Alibaba Cloud IoT Platform

Quick Start

Issue: 20181217

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade

secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *{stand | slave}*

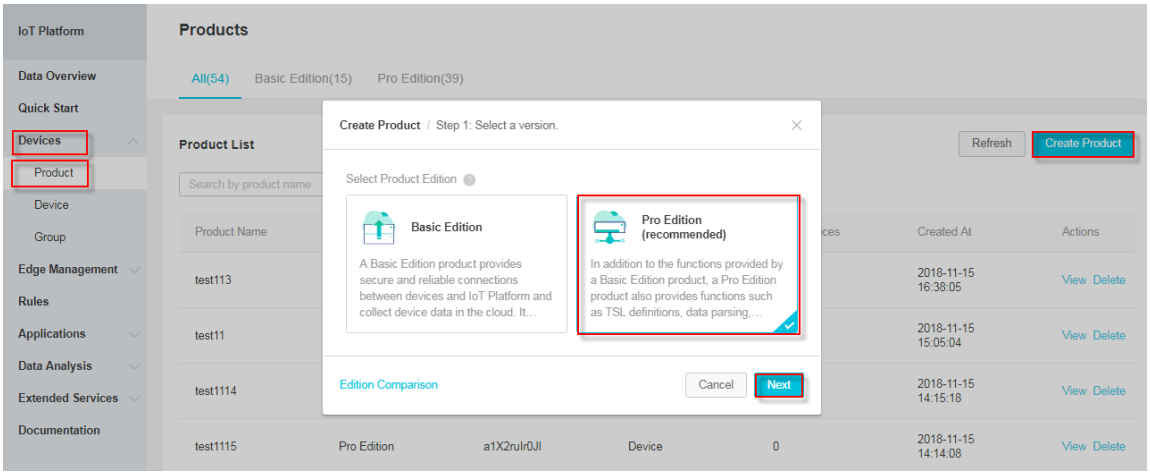
Contents

1 Create products and devices

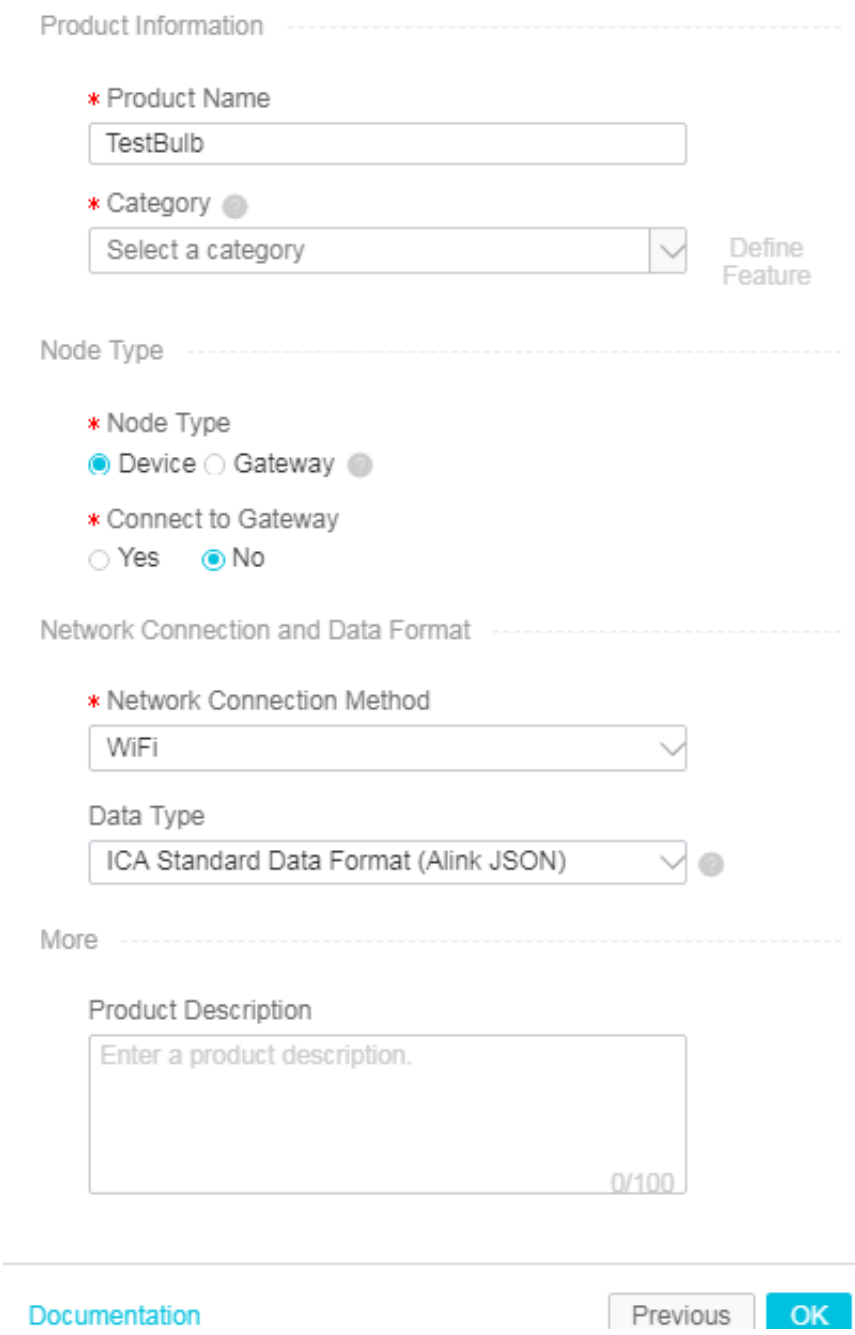
The first step in using IoT Platform is to create products and devices. A product is a collection of devices that typically have the same features. You can manage devices in batch by managing the corresponding product.

Procedure

1. Log on to the *IoT Platform console*.
2. Create a product.
   a) In the left-side navigation pane, click **Devices** > **Product**. On the **Products** page, click **Create Product**.
   b) Select **Pro Edition**, and then click **Next**.



   c) Enter all the required information and then click **OK**.



The parameters are described as follows:

Parameter	Description
Product Name	In this example, the product is named as **TestBulb**. The product name must be unique within the account. A Product name is 4 to 30 characters in length, and can contain Chinese characters, English letters, digits and underscores. A Chinese character counts as two characters.
Category	In this example, the product category is **Custom category** indicating that features of the product is self-defined.

Parameter	Description
Node Type	In this example, the node type is **Device**. • **Device**: Indicates that devices of this product cannot be mounted with sub-devices. This kind of devices can connect to IoT Platform directly or as sub-devices of gateway devices. • **Gateway**: Indicates that devices of this product connect to IoT Platform directly and can be mounted with sub-devices. A gateway can manage sub-devices, maintain topological relationships with sub-devices, and synchronize topological relationships to IoT Platform.
Connect to Gateway **Note:** This parameter appears if the node type is **Device**.	Indicates whether or not devices of this product can be connected to gateways as sub-devices. • **Yes**: Devices of this product can be connected to a gateway. • **No**: Devices of this product cannot be connected to a gateway.
Network Connection Method **Note:** This parameter appears if you select **No** for **Connect to Gateway**.	Select a network connection method for the devices. In this example, **WiFi** is selected.
Data Type	Select a format in which devices exchange data with IoT Platform. In this example, **ICA Standard Data Format (Alink JSON)** is selected. ICA Standard Data Format (Alink JSON): The standard data format defined by IoT Platform for device and IoT Platform communication.
Product Description	Describe the product information. You can enter up to 100 characters.

Once the product is created successfully, it appears in the product list.

