS abstraction layer to adapt to different OSs, such as FreeRTOS, Linux, Nova OS, μC/OS-II, and OpenHarmony LiteOS-M. IoT Device SDK Tiny supports MQTT(S), LwM2M, and CoAP. It uses Mbed TLS for encryption, while IoT Device SDK C uses OpenSSL. For details, see **Introduction to IoT Device SDKs**.

Issues Related to IoT Device SDK C Tiny

1. **What Do I Do If "mqtt_imp_init: ###please implement mqtt by yourself####" Is Displayed in Logs?**

   Check whether the compilation architecture supports the **__attribute__ ((weak))** function. If not, comment out all these functions. If **link_tcpip_imp_init:###please implement this function by yourself####** is displayed, check whether the network layer adaptation is implemented.

2. **What Do I Do If the Task Execution Sequence Is Inconsistent with the Task Priority When Using the SDK?**

   By default, task priorities in the SDK range from 0 to 31 in descending order. You can adjust task priorities based on the OS.

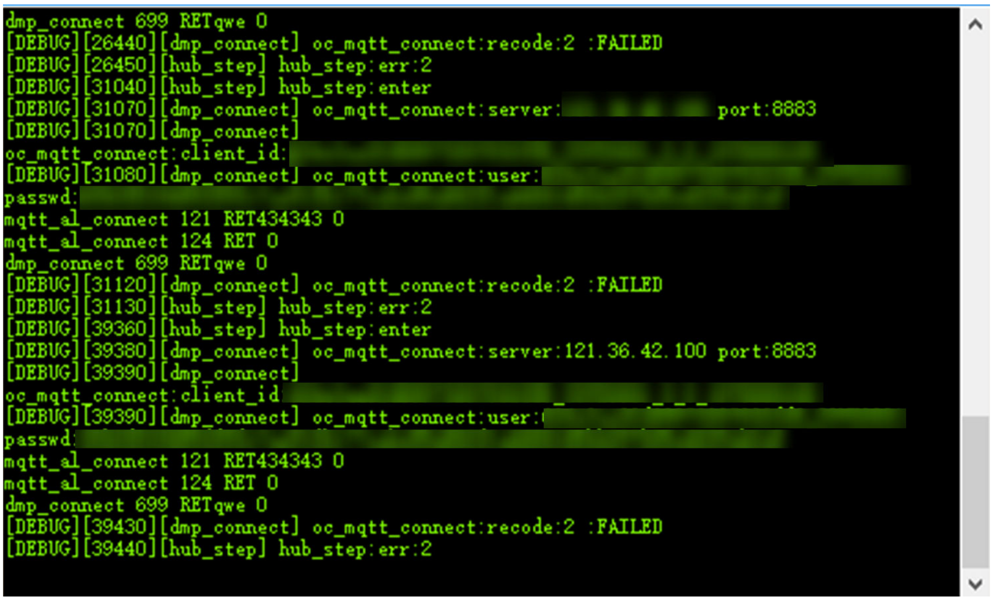
3. **What Steps Are Included in the SDK Porting Process?**

   The porting process includes registering an OS with the OS abstraction layer (OSAL) and registering TCP/IP with the service abstraction layer (SAL). In addition, you can perform modular tailoring as required. For details, see **Developer Guide**.

4. **What Do I Do If Error Code 2 Is Returned When MQTT Is Used to Connect to Huawei Cloud?**

   The following figure shows the logs.

   Figure 4-1 MQTT connection error

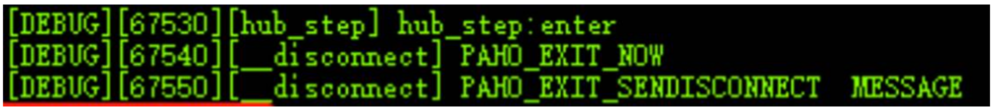


```
dmp_connect 699 RETqwe 0
[DEBUG][26440][dmp_connect] oc_mqtt_connect:recode:2 :FAILED
[DEBUG][26450][hub_step] hub_step:err:2
[DEBUG][31040][hub_step] hub_step:enter
[DEBUG][31070][dmp_connect] oc_mqtt_connect:server:[redacted] port:8883
[DEBUG][31070][dmp_connect]
oc_mqtt_connect:client_id:[redacted]
[DEBUG][31080][dmp_connect] oc_mqtt_connect:user:[redacted]
passwd:[redacted]
mqtt_al_connect 121 RET434343 0
mqtt_al_connect 124 RET 0
dmp_connect 699 RETqwe 0
[DEBUG][31120][dmp_connect] oc_mqtt_connect:recode:2 :FAILED
[DEBUG][31130][hub_step] hub_step:err:2
[DEBUG][39360][hub_step] hub_step:enter
[DEBUG][39380][dmp_connect] oc_mqtt_connect:server:121.36.42.100 port:8883
[DEBUG][39390][dmp_connect]
oc_mqtt_connect:client_id[redacted]
[DEBUG][39390][dmp_connect] oc_mqtt_connect:user:[redacted]
passwd:[redacted]
mqtt_al_connect 121 RET434343 0
mqtt_al_connect 124 RET 0
dmp_connect 699 RETqwe 0
[DEBUG][39430][dmp_connect] oc_mqtt_connect:recode:2 :FAILED
[DEBUG][39440][hub_step] hub_step:err:2
```

   This is caused by a network error. Check whether your development board is connected to the network. If connected, check whether the IP address, domain name, and port number of the platform to be connected are correct. In the Linux environment, ping the platform address to check whether the address can be pinged.

5. **What Do I Do If a Device Does not Automatically Reconnect to Huawei Cloud After the Disconnected Network is Restored?**

Figure 4-2 No reconnection

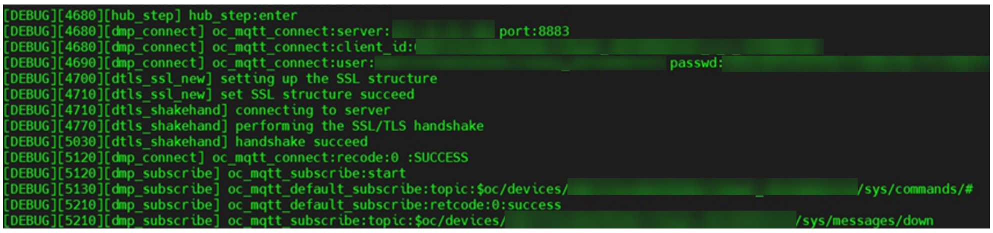


```
[DEBUG][67530][hub_step] hub_step:enter
[DEBUG][67540][__disconnect] PAHO_EXIT_NOW
[DEBUG][67550][__disconnect] PAHO_EXIT_SENDISCONNECT  MESSAGE
```

The log shows that Paho has exited and the device has been disconnected from the platform and goes offline. Change the sleep time in the **__loop_entry()** function in the **network\mqtt\paho_mqtt\port\paho_mqtt_port.c** file in the SDK directory from 1 ms to 100 ms (**osal_task_sleep(100)**). Then, check whether automatic reconnection is successful.

6. **What Do I Do If I Am Stuck in the Topic Subscription Process When MQTTS Is Used to Connect to Huawei Cloud?**

Figure 4-3 Suspended topic subscription process



```
[DEBUG][4680][hub_step] hub_step:enter
[DEBUG][4680][dmp_connect] oc_mqtt_connect:server:[redacted] port:8883
[DEBUG][4680][dmp_connect] oc_mqtt_connect:client_id:[redacted]
[DEBUG][4690][dmp_connect] oc_mqtt_connect:user:[redacted] passwd:[redacted]
[DEBUG][4700][dtls_ssl_new] setting up the SSL structure
[DEBUG][4710][dtls_ssl_new] set SSL structure succeed
[DEBUG][4710][dtls_shakehand] connecting to server
[DEBUG][4770][dtls_shakehand] performing the SSL/TLS handshake
[DEBUG][5030][dtls_shakehand] handshake succeed
[DEBUG][5120][dmp_connect] oc_mqtt_connect:recode:0 :SUCCESS
[DEBUG][5120][dmp_subscribe] oc_mqtt_subscribe:start
[DEBUG][5130][dmp_subscribe] oc_mqtt_default_subscribe:topic:$oc/devices/[redacted]/sys/commands/#
[DEBUG][5210][dmp_subscribe] oc_mqtt_default_subscribe:retcode:0:success
[DEBUG][5210][dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/[redacted]/sys/messages/down
```

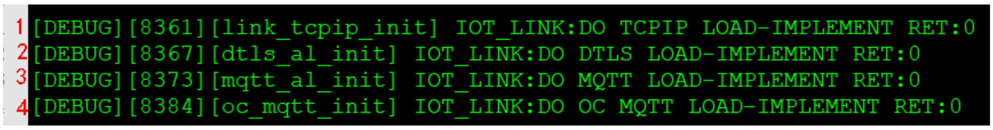
Change the value of **CONFIG_PAHO_LOOPTIMEOUT** in the **iot_config.h** file to **1000**.

7. **What Do I Do If a Developer Board Successfully Connects to Huawei Cloud Using MQTT But Connection Fails When MQTTS Is Used?**

The cause may be that the development board memory is insufficient. Check the remaining memory, which should be greater than 60 KB. For OpenHarmony L0 devices, call **LOS_MemPoolSizeGet(m_aucSysMem0)** to obtain the total memory and **LOS_MemTotalUsedGet(m_aucSysMem0)** to obtain the used memory. Then, you can get the remaining memory. If Shell is ported, you can run the **free** command to obtain the memory details.

8. **How Do I Interpret the Following Log Content Generated When MQTT Is Used to Connect to Huawei Cloud?**

Figure 4-4 Normal MQTT connection setup



```
1 [DEBUG][8361][link_tcpip_init] IOT_LINK:DO TCPIP LOAD-IMPLEMENT RET:0
2 [DEBUG][8367][dtls_al_init] IOT_LINK:DO DTLS LOAD-IMPLEMENT RET:0
3 [DEBUG][8373][mqtt_al_init] IOT_LINK:DO MQTT LOAD-IMPLEMENT RET:0
4 [DEBUG][8384][oc_mqtt_init] IOT_LINK:DO OC MQTT LOAD-IMPLEMENT RET:0
```

Lines 3 and 4 indicate that MQTT is used for connection. If MQTTS is used, line 2 will be displayed. If TCP/IP that adapts to SAL is used, line 1 will be displayed. If the corresponding log content is not displayed, find the macro of the initialization function of the corresponding function based on the

link_main.c file, and then check whether the macro is enabled in **iotlink_config.h**.

9. **Why a Device Is Powered Off But Displayed as Online on IoTDA?**

   If a device does not proactively disconnect from IoTDA, the device disconnection time is related to the MQTT **lifetime** in the code, which is 1.5 times the heartbeat time. When establishing an MQTT connection, set the **lifetime** parameter.

10. **List of Devices with IoT Device SDK Tiny (Code Attached) Ported**

    See **Porting Device List**.

What Do I Do If the Error Message "Too many publishes in progress" Is Displayed When the IoT Device SDK (Java) Is used to Report Data?

There are too many concurrent messages. To solve this problem, increase the value of **MAX_FLIGHT_COUNT** in **MqttConnection.java**.

5 LwM2M/CoAP Device Access

How Do I Connect Devices to the IoT Platform Through LwM2M over CoAP Protocols?

1. Development on the platform: Create products, develop product models and codecs on the platform, and register devices. For details, see **Creating a Product**, **Developing a Product Model**, **Developing a Codec**, and **Registering a Device**.
2. Development on the device: Use modules and Tiny SDKs on the device side for access. For details, see **IoT Device SDK Tiny (C) User Guide**.
3. (Optional) Develop applications.

How Do I Know the Strength of the NB-IoT Network Signal?

Run the **AT+CSQ** command to query the NB-IoT signal strength.

The returned value is **+CSQ. <rssi>,<ber>**

A larger value of **rssi** indicates a stronger signal. The formula for calculating the signal strength is as follows: Signal strength = 113 dBm + (rssi x 2)

- rssi = 0: The signal is poor.
- rssi = 31: The signal is very strong.
- rssi = 99: There is no signal.
- The **ber** field is not used. It is fixed at 99.

If there is no signal or the signal is poor, contact the carrier.

Why Did the NB-IoT Module Not Attach to the Network?

1. Run the **AT+NUESTATS** command to check whether there is a network signal.
2. If the value of **Signal power** is **0**, there is no network signal. Check whether the frequency band corresponding to the base station has been released, or move the device to somewhere where the signal is stronger and try again.
3. Run the **AT+NBAND?** command to check whether the configured frequency band is the same as that of the module.

What Do I Do If an NB-IoT Module Failed to Be Bound to a Device?

When the NB-IoT module and real NB-IoT network are used to access the IoT platform, the first step is to bind a device.

If the device fails to be bound, you can troubleshoot the error by checking the following items:

- **When you register the device on the platform, are the values of nodeId correct? Is the timeout period too short?**

  The values of **nodeId** must be the IMEI of the NB-IoT module. In addition, the timeout period should be long enough, so that the device can send a binding request to the platform within the period after the registration is successful.

- **Is the product information used during device registration the same as that in the product model?**

  If you register the device in the IoTDA console, ensure that the correct product model is selected. If you call an API to register the device, make sure the value of **deviceInfo** is the same as that defined in the profile.

- **Is the signal from an NB-IoT base station reaching the module?**

  Run the **AT+CSQ?** command to check the NB-IoT signal strength. If there is no signal or the signal strength is weak, contact the carrier.

- **Can the NB-IoT module be attached to the network?**

  Run the **AT+CEREG?** command to obtain the network registration details. If the returned status is unregistered or rejected, contact the carrier. The version of the NB-IoT module may not match the version of the carrier's base station.

- **Can the NB-IoT module ping the Huawei Cloud IoT platform?**

  Run the **AT+NPING** command to ping the Huawei Cloud IoT platform. If the platform cannot be pinged, it indicates that the carrier's network cannot reach the public network. Contact the carrier to check if its core network is connected to the public network and if the core network can be connected to only the carrier's IoT platform. Work with the carrier on the connection to the public network.

- **Are the domain name and port of the platform correctly set for the NB-IoT module?**

  Run the **AT+NCDP** command to set the domain name and port of the platform. To obtain the domain name and port, log in to the IoTDA console and check the connection details of CoAP or CoAPs devices.

- **Does the AT command sent to the NB-IoT module end with \r\n?**

  Each command sent to the NB-IoT module must end with **\r\n**. If not so, the command is cached in the NB-IoT module.

- **Is the transmitted data in the "SENT" status in the NB-IoT module?**

  Run the **AT+NQMGS** command to check the status of the commands sent.

  PENDING: The data was sent but the platform has not responded.

  SENT: The data was sent and the platform has responded.

  ERROR: Data reporting is abnormal.

  If the status is **PENDING** or **ERROR**, there may be network issues. Check the base station and core network.

- **Can the AT+NMGS data sent by the NB-IoT module be parsed properly?**

  Use the **codec test tool** to check whether the streams to be sent can be parsed.

What Can I Do If an NB-IoT Module Cannot Report Data?

Ensure that the NB-IoT module has been bound to a device. The binding is performed when the NB-IoT module reports the first piece of data to the IoT platform. If the binding fails, the device is not activated on the platform. In this case, resolve the fault by referring to **What Do I Do If an NB-IoT Module Failed to Bind to a Device**.

Let's assume that the device binding is successful and the device is displayed as online on the platform. Check the following items:

- **Does the AT+NMGS command sent to the NB-IoT module end with \r\n?**

  Each command sent to the NB-IoT module must end with **\r\n**. If not so, the command is cached in the NB-IoT module.
- **Can the payload of the sent AT+NMGS be parsed by the codec?**

  Use the **codec test tool** to check the payload in the code stream to be sent. Check whether the message structure is correct and complies with the definition in the product model.

Why Was a 513 Message Reported During the Connection of an NB-IoT Device?

After being powered on, devices initiate a TUP registration flow to the IoT platform. TUP is a Huawei proprietary protocol over CoAP. It is similar to LwM2M. The TUP registration flow on HiSilicon chipsets cannot exceed 4 seconds. If the TUP registration is not completed within 4 seconds, a 513 message is reported.

When a 513 message is reported, do as follows:

1. If a 513 message is reported due to poor network performance, contact the NB-IoT network carrier to check the network status.
2. You are advised to restart the device and run **AT+NMGS**. When you send service data by running **AT+NMGS**, registration is triggered. If the subscription to the t/d resources (resources for receiving and sending service data) is not received within 4 seconds, an error is returned, but the registration is continued based on the CoAP-layer retransmission. Only when the subscription to the t/d resources is not received within 160 seconds, will the registration fail. The 160 seconds is sufficient for device registration. If an error message is returned after 4 seconds, only the data of the first packet is discarded. You are advised to restart the device and run **AT+NMGS** to trigger the registration.

   You can run **AT+NMSTATUS?** to query registration status. If **+NMSTATUS:MO_DATA_ENABLED** is returned, the registration is successful.

Why Does Data Reporting Fails When an NB-IoT Card Is Used in Another Device?

Some carriers' cards are bound to devices. If a device is changed, the card bound to the device may be unavailable. In this case, contact the carrier.

What Can I Do If I Cannot Connect a Registered NB-IoT Device to the IoT Platform?

- Run the **AT+CGATT=1** command in the module and see if there is an error reported. If there is, do as follows:
  - Contact the NB-IoT network carrier to check if the NB-IoT card works properly.
  - Contact the module manufacturer to check if the model works properly.
- If no error is reported, check if the IP address and port number of the platform are correct.

NOTE

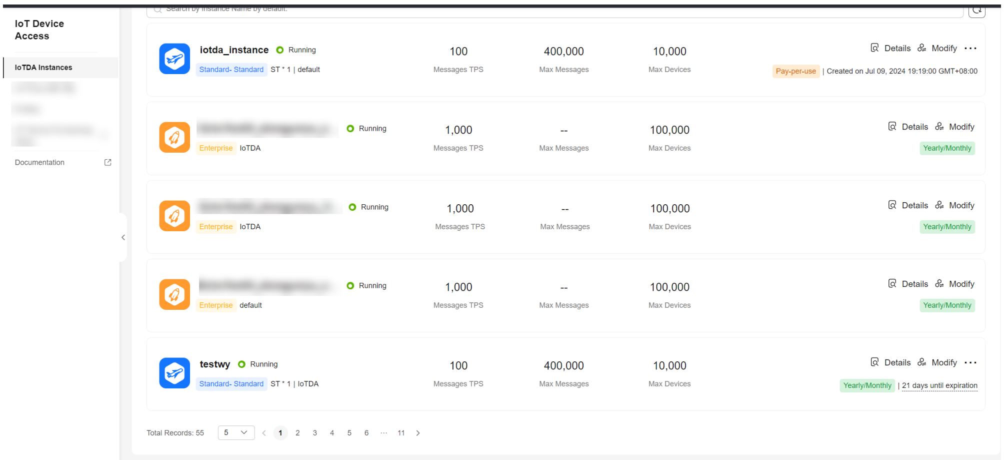
You can obtain the IP address and port number from the platform service provider. Port 5683 is used for unencrypted access, whereas port 5684 is used for encrypted access.

6 MQTT-based Device Access

What Are the Domain Name and Port Number for MQTT-based Device Access?

1. Log in to the **IoTDA console**. In the navigation pane, choose **IoTDA Instances**. Click the target instance card.

Figure 6-1 Instance management - Changing instance



iotda_instance Running — Standard- Standard ST * 1 | default — 100 Messages TPS — 400,000 Max Messages — 10,000 Max Devices — Pay-per-use | Created on Jul 09, 2024 19:19:00 GMT+08:00

[redacted] Running — Enterprise IoTDA — 1,000 Messages TPS — -- Max Messages — 100,000 Max Devices — Yearly/Monthly

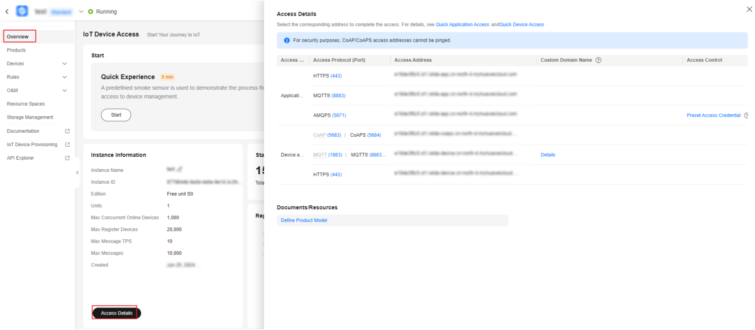
[redacted] Running — Enterprise IoTDA — 1,000 Messages TPS — -- Max Messages — 100,000 Max Devices — Yearly/Monthly

[redacted] Running — Enterprise default — 1,000 Messages TPS — -- Max Messages — 100,000 Max Devices — Yearly/Monthly

testwy Running — Standard- Standard ST * 1 | IoTDA — 100 Messages TPS — 400,000 Max Messages — 10,000 Max Devices — Yearly/Monthly | 21 days until expiration

2. In the navigation pane, choose **Overview**. Click **Access Details** to check your platform access address.

Figure 6-2 Obtaining access information



A predefined smoke sensor is used to demonstrate the process fr access to device management.

Instance Information

Instance Name	[redacted]
Instance ID	[redacted]
Edition	Free unit S0
Units	1
Max Concurrent Online Devices	1,000
Max Register Devices	20,000
Max Message TPS	10
Max Messages	10,000
Created	[redacted]

Access Details

Select the corresponding address to complete the access. For details, see Quick Application Access andQuick Device Access

For security purposes, CoAP/CoAPS access addresses cannot be pinged.

Access ...	Access Protocol (Port)	Access Address	Custom Domain Name	Access Control
Applicati...	HTTPS (443)	[redacted]		
	MQTTS (8883)	[redacted]		
	AMQPS (5671)	[redacted]		Preset Access Credential
Device a...	CoAP (5683) | CoAPS (5684)	[redacted]		
	MQTT (1883) | MQTTS (8883...	[redacted]	Details	
	HTTPS (443)	[redacted]		

Documents/Resources

Define Product Model

Does the Server CA Certificate Need to Be Verified During MQTT Connection?

To use TLS for MQTT access, you need to load the server CA certificate. Refer to the certificate usage in **Using MQTT Demos for Access**, or download the platform CA certificate from **Obtaining Resources**.

What If Device Authentication Fails?

1. Run the ping, telnet, and traceroute commands to check whether the network is normal. If the network is abnormal, check the local firewall policy.
2. Check whether the device certificate or information, device ID, and device secret are correct.
3. Log in to the IoTDA console, choose **Devices** > **All Devices**, click the target device to access its details page, click the **Message Trace** tab and start message tracing. You can view related logs on this page after enabling the function.
4. Alternatively, choose **O&M** > **Run Logs** in the navigation pane, and configure the run logs by referring to **Description of Run Logs**. After the run logs are enabled, you can view the error information in the logs.