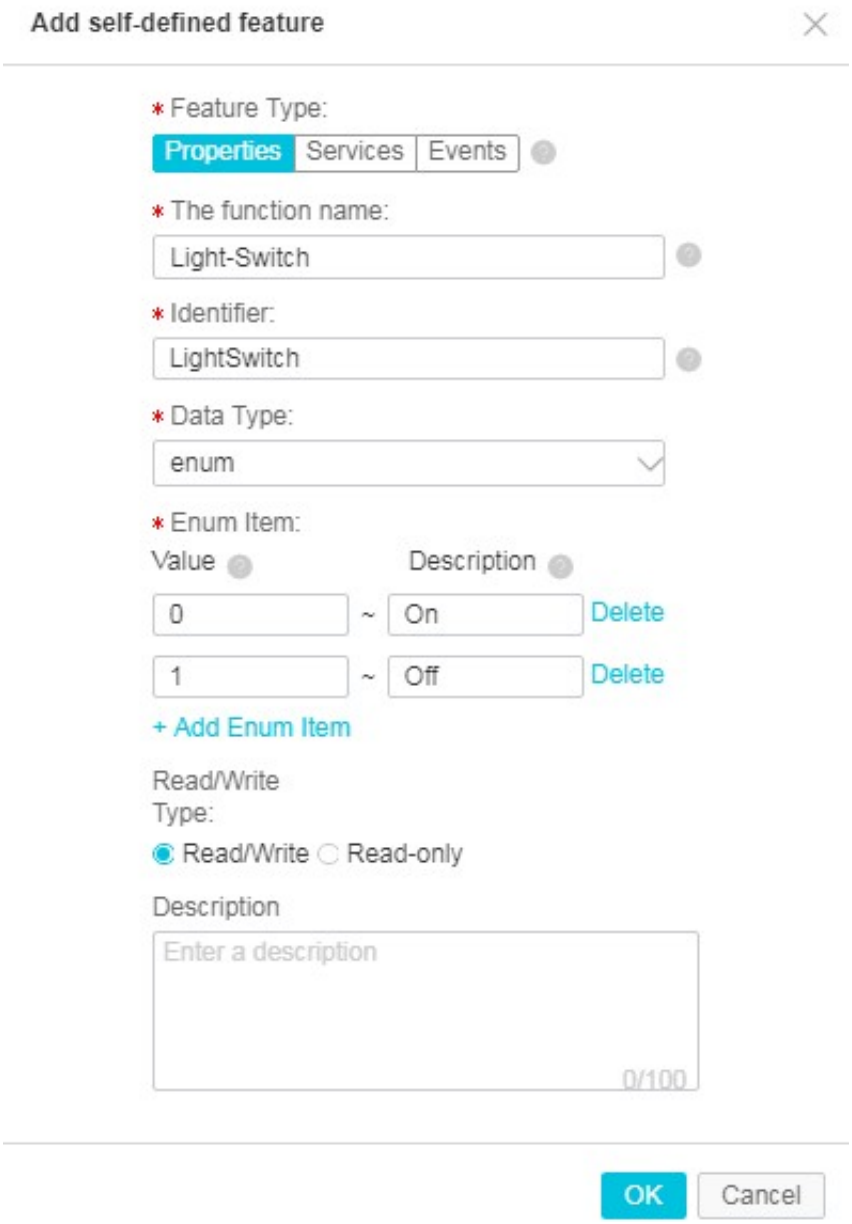
3. Define features for the product.

   a) In the product list, find the product and click **View**.