How Do I Analyze the Cause of Device Disconnection?

Use the **disconnection analysis** function.

How Do I Develop MQTT Devices on the IoT Platform?

1. Create products, develop product models (optional), register devices, and develop codecs (optional) on the platform.
2. Use SDKs or native protocols to connect devices to the platform.
3. (Optional) Perform operations on applications.

How Do I Connect MQTT Devices to the IoT Platform?

You can use the native MQTT protocol or an SDK. If you choose to use an SDK, select a proper SDK based on the programming language, device operating system, and resources. For details, see **Development on the Device Side**.

7 Products Models

This section describes the problems that may occur when you use product models and the solutions.

How Can I Develop a Product Model?

The IoT platform supports online and offline development of product models. For details, see **Developing a Product Model Online** and **Developing a Product Model Offline**.

How Do I Select a Data Type When Developing a Product Model Online?

The rules for configuring the data type are as follows:

- **int**: If the reported data is an integer or Boolean value, set the data type to **int**. The plug-in can be of the int or array type.
- **decimal**: If the reported data is a decimal, set the data type to **decimal**. The plug-in can be of the string, int, or array type.
- **string**: If the reported data is a string, enumerated value, or Boolean value, set the data type to **string**. If enumerated or Boolean values are reported, use commas (,) to separate the values. The plug-in can be of the string or array type.
- **dateTime**: If the reported data is a date, set the data type to **dateTime**. The plug-in can be of the string or array type.
- **jsonObject**: If the reported data is in JSON structure, set the data type to **jsonObject**. The plug-in can be of the string or array type.

Should the Values of serviceId and serviceType in a Product Model Be the Same?

Not necessary. For offline codec development, the values of **serviceId** and **serviceType** can be different. One **serviceType** can correspond to multiple **serviceId** values. For online codec development, to simplify the development process and make it easy for users to understand, **serviceType** is set to the same value as **serviceId**. The value should be consistent with the service type on the platform.

Why Does a Child Device Added Through the Gateway Not Belong to Any Product?

When a child device is added by calling an AgentLite API, the device information carried in the request must be consistent with the product model defined on the IoT platform for the child device. Otherwise, the child device cannot match with the product. Delete the child device that does not belong to any product and add it again. Ensure that the device information is correct.

How Can I Upload a Product Model with a Codec on the IoTDA Console?

IoTDA does not support the upload of product models with codecs.

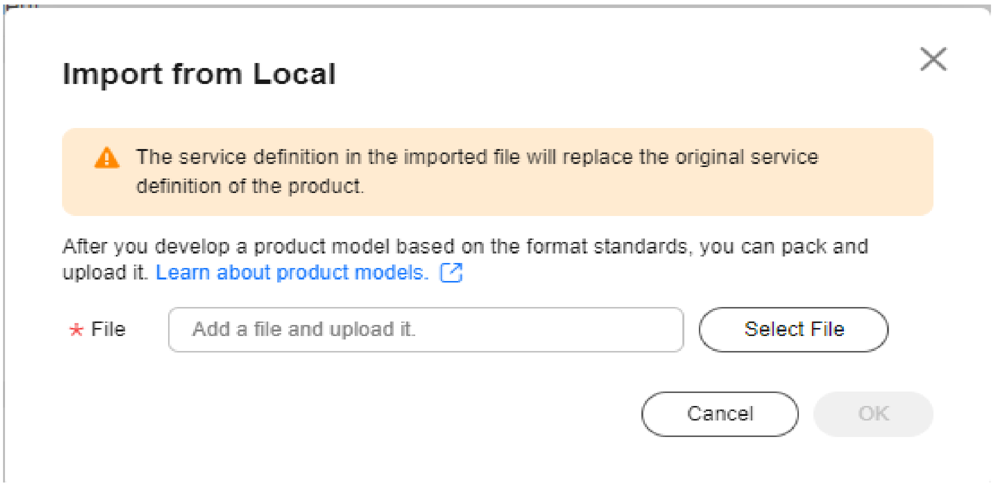
If you want to use product models with codecs, develop or import product models and codecs on the IoTDA console.

Why Is the Upload Button Unavailable When I Want to Upload a Product Model to the IoTDA Console?

Description:

On the product details page, you upload a product model file, but cannot click **OK**.

Figure 7-1 Uploading a model file



Import from Local

The service definition in the imported file will replace the original service definition of the product.

After you develop a product model based on the format standards, you can pack and upload it. Learn about product models.

* File: Add a file and upload it.

Possible Causes:

1. The file name does not meet requirements.
2. The uploaded file is not in ZIP format.

Solutions:

1. Check whether the name of the product model is in the format of **deviceType_manufacturerId_model.zip** and whether the values of **deviceType**, **manufacturerId**, and **model** are the same as those defined in the **devicetype-capability.json** file.
2. Check whether the product model is compressed to a ZIP file. If it is compressed as something else, decompress the file, compress it as a ZIP file, and upload it.

What Should I Do If the File Format Was Incorrect When I Was Uploading a Product Model to the IoTDA Console?

Description:

On the product details page, you upload a product model file, and a message is displayed, indicating that the file format is incorrect.

Possible Causes:

1. The JSON file format is incorrect.
2. The value of **commands/properties** in the **servicetype-capability.json** file is not in array format.
3. The package contains unrelated files.

Solutions:

1. Check the format of each JSON file by using a format verification tool.
2. Check whether the values of **commands** and **properties** in the **servicetype-capability.json** file are in the array format (the value should be enclosed in brackets).
3. Check whether the compressed package contains unrelated files or hidden files. If yes, delete them and upload the package again.

Why Was a Message Displayed Indicating the Manufacturer ID and Device Model Exist When I Was Uploading a Product Model to the IoTDA Console?

Description:

On the product details page, you upload a product model file, and a message is displayed, indicating that the manufacturer ID and device model exist.

Possible Causes:

The product model and plug-in of the same device model and manufacturer ID already exist on the IoT platform.

Solutions:

1. Delete the plug-in and product model of another product or account (delete the plug-in first), and then import the product model.
2. Modify the device model and manufacturer ID, and then import the product model.

8 Message Communications

What Do I Do If Data Reporting Fails?

1. If the device was registered by calling an API, check whether the device has been deleted by the IoT platform because the device did not go online within the timeout period. If the device has been deleted, re-register the device and report data again.
2. Check whether the product information specified when you called the API to register the device is consistent with that in the product model.
3. Check whether the property names in the reported message are the same as the service properties defined in the product model.
4. If the fault persists, check whether the network connection between the device and the platform is normal and whether the device is running properly.

Should I Use the Message Reporting API or Property Reporting API to Report Device Data?

It depends. If you want the data to be parsed by the IoT platform, use the property reporting API. You need to develop product models in this case. If you want the data to be transparently transmitted by the platform, use the message reporting API. Product models are not required.

How Can I Check Whether Device Message Reporting Is Successful on the IoT Platform?

Device messages are transparently transmitted, so you can check the report result by message tracing and run logs.

- Message tracing: Log in to the IoTDA console, access your instance, choose **Devices** > **All Devices** in the navigation pane, click a device to access its details page, and click **Start Trace** on the **Message Trace** tab page.
- Run logs: Configure run logs by referring to **Usage of Run Logs** and view the run logs after the function is enabled.

Why Can't I See the Data Reported by a Device on the IoTDA Console?

Possible Causes:

The reported device properties do not match the properties defined in the product model.

Solutions:

1. View the product information of the device and check whether the properties in the product model are consistent with those reported by the device.
2. Log in to the IoTDA console, choose **Devices > All Devices** in the navigation pane, select a device to access its details page, and click **Start Trace** on the **Message Trace** tab page. After the device reports the data again, check whether related logs exist. If no log exists, the data does not reach the IoT platform. In this case, check the network status on the device side. If logs exist, view the error information in the logs.
3. You can also choose **O&M** > **Run Logs** in the navigation pane, and configure the run logs by referring to **Description of Run Logs**. After the run logs are enabled, you can view the error information in the logs.

Why Was Data Displayed in the Device Shadow Inconsistent with That Reported by the Device?

When a device accesses a service, the binary data reported by the device is encoded in Base64 format, so it initially appears to be inconsistent. Once the data is decoded, it will be consistent.

Why Can't a Device Report Data Using a Custom Topic?

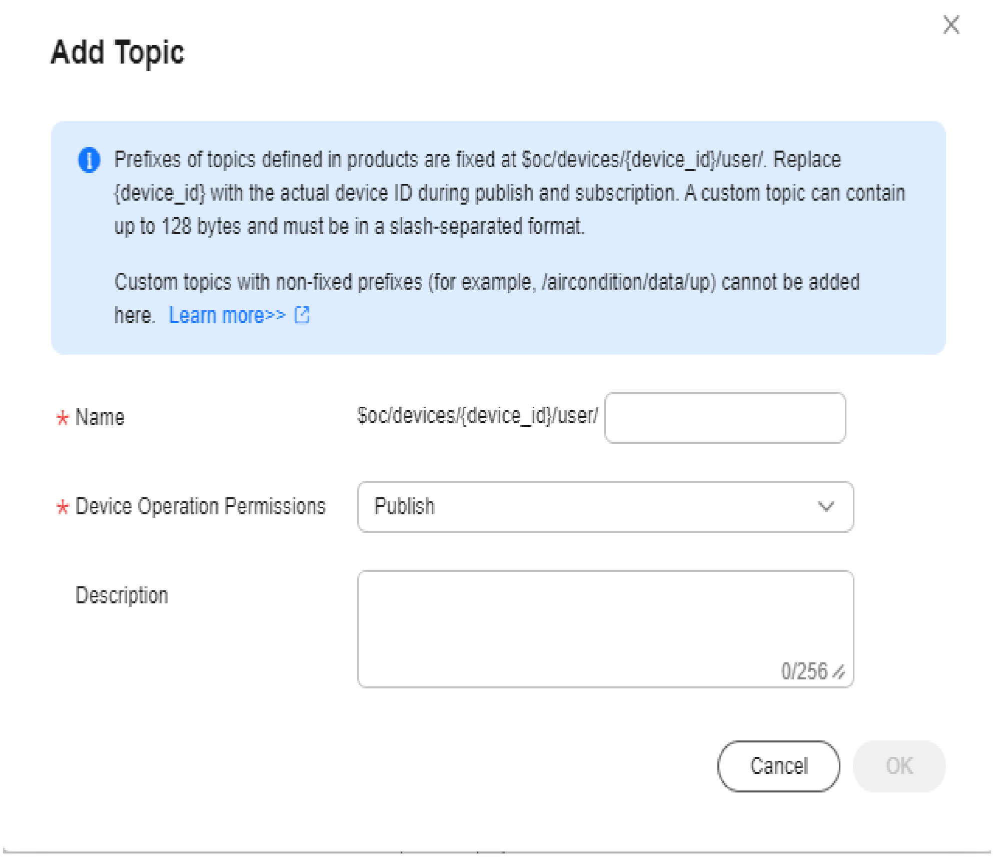
Possible Causes:

The custom topic does not match or does not have the permission to push data.

Solutions:

The prefix of the message topic has been specified as **$oc/devices/{device_id}/user/** by the platform. You can only add custom parts of non-variables after the preset prefix. Before using a custom topic to report data, ensure that the topic exists and has the release permission.

Figure 8-1 Topic management - Adding a custom topic



Add Topic

Prefixes of topics defined in products are fixed at $oc/devices/{device_id}/user/. Replace {device_id} with the actual device ID during publish and subscription. A custom topic can contain up to 128 bytes and must be in a slash-separated format.

Custom topics with non-fixed prefixes (for example, /aircondition/data/up) cannot be added here. Learn more>>

* Name: $oc/devices/{device_id}/user/

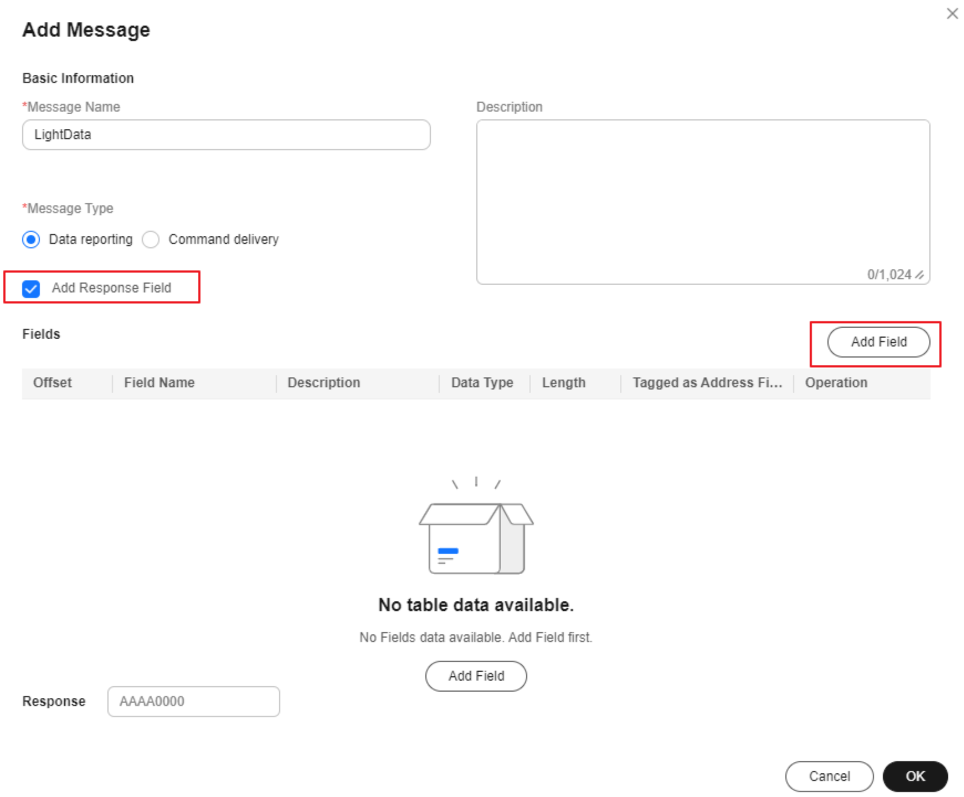
* Device Operation Permissions: Publish

Description: 0/256

Why Did Devices Fail to Receive Data Reporting Responses?

- If the codec was developed online, **Add Response Field** must be selected under **Data Reporting**.

Figure 8-2 Adding a message



Add Message

Basic Information

*Message Name

LightData

Description

0/1,024

*Message Type

● Data reporting ○ Command delivery

☑ Add Response Field

Fields

Offset	Field Name	Description	Data Type	Length	Tagged as Address Fi...	Operation

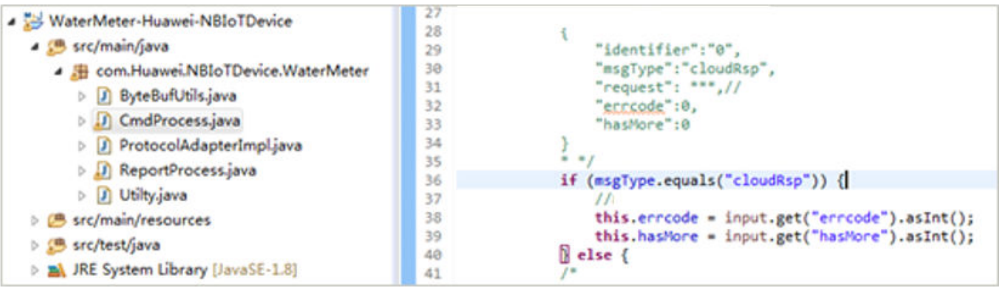
No table data available.

No Fields data available. Add Field first.

Response AAAA0000

- If the codec is developed offline, the cloudRsp logic must be defined in the codec code.

Figure 8-3 Codec code example



```
WaterMeter-Huawei-NBIoTDevice
  src/main/java
    com.Huawei.NBIoTDevice.WaterMeter
      ByteBufUtils.java
      CmdProcess.java
      ProtocolAdapterImpl.java
      ReportProcess.java
      Utility.java
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.8]
```

```

        {
            "identifier":"0",
            "msgType":"cloudRsp",
            "request": ***,//
            "errcode":0,
            "hasMore":0
        }
        * */
        if (msgType.equals("cloudRsp")) {
            //
            this.errcode = input.get("errcode").asInt();
            this.hasMore = input.get("hasMore").asInt();
        } else {
        /*
```

Why Is Data Reporting Successful at One Place But Fails Elsewhere?

Contact the NB-IoT network carrier to check whether the NB-IoT card has geographical restrictions and the local NB-IoT network status.

How Do I Convert a String Reported by an NB-IoT Device into Binary Code?

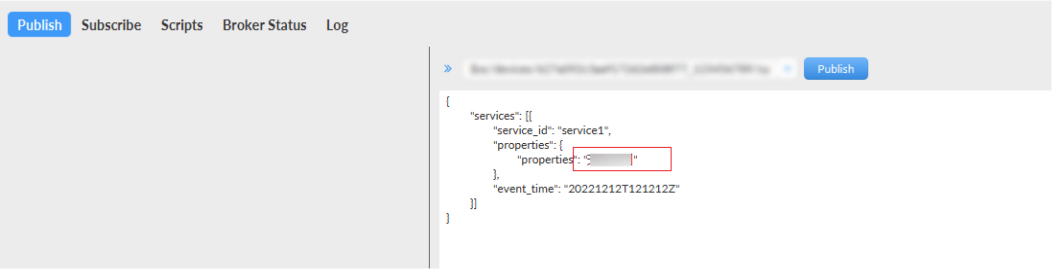
Convert the string to ASCII format first, and then to the binary code. For example, convert **abc** to ASCII (**979899**) and then to binary (**011000010110001001100011**).

Why Are Garbled Characters Displayed on the Platform When Chinese Data Is Reported?

Description:

When the MQTT.fx device simulator is used to report data, the JSON character string contains Chinese characters, as shown in the following figure.

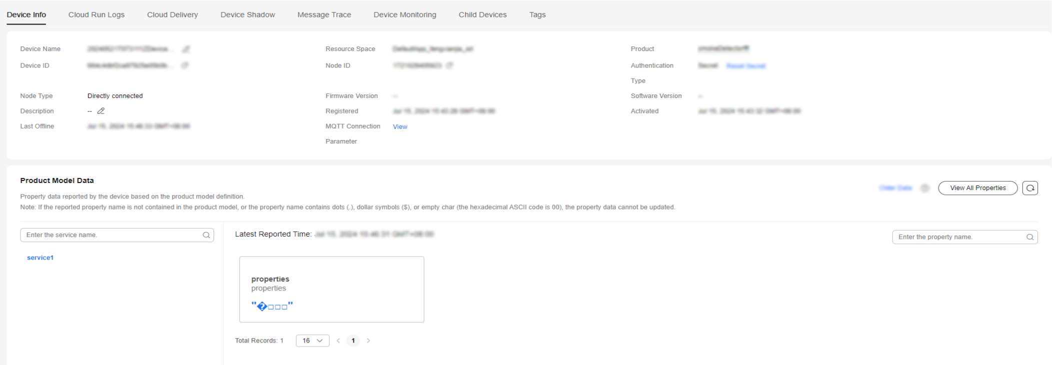
Figure 8-4 MQTT.fx data reporting



```
{
    "services": [{
        "service_id": "service1",
        "properties": {
            "properties": "[redacted]"
        },
        "event_time": "20221212T121212Z"
    }]
}
```

Garbled characters are displayed after alarms are reported to the IoTDA platform, as shown in the following figure.

Figure 8-5 Device - Example of garbled characters in device properties



Device Name [redacted]
Device ID [redacted]
Node Type Directly connected
Description --
Last Offline [redacted]
Resource Space [redacted]
Node ID [redacted]
Firmware Version --
Registered [redacted]
MQTT Connection Parameter View
Product [redacted]
Authentication Type [redacted]
Software Version --
Activated [redacted]

Product Model Data

Property data reported by the device based on the product model definition.

Note: If the reported property name is not contained in the product model, or the property name contains dots (.), dollar symbols ($), or empty char (the hexadecimal ASCII code is 00), the property data cannot be updated.

Latest Reported Time: [redacted]

properties
properties
"�□□□"

Total Records: 1 16 < 1 >

Possible Causes:

The MQTT.fx device simulator does not support Chinese characters.

Solution

1. Do not use Chinese characters when interacting with the platform.
2. Perform Unicode encoding on Chinese characters in the reported data.
3. Use a different third-party device simulator. The **device simulator** developed by IoTDA is recommended.

How Do I Deliver a Command to a Device Using LwM2M over CoAP?

Choose **Devices** > **All Devices** in the navigation pane, click a device to access its details page, click the **Cloud Delivery** tab, click the **Command Delivery** tab, and click **Deliver Command**.

For details, see **LwM2M/CoAP Device Command Delivery**.

How Do I Deliver Commands to an MQTT Device?

Choose **Devices** > **All Devices** in the navigation pane, click a device to access its details page, click the **Cloud Delivery** tab, click the **Command Delivery** tab, and click **Deliver Command**.

For details, see **MQTT Device Command Delivery**.

What Should I Do If a Command Fails to Be Delivered?

Description:

An error occurs when you call the API for delivering a command, or the API call successes but the device does not receive the command.

Possible Causes:

1. The API does not support the device protocol.
2. The downstream topic subscribed by the device is incorrect, or the upstream topic and message body of the device are incorrect.

Solutions:

1. Check whether the API supports the device protocol. Synchronous commands can be delivered only to MQTT devices.
2. For synchronous command delivery:
   - Check whether the device has subscribed to the downstream topic. If not, the device cannot receive commands from the platform. For details, see **Platform Delivering a Command**.
   - If the device has received the command, ensure that the device responds to the platform within 20 seconds after receiving the command (however, there is no time limit on a device's response to a message delivered by the platform), and the upstream topic and message body of the response are correct. Otherwise, an error will be reported by the API. **request_id** is carried in the downstream message.
3. For asynchronous command delivery:

   Check the value of input parameter **send_strategy** to see if the command is delivered immediately or cached before delivery.

   - Immediate delivery: The device receives the command immediately if the device is online.
   - Delayed delivery: The device receives the command only after the device reports data.

Can a Command Be Successfully Delivered When the Device Is Abnormal or Offline?

If the command is a synchronous one, the delivery will fail.

If the command is an asynchronous one and the input parameter **send_strategy** is set to **delay**, the command will be cached and will not be delivered until the device reports data or goes online.

Why Does a Command or Properties Delivery Always Time Out?