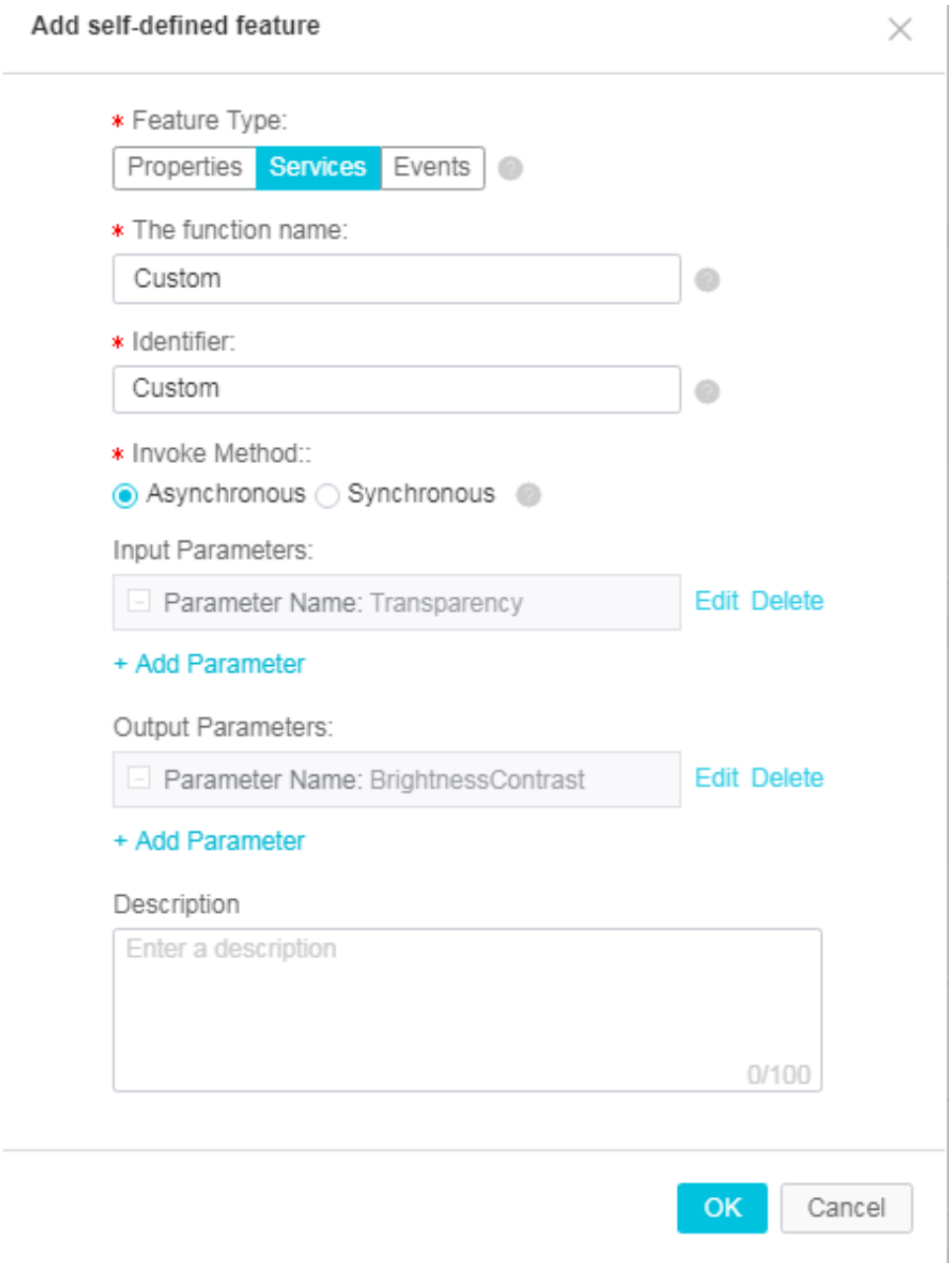
   b) On the product details page, click **Define Feature**.

   c) Click **Add Feature** corresponding to **Self-Defined Feature**.