When you call an API to deliver commands or properties or to query properties, the intended device needs to respond in a timely manner. Otherwise, the delivery or query times out.

Does the IoT Platform Support Asynchronous Command Resending?

Yes, the platform supports asynchronous command resending. If the platform does not receive an ACK message after sending an asynchronous command to a device, the platform resends the command after 10s to 15s. You can see when the command was sent on the IoTDA console (to be specific, choose **Devices** > **All Devices** in the navigation pane, click **View** in the row of a device, and choose the **Commands** tab page). If the platform still does not receive an ACK message from the device, the platform resends the command after 20s to 30s, and tries again another 40s to 60s. 80 to 180 seconds after the preceding attempts, if the platform still does not received an ACK message from the device, the command status changes to **Timed out**.

What Are the Command and Message Statuses on the IoT Platform?

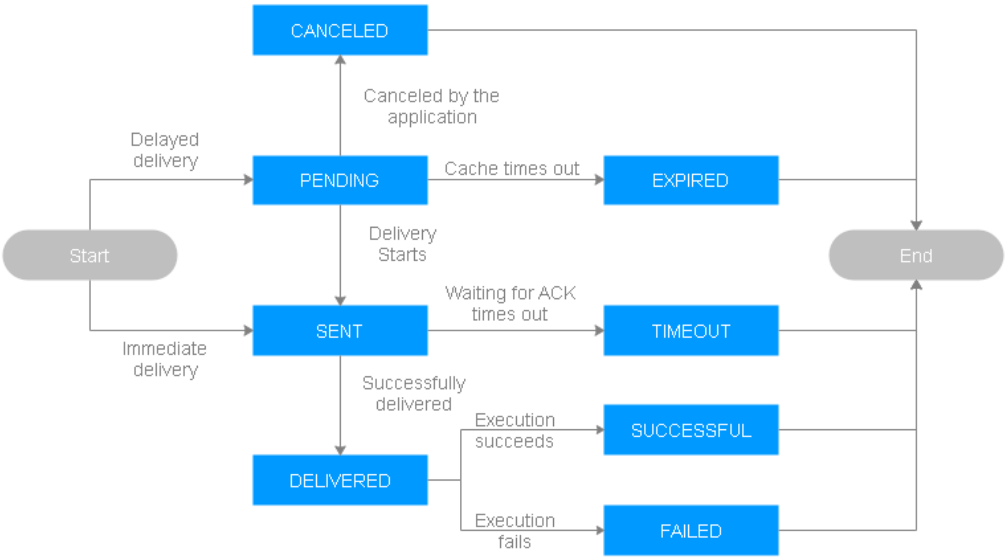
Command Status for Devices That Use LwM2M over CoAP

Commands sent by the platform can be in one of the following statuses:

- **EXPIRED**: The command cache duration has expired on the platform and is not delivered to the device.
- **SUCCESSFUL**: The platform has delivered the command to the device and received an execution result from the device.
- **FAILED**: No result is displayed after the command is parsed by the codec, or the execution result contains **ERROR CODE**.
- **TIMEOUT**: The platform fails to receive an ACK message from the device within a specified period.
- **CANCELED**: The command has been canceled on the application side.
- **PENDING**: The platform has cached the command and has not delivered it to the device.
- **SENT**: The platform has delivered the command to the device.
- **DELIVERED**: The platform has delivered the command to the device and received an ACK message from the device.

The figure below shows the conversion between command statuses.

Figure 8-6 Command status conversion

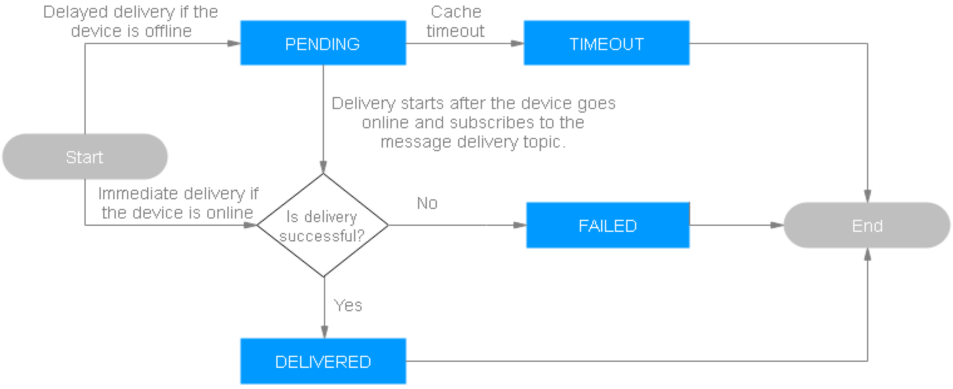


Message Status for Devices That Uses MQTT

- **PENDING**: The platform has cached the message because the device is offline.
- **TIMEOUT**: The **PENDING** message has been cached on the platform for more than 1 day and has not been delivered.
- **DELIVERED**: The message has been delivered to the device.
- **FAILED**: The message fails to be delivered to the device.

The figure below shows the conversion between message statuses.

Figure 8-7 Message status transition


Delayed delivery if the device is offline
PENDING
Cache timeout
TIMEOUT
Delivery starts after the device goes online and subscribes to the message delivery topic.
Start
Immediate delivery if the device is online
Is delivery successful?
No
FAILED
End
Yes
DELIVERED

Can the IoT Platform Deliver Commands in Batches?

Yes. An application can deliver commands in batches by calling the API for creating a batch task.

NOTE

Commands delivered in batches are pending commands.

What Do I Do If Calling Synchronous APIs Failed or Timed Out?

Generally, synchronization command delivery times out because a device does not respond within 20 seconds. Use a physical device or simulator to simulate response reporting (**Response to Synchronous Command Delivery**, **Response to Property Modification Request**, or **Response to Property Query Request**). Note that the value of **request_id** in the response reported by the device must be the same as that delivered by the platform.

Why Only One Command is Successful and the Statuses of Other Commands Are Sent When I Deliver Multiple Commands to an LwM2M Device?

Commands are asynchronously delivered to LwM2M devices one by one. After the response of the previous command is received, the next command can be delivered.

Why Did an NB-IoT Device Keep Receiving a Same Control Command from the IoT Platform?

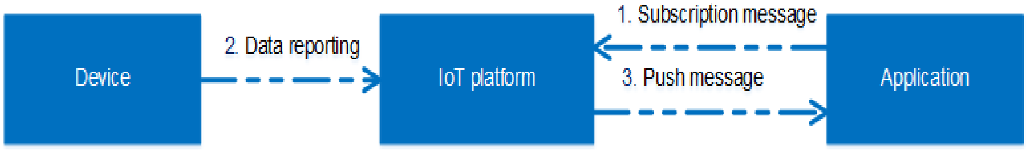
After receiving a command delivered by the platform, the device must return an ACK response to the platform. Otherwise, the platform considers that the command has failed to be delivered, and delivers the command repeatedly until it receives the ACK response or the command expires.

9 Subscription and Push

How Do I Obtain the Callback URL When Calling the Subscription API?

The following assumes that an application tries to subscribe to device data changes. The process of subscription and push is as follows:

Figure 9-1 Subscription and push process



1. The application calls an IoT platform API to subscribe to device data changes (the request carries the callback URL and notification type). The platform stores the callback URL and the notification type in the subscription list.
2. The device reports data to the platform.
3. The platform automatically pushes the device data to the application based on the callback URL stored during the subscription.

What Is a Callback URL?

The callback URL is the RESTful API address defined by the application for external access. When the platform pushes messages to the application, it calls the RESTful API of the application to send data to the application.

NOTE

The request method of the callback URL must be POST.

How Do I Obtain a Callback URL?

The callback URL consists of the connection protocol, access address of the application, and URL of the RESTful API. An example is **https://server:port/URL**.

For security reasons, HTTPS is recommended for communications between the application and platform. When HTTPS is used, the platform needs to load a certificate. For details, see **Creating a Commissioning Certificate**.

The access address of the application depends on the network where the application is located.

- If the application is deployed on the public network, the access address of the application is *Public IP address of the application*:*Port* (or *Domain name*:*Port*).
- If the application is deployed on a LAN, you must configure network address translation (NAT) for the application to generate a public network access address for the application. For details, obtain the configuration procedure of the NAT tool from the Internet.

NOTE

The callback URL can be the same or different for subscription of different notification types. It can be defined based on service requirements.

How Is an Application Notified of Changes in Command Status?

Configure the **callbackUrl** parameter when an application calls the API for creating a command. (The IP address and port number of this parameter value must be the same as those of the subscription callback URL.) When the command status changes, the IoT platform will push a message to the URL.

Can a Domain Name Be Used in a Callback URL?

Yes. You can use either an IP address or a domain name. If a domain name is bound to multiple IP addresses, perform DNS resolution to confirm that the IP address is reachable. In this case, you are advised to use the IP address in the callback URL.

Can an Application Subscribe to Platform Data When the Application Only Has an Internal IP Address?

If you are an enterprise edition user, ensure you choose the VPC of your purchased enterprise instance for your application, so the application can use the internal IP address for subscription.

Otherwise, your application cannot use an internal IP address to subscribe to the platform. Network address translation (NAT) is required when the intranet of a company or campus is used. You can use NAT to obtain a public IP address.

Why Was the Callback URL Invalid During the Subscription API Call?

A callback URL must contain a public IP address (or domain name), port, and file path.

- Correct format: **http://***[IP address]*:*Port*/*[File path]* or **http://***[IP address]*:*Port*/
- Incorrect format: **http://***[IP address]*:*Port* or **http://***[IP address]*/*[File path]*

Must the Callback URL Be an HTTPS URL?

No. The IoT platform supports pushes to HTTP and HTTPS URLs. However, you are advised to use HTTPS for encrypted transmission.

Can I Change the Callback URL?

Yes. A callback URL can be changed. To change the IP address and port number of the callback URL, call the API for deleting subscriptions in batches to delete the previous callback URL, upload a new CA certificate, and make subscriptions again.

Must the Certificate of the Callback URL Be Issued by an Authoritative Organization?

No. Custom certificates are also supported. However, it is recommended that you use a certificate issued by an authoritative organization.

What Should I Do If the Push Certificate Is Invalid?

If the certificate was provided by yourself, create a new one and upload it to the IoT platform.

If the certificate was provided by the manufacturer, contact the manufacturer for a new one.

How Can I Obtain the subscriptionId Needed in Calling the API for Deleting a Subscription?

The IoT platform returns the **subscriptionId** when the subscription API is called. To query the value of **subscriptionId**, call the API for querying subscriptions in batches.

How Can I Obtain the Address Used by the IoT Platform to Push Messages?

When configuring firewall policies on an application, you can obtain the address from the platform support personnel.

Can the IoT Platform Push Data Reported by Different Devices Under the Same Application to Two Servers?

Yes. The two servers can receive data pushed by the platform if their callback URLs are different.

Why Did an Application Not Receive Push Messages After an NB-IoT Device Subscribed to Message Confirmation and Command Response Notifications?

The message confirmation and command response notifications are not used on NB-IoT networks (CoAP and LwM2M). On NB-IoT networks, to enable an application to receive command response notifications, configure the **callbackUrl** parameter when calling the API for creating a command.

How Do I Obtain a Device Property Unit in the Device Property Reporting Notification?

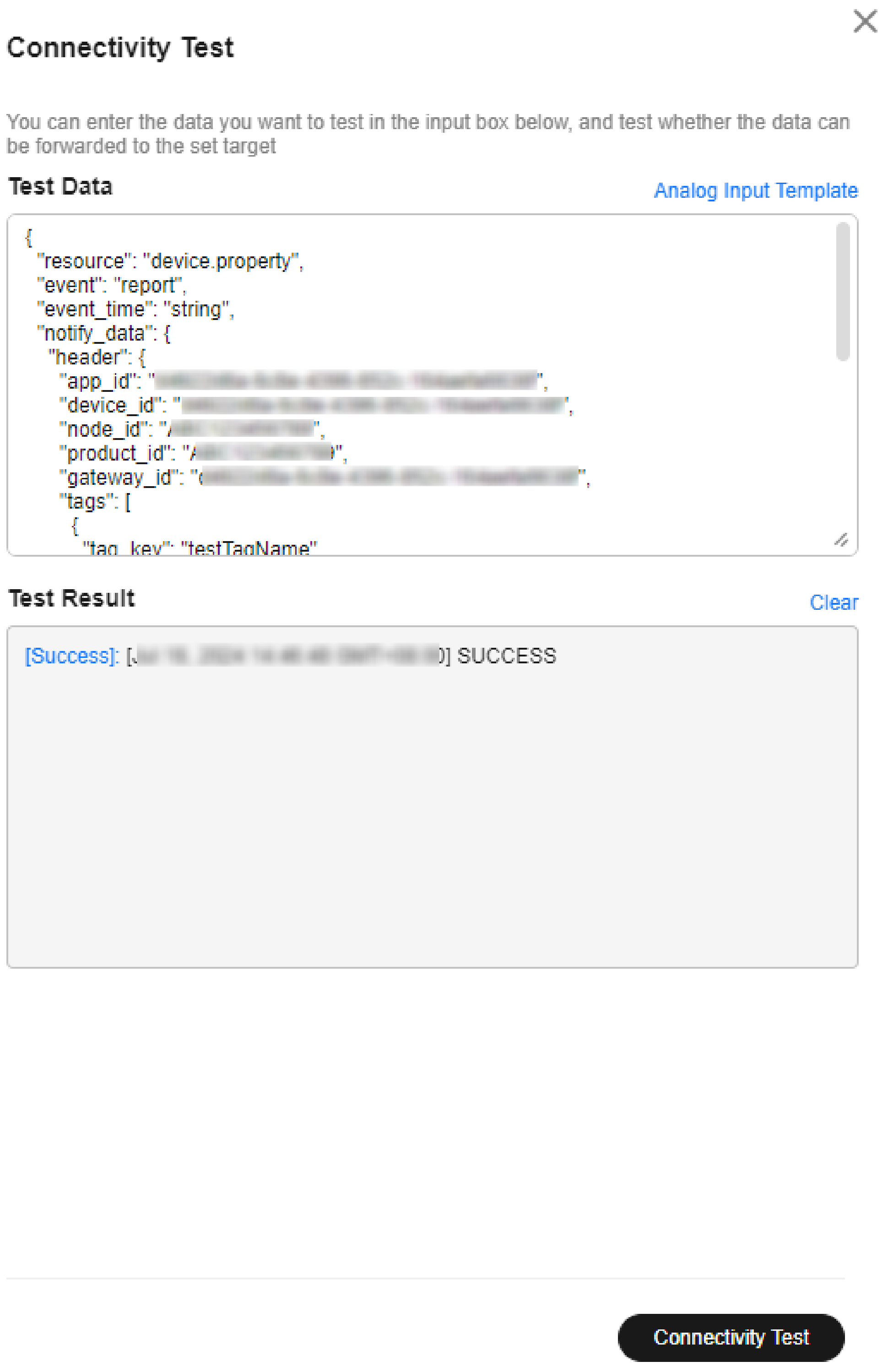
You can obtain the device property unit using the API for querying a product based on the product model of the device. For details, see **Query a Product**.

What Do I Do If Message Push Fails After Subscription?

Description:

An application does not receive pushed messages after subscription, or the result of the **connectivity test** on **Third-party application (HTTP push)** is displayed as failed on the IoTDA console.

Figure 9-2 Forwarding target - Connectivity test results



Connectivity Test

You can enter the data you want to test in the input box below, and test whether the data can be forwarded to the set target

Test Data

```
{
  "resource": "device.property",
  "event": "report",
  "event_time": "string",
  "notify_data": {
    "header": {
      "app_id": "[redacted]",
      "device_id": "[redacted]",
      "node_id": "[redacted]",
      "product_id": "[redacted]",
      "gateway_id": "[redacted]",
      "tags": [
        {
          "tag_key": "testTagName"
```

Test Result

```
[Success]: [[redacted]] SUCCESS
```

Possible Causes:

1. If the subscription callback URL is an HTTPS address, the server certificate may be faulty or the CA certificate corresponding to the server certificate may have not been uploaded to the platform.
2. The service port corresponding to the subscription callback URL is not enabled.
3. If the subscription address is a domain name and the domain name is bound to multiple IP addresses, the DNS resolution may be faulty.

Solutions:

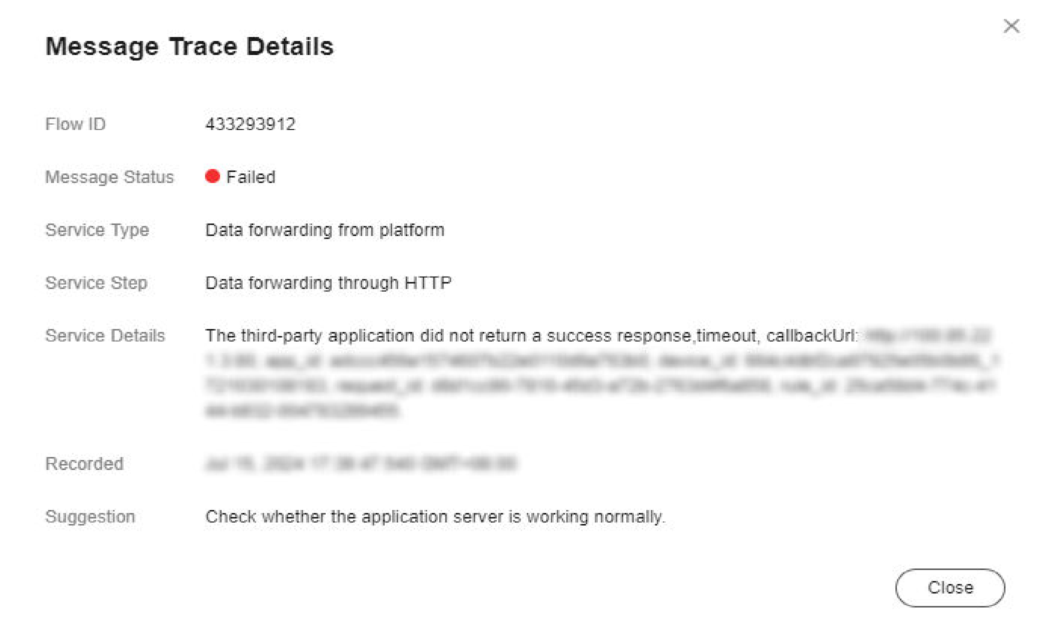
1. If the subscription callback URL is an HTTPS address, apply for a certificate from a CA during commercial use and upload the certificate to the platform by referring to **Uploading a Certificate for Device Access to the IoT Platform**. Before commissioning, ensure that the **certificate creation** process is correct. If **Check Common Name** is selected during certificate upload, ensure that the domain name of the application is the same as the common name of the certificate. You can use the **certificate verification tool** to view the common name of the certificate.
2. Check whether the server port corresponding to the subscription callback URL is enabled. Run the **telnet** command on another computer on the external network to check whether the IP address and port are enabled. If they are not enabled, enable them or change the IP address and port of the subscription callback URL.
3. If the subscription callback URL is a domain name and the domain name is bound to multiple IP addresses, check whether the IP addresses after DNS resolution are correct and reachable.
4. Use the **message tracing** function on the IoTDA console to trigger the push. Then, view the message tracing data, check whether the platform triggers subscription and push and whether the push is successful.
5. Capture packets on the application and check whether the message pushed by IoTDA is received.

What Happens If an Application Receives Data but an Error Is Displayed on the Platform Indicating that the Push Failed?

Description:

The application receives the data, but the message tracing details show that the push fails.

Figure 9-3 Message tracing -HTTP push failure details



Message Trace Details

Flow ID	433293912
Message Status	● Failed
Service Type	Data forwarding from platform
Service Step	Data forwarding through HTTP
Service Details	The third-party application did not return a success response,timeout, callbackUrl: [redacted]
Recorded	[redacted]
Suggestion	Check whether the application server is working normally.

Possible Causes:

If an application does not return the 200 OK status code to IoTDA within 15 seconds after receiving the data, and the platform considers that the push fails.

Solutions:

After receiving the data, the application needs to return the status code 200 OK.

What Is the Difference Between deviceDataChanged and deviceDatasChanged?

If an application subscribes to device data change notifications (**deviceDataChanged**) or batch device data change notifications (**deviceDatasChanged**), the IoT platform pushes notifications of device data changes to the application either way, but the format for encapsulating data is different.

For example: A water meter can have two service types: Battery and Connectivity. The data of the two services is reported each time.

- If **deviceDataChanged** is subscribed to, the platform pushes the data to the application twice, Battery service data for the first time, and Connectivity service data for the second time. For example:

```
{
    "notifyType":"deviceDataChanged",
    "deviceId":"********",
    "gatewayId":"********",
    "requestId":null,
    "service":{
        "serviceId":"battery",
        "serviceType":"battery",
        "data":{
            "batteryLevel":66
        },
        "eventTime":"20170211T034003Z"
    }
}
{
    "notifyType":"deviceDataChanged",
    "deviceId":"********",
    "gatewayId":"********",
    "requestId":null,
    "service":{
        "serviceId":"Connectivity",
        "serviceType":"Connectivity",
        "data":{
            "signalStrength":72,
            "cellId":4022250974,
            "tac":61374,
            "mnc":91,
            "mcc":235
        },
        "eventTime":"20170211T092317Z"
    }
}
```

- If **deviceDatasChanged** is subscribed to, the platform includes information for both services in a single batch of data sent to the application.

```
{
    "notifyType":"deviceDatasChanged",
    "requestId":null,
    "deviceId":"********",
    "gatewayId":"********",
    "services":[
        {
            "serviceId":"battery",
            "serviceType":"battery",
            "data":{
                "batteryLevel":66
            },
            "eventTime":"20170211T034003Z"
        },
        {
            "serviceId":"Connectivity",
            "serviceType":"Connectivity",
            "data":{
                "signalStrength":72,
                "cellId":4022250974,
                "tac":61374,
                "mnc":91,
                "mcc":235
            },
            "eventTime":"20170211T034003Z"
        }
    ]
}
```

Why Does the Application Receive Multiple Push Messages After a Device Reports a Piece of Data?

The possible causes are as follows:

- If **notifyType** is **deviceDataChanged** and the contents of the multiple push messages are different, the data reported by the device may involve multiple services (the **data** parameter contains multiple JSON objects). As a result, the IoT platform splits the data into multiple push messages.
- If two messages are received, and **notifyType** of one message is **deviceDataChanged** and **notifyType** of the other is **deviceDatasChanged**, you have subscribed to both device data change notifications and batch

device data change notifications. As a result, the platform pushes two messages.

- If the same message is pushed repeatedly, the application may not return a response to the push message in a timely fashion. The platform considers the push to have failed and enables re-push.

10 Codecs

How Can I Develop Codecs?

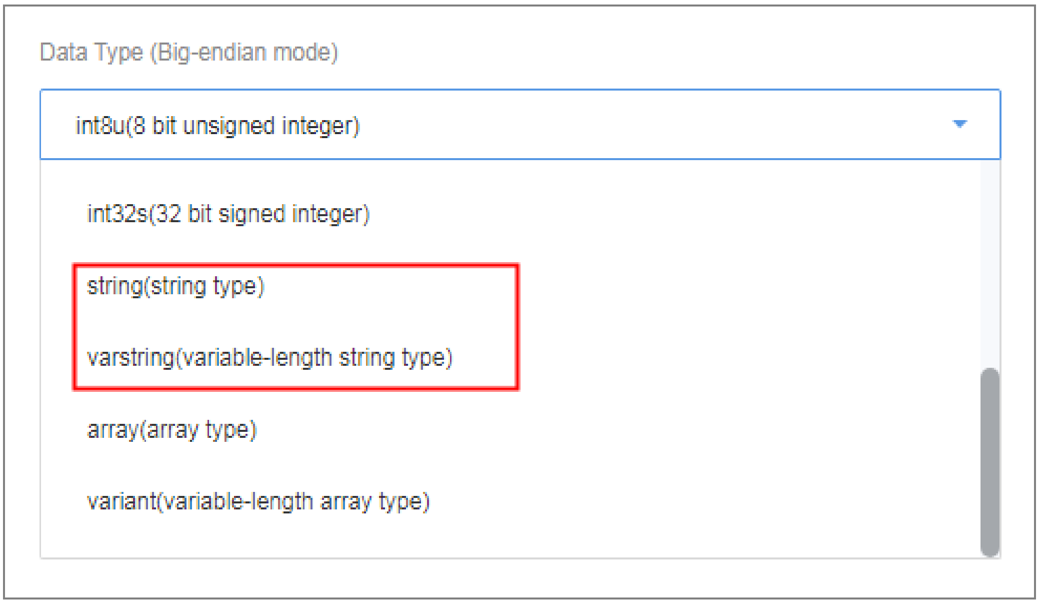
The IoT platform supports the codec online development. For details, see **Online Development**.

What Is the Code Rule for Fields of the string and varstring Types?

If **Data Type** of a field is **string(string type)** or **varstring(variable-length string)**, the codec performs encoding and decoding using ACSII.

For details on the codec online development, see **Codec for Strings and Variable-Length Strings**.

Figure 10-1 Data type example 1



Data Type (Big-endian mode)

int8u(8 bit unsigned integer)

int32s(32 bit signed integer)

string(string type)

varstring(variable-length string type)

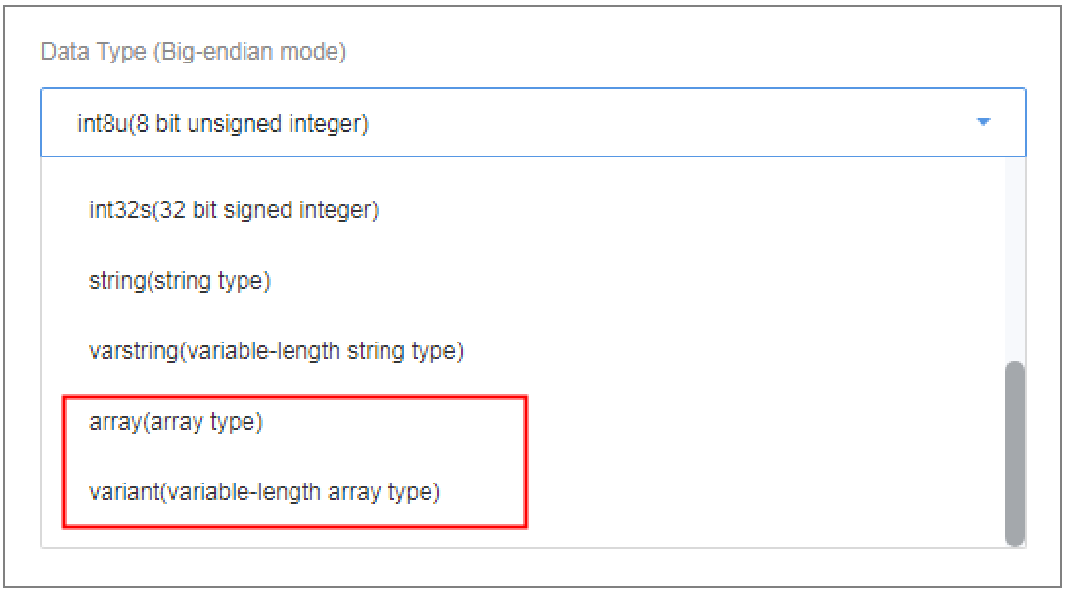
array(array type)

variant(variable-length array type)

What Is the Code Rule for Fields of the array and variant Types?

If **Data Type** is **array(array type)** or **variant(variable-length array type)**, the codec performs encoding and decoding using Base64. For details on the codec online development, see **Codec for Arrays and Variable-Length Arrays**.

Figure 10-2 Data type example 2



Data Type (Big-endian mode)

int8u(8 bit unsigned integer)

int32s(32 bit signed integer)

string(string type)

varstring(variable-length string type)

array(array type)

variant(variable-length array type)

How Do I Use messageId During Online Codec Development?

When messages of the same type are created, such as two data reporting messages, **messageId** must be configured to distinguish these messages and this parameter in each message must be in the same place on the list. The **messageId** field applies to the following scenarios:

- There are two or more data reporting messages or command delivery messages.
- A command response can be regarded as a type of data reporting message. Therefore, if a command response exists, **messageId** must be added to the data reporting message.
- A data reporting response can be regarded as a type of command delivery message. Therefore, if a data reporting response exists, **messageId** must be added to the command delivery message.

How Do I Configure a Command Delivery Response During Online Codec Development?

After receiving a command, a device sends an ACK message or a command delivery response. The ACK message indicates that the command has reached the device, and the command delivery response indicates the command execution result. If the device needs to return a command delivery response after receiving the command, the **messageId** and **mid** fields must be configured:

- **messageId** must be configured in both the data reporting message and the command response, and this field in the two messages must be in the same place on the list, so that the codec can distinguish between the two.
- **mid** must be configured in the command delivery message and the command response, and this field in the two messages must be in the same place on the list, so that the codec can associate the two.

What Do I Do If a Codec That Was Developed Online Fails to Be Deployed?

Description:

You develop a codec on the development page and click **Deploy**. The deployment fails.

Possible Causes:

1. If the codec is successfully downloaded but fails to be deployed, there may be network issues.
2. If the codec fails to be downloaded, the possible causes are as follows:
   - The value of **messageId** in the messages of the same type (for example, two data reporting messages) is duplicate or their positions in the messages are different.
   - **Data Type** is not set to **int** for **messageId**.
   - The default value of a field is not in hexadecimal format.
   - A field name contains a Java keyword, for example, **type** or **int**.

Solutions:

1. If the codec is successfully downloaded but fails to be deployed, check the network status and try again.
2. If the codec fails to be downloaded, check whether the codec definition meets the requirements.

What Do I Do If a Script-based Codec or a Codec Developed Offline Fails in Encoding or Decoding?

- For a script-based codec, choose **Products** > **Codec Development** > **Edit Script** to debug the codec and modify the codec based on the debugging result.
- For a codec developed offline, debug the codec using the **codec test tool** and modify the codec based on the test result.

What Do I Do If a Codec That Was Developed Offline Fails to Be Deployed?

1. Use the **IoT codec test tool** to check the codec.
2. Rectify the fault based on the error code returned by the tool. For details on how to handle different errors, see *User Guide for IoT Codec Test Tool* in the tool package.

What Do I Do If a Codec That Was Developed Offline Fails to Be Uploaded?

Codecs that are developed offline must be checked before being uploaded to the IoT platform. Use the **IoT codec test tool** to check the codec package and rectify the fault based on the error message.

11 OTA Upgrades

What Is a Software or Firmware Upgrade?

A software upgrade refers to the upgrade of the system software and application software of the device. A firmware upgrade refers to the upgrade of the underlying drivers of the device hardware.

You can upload a software/firmware upgrade package to the IoTDA platform or use a file associated with an object on OBS for device remote upgrades.

Can the IoT Platform Download Software or Firmware Packages from Third-party Servers?

No. Currently, you can only use software/firmware packages uploaded to the IoT platform or associated with objects on OBS. To upload a software/firmware package, choose **Devices > Software/Firmware Upgrades** in the navigation pane, and go to the **Software List** or **Firmware List** tab page.

Can the Target Version Be Earlier Than the Source Version?

Yes. Software or firmware can be rolled back from a later version to an earlier version. The IoT platform checks whether the target version is the same as the source version. If they are consistent, the message "The current version is the same as the target version" is displayed, and the platform does not initiate a rollback or upgrade. If they are inconsistent, the platform starts a rollback or upgrade.

How Do I Obtain Software or Firmware Packages and Their Version Numbers?

Obtain the software package and its version number from the device manufacturer. Obtain the firmware package and its version number from the module manufacturer.

What Are Common Software or Firmware Upgrade Errors?

If an error occurs during upgrade, you can view the error messages on the **Execution Details** tab page. (To access this page, choose **Software/Firmware**

Upgrades in the navigation pane of the IoTDA console, go to the upgrade task list, and click **View** in the row of a task.) Common error messages are as follows:

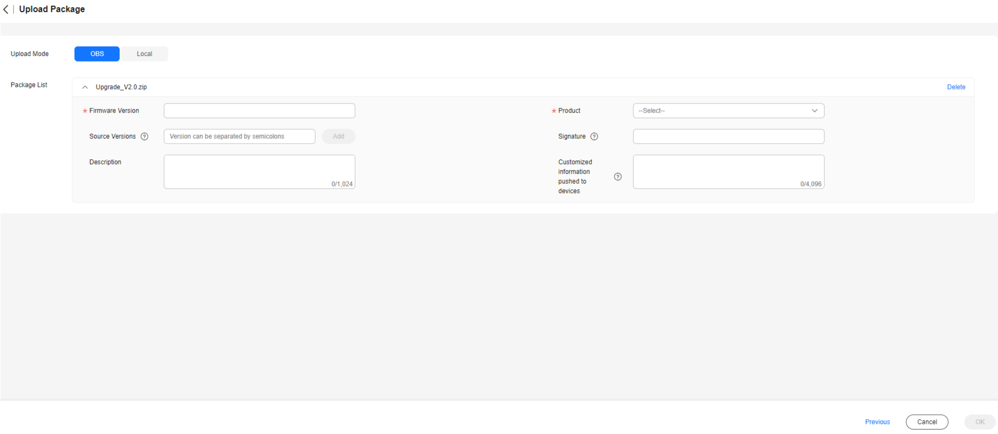
Table 11-1 Software/Firmware upgrade error description

Error Message	Description	Recommendation
Device Abnormal is not online	The device is not online.	Check whether the device is online.
Task Conflict	A task conflict has occurred.	Check whether there is an ongoing software or firmware upgrade for the current device.
Waiting for the device online timeout	The device did not go online within the specified time.	Check whether the device is connected to the network.
Waiting for report cellId timeout	The device did not report the cell ID within the specified time.	View the module logs to check whether the device has reported the cell ID.
Waiting for report device firmware version timeout	The device did not report the firmware version within the specified time.	View the module logs to check whether the device has reported the firmware version.
Wait for the device to report upgrade result timeout	The device did not report the upgrade result within the specified time.	View the module logs to check whether the device has reported the upgrade result.
Updating timeout and query device version for check timeout	The device did not report the upgrade result or device version within the specified time.	View the module logs to check whether the device has reported the upgrade result and device version.
Waiting for device downloaded package timeout	The device did not download the firmware package within the specified time.	View the module logs to check whether the device has downloaded the firmware package.
Integrity check failure for new downloaded package	The integrity check of the downloaded firmware package failed.	View the module logs to check whether the downloaded firmware package is complete.
Unsupported package type	The firmware package type is not supported.	View the module logs to check whether the device status and firmware package provided by the manufacturer are correct.

Error Message	Description	Recommendation
Not enough storage for the new firmware package	The storage space is insufficient for the firmware package.	Check the storage space of the device.
Out of memory during downloading process	The memory was insufficient during the download.	Check the device memory.
Connection lost during downloading process	The connection was interrupted during the download.	Check the device connection status.
Invalid URI	The URI is invalid.	Check whether the download address of the firmware package is correct.
Firmware update failed	The firmware failed to be upgraded.	View the module logs to check the device firmware.

What Should I Enter for the Source Versions When I Upload a Firmware Package on the IoTDA Console?

Figure 11-1 OTA upgrade - Firmware uploading example



Upload Package

Upload Mode: OBS

Package List: Upgrade_V2.0.zip

The source versions should include the current versions of devices whose firmware you want to upgrade. Otherwise, the upgrade is not allowed. To obtain the current version of a device, log in to the IoTDA console, choose **Devices** > **All Devices** in the navigation pane, and click **View** in the row of the target device.

Are Services Interrupted During a Software or Firmware Upgrade?

Yes. Services are interrupted during the upgrade. Services such as data reporting and property modification cannot be conducted during a module or chip upgrade.

How Can I Retry When Some Devices in the Group Fail to Be Upgraded?

If the entire task fails, only the failed subtasks are retried. The devices that have been successfully upgraded will not be upgraded again.

The retry interval and retry times can be configured for software upgrade, but only the retry times can be configured for firmware upgrade.

Is Resumable Transmission Supported When the IoT Platform Is Sending Software or Firmware Packages to the Devices?

Yes. Devices must record the fragment location of the software or firmware packages when the transmission is interrupted, so that package fragments that are not transmitted can be sent after the transmission is restored. If the device detects that the verification code of the upgrade package changes after the transmission is restored, the upgrade package is a new one. You need to clear the previous fragments and download the upgrade package again.

Why Does a Software/Firmware Upgrade Task Fail to Be Created When I Select Multiple Upgrade Packages?

The source versions supported by the selected upgrade package are duplicate, or the number of selected upgrade packages exceeds the upper limit. When creating an upgrade task, you can select up to 10 upgrade packages. Supported source versions of the upgrade packages must be unique. If no source version is specified for an upgrade package, the package can be used to upgrade devices of all source versions by default.

Which Devices Will Be Retried If I Click Retry All on the Task Details Page?

If you click **Retry All**, the system retries the failed, to-be-retried, or stopped status batch tasks.

Why Does a Device Fail to Download the Upgrade Package During the Upgrade?

Check whether the upgrade package link times out. For details about the upgrade link timeout duration, see section **Platform Delivering an Upgrade Event**.

Why Does an Upgrade Package Fail to Be Downloaded After the Upgrade Result Is Reported?

After the platform delivers an upgrade package, the device downloads the upgrade package and reports the upgrade result to the platform. If the upgrade result is reported, the platform considers that the upgrade is complete. Therefore, you cannot download the upgrade package again.

What Do I Do If a Device Upgrade Task Times Out?

A software/firmware upgrade task consists of multiple phases. Different phases have different timeout periods. If the device does not report a response within the timeout period of a phase, the upgrade task times out. For details about the timeout period, see **Firmware Upgrades**.

12 Application Integration

How Do an Application Call a Platform API for Authentication?

A customer application can call the IAM API to obtain the X-Auth-Token or integrate the SDK to use the AK/SK authentication. The password of a Huawei Cloud account may need to be updated periodically. It is recommended that the production system integrate the **SDKs for the Application Side** of IoTDA and use AK/SK authentication.

What Can I Do If the SDK on the Application Side Reports the Missing request header'X-Auth-Token' for method parameter of type String?

Description:

An application is integrated with the IoTDA SDK and uses the AK/SK for authentication. However, the status code 400 is returned, and the error message is **Missing request header 'X-Auth-Token' for method parameter of type String**.

Possible Causes:

The AK/SK signature algorithm does not match the signature algorithm supported by the cluster.

Solutions:

1. Use the common **AK/SK Signature Algorithm** to access the access address of ab IoTDA basic edition instance in the CN North-Beijing4 region.
2. When accessing the access address of an IoTDA standard or enterprise edition instance, you need to specify the derived AK/SK signature algorithm. For details, see **SDKs for the Application Side**.

How Do I Obtain the Access Addresses and Certificates of the Old and New Domain Names?

No CA certificate is provided for the new domain name. For details about the access addresses of the new and old domain names and the certificate for the old domain name, see **Obtaining Resources**.

What Are the Differences in Access Authentication If I Use the New or Old Domain Names?

When you use the new domain name, the IAM API for authentication is called, and the Huawei Cloud account and password need to be carried in the request. When you use the old domain name, the authentication API of the IoT platform is called, and the application ID and secret need to be carried.

How Does an Application Obtain Data Reported by Devices to the IoT Platform?

Data forwarding rules are configured based on the rule engine.

1. Data forwarding rules are configured based on the rule engine. The IoT platform calls the configured rules to forward device data to other Huawei Cloud services (such as Kafka, DIS, and OBS), third-party applications
2. The IoT platform calls the configured rules to forward device data to third-party cloud services (such as Kafka, DIS, and OBS), from which you can obtain the data to implement your service.

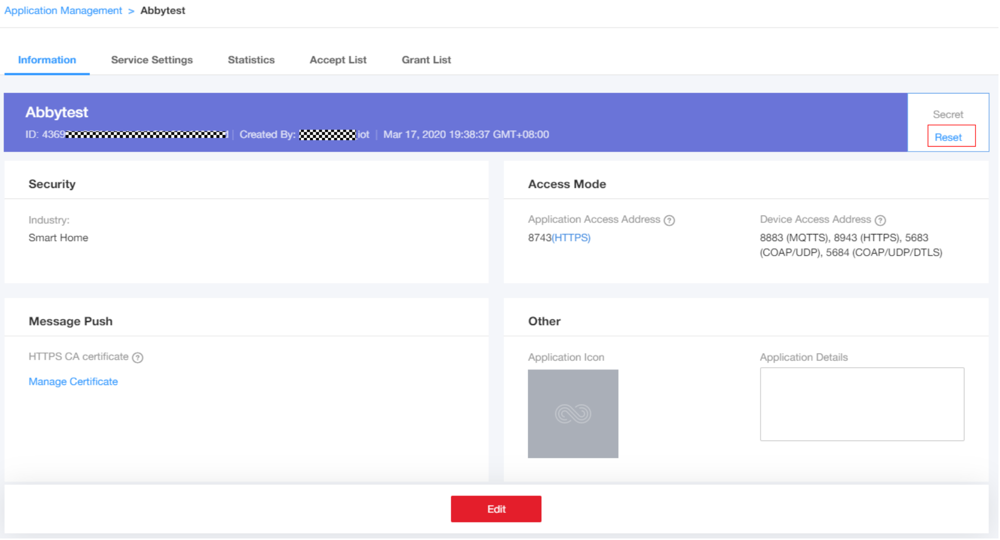
How Do I Obtain the ID and Secret of an Application?

NOTE

If you have enabled IoTDA before 00:00 on March 20, 2020 (GMT+08:00), you can perform the preceding operations to obtain the ID and secret. Otherwise, **submit a service ticket** to contact technical support.

1. Check the access address and certificate of the API of the earlier version and click the button to access IoTDA.
2. Select an application in the application list. The application ID is shown on the displayed page. Click **Reset** to obtain the secret.

Figure 12-1 Application details page



Abbytest
ID: 4369[redacted] | Created By: [redacted]iot | Mar 17, 2020 19:38:37 GMT+08:00

Security
Industry:
Smart Home

Access Mode
Application Access Address ?
8743(HTTPS)
Device Access Address ?
8883 (MQTTS), 8943 (HTTPS), 5683 (COAP/UDP), 5684 (COAP/UDP/DTLS)

Message Push
HTTPS CA certificate ?

Other
Application Icon
Application Details

Why Did an Application Fail to Call an API?

1. Check whether the network connection of the application is normal.
2. Check whether the **application access address and port number** of the IoT platform in the request are correct.
3. Run **ping {***application access address of the platform***}** to check whether the application can access the application access address of the platform.
4. Run **telnet {***application access address of the platform* **} {***port number***}** to check whether the application access port of the platform can be enabled.
5. Check whether the application is integrated with the platform certificate. The certificate can be downloaded from **Obtaining Resources**.
6. Check whether the parameters are correctly set based on the **API Reference**.
7. When V3 APIs are used, the validity period of **accessToken** returned by calling the authentication API is one hour. After the validity period expires, other APIs cannot be called. In this case, call the authentication API to obtain the new **accessToken** and then call other APIs again.
8. When V5 APIs are used, the validity period of **X-Auth-Token** returned by calling the **authentication** API is 24 hours. After the validity period expires, other APIs cannot be called. In this case, call the authentication API to obtain the new **X-Auth-Token** and then call other APIs again.
9. If the previous steps do not resolve the issue, troubleshoot based on the error code returned when the API was called. For details about the error codes, see **Error Codes**. If an application has encapsulated error codes, you can use Postman to call the same API to obtain the original error code and description returned by IoTDA, and then rectify the fault based on the recommendations provided.

   For example, if a message is displayed indicating that the node ID already exists when the API **Creating a Device** is called, check whether a device with the same node ID exists under your account. If no device with the same node ID is found under your account, **submit a service ticket** and contact technical support to check whether the node ID is used for device access.

How Does an Application Obtain Device IMEIs?

After the **bindDevice** notification is subscribed to, the IoT platform pushes the IMEI and **deviceId** to the application when a device is connected to the platform. For details, see the description of the API for subscribing to service data.

How Does an Application Deliver a Cached Command?

Set **expireTime** to a value greater than **0** when calling the API for creating a command.

expireTime indicates the validity period of a command, which is measured in seconds. The command will not be delivered after the specified time elapses. If this parameter is not specified, the default validity period is 48 hours (172,800 seconds).

Do I Have to Use Java to Call the APIs of the IoT Platform?

No. The APIs of the platform are standard RESTful APIs that support multiple languages such as Java, Python, and Go. For details, see **SDKs for the Application Side**.

Why Were Devices Registered on the IoT Platform Deleted After a Period of Time?

When a device is registered by calling an API, the **timeout** parameter must be configured. After the registration is complete, if the device is not bound to the platform within the time specified by **timeout**, the platform will delete the device.

The value range of **timeout** is 0 to 2147483647 (68 years), which is measured in units of seconds. When this parameter is set to **0**, the timer never expires and the device is never deleted as a result of a timeout.

Can I Create the Same Product in Different Instances? Can the Product Name and Product ID Be the Same in Different Resource Spaces or the Same Resource Space?

The same product can be created in different instances. The same product can be created in different resource spaces of the same instance.

Can I Create the Same Device in Different Instances?

The same device can be created in different instances. The same device cannot be created under the same instance.