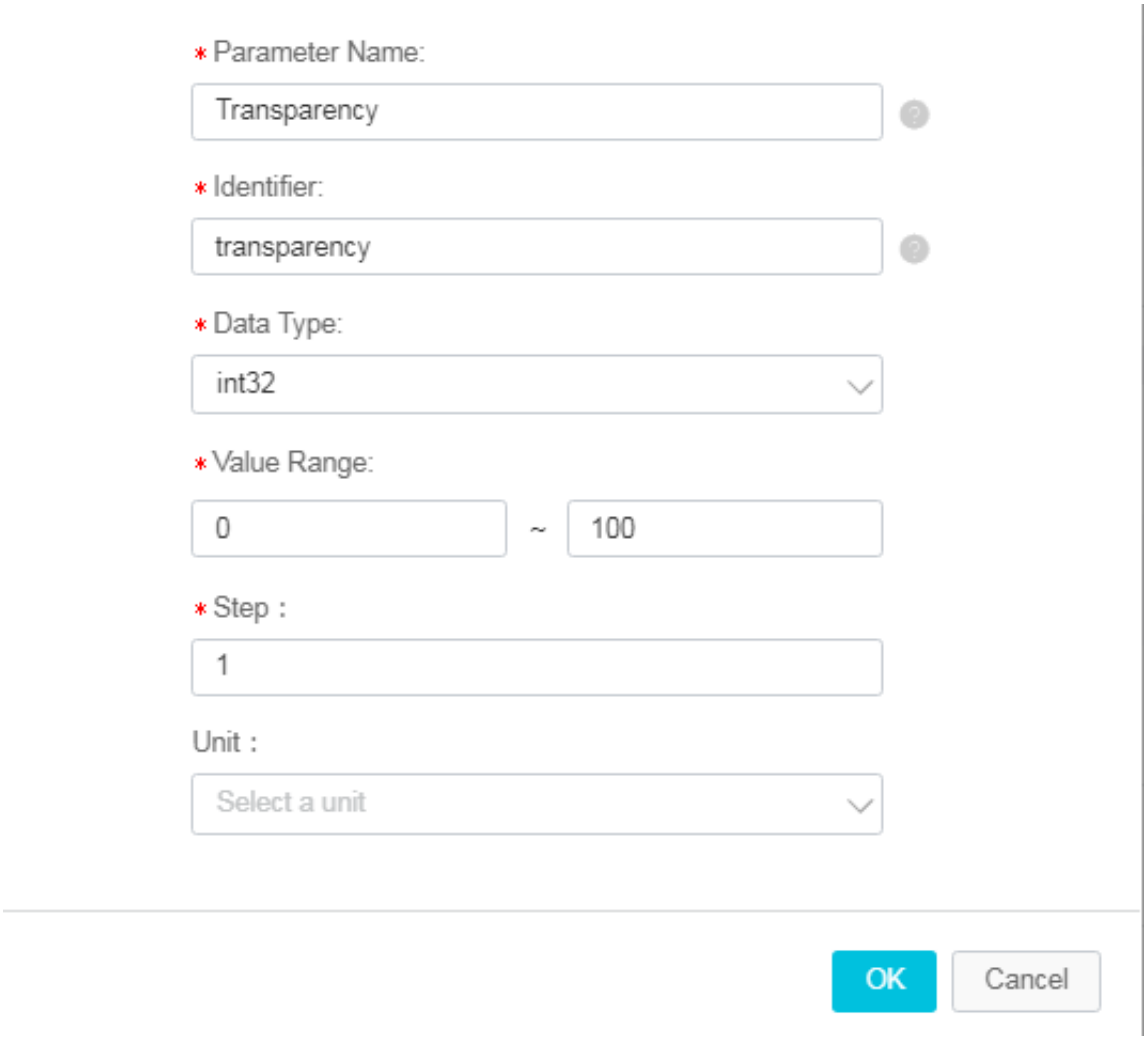
d) Define a property. In this example, a light switch property is defined. 0 indicates turning the light on and 1 indicates turning the light off.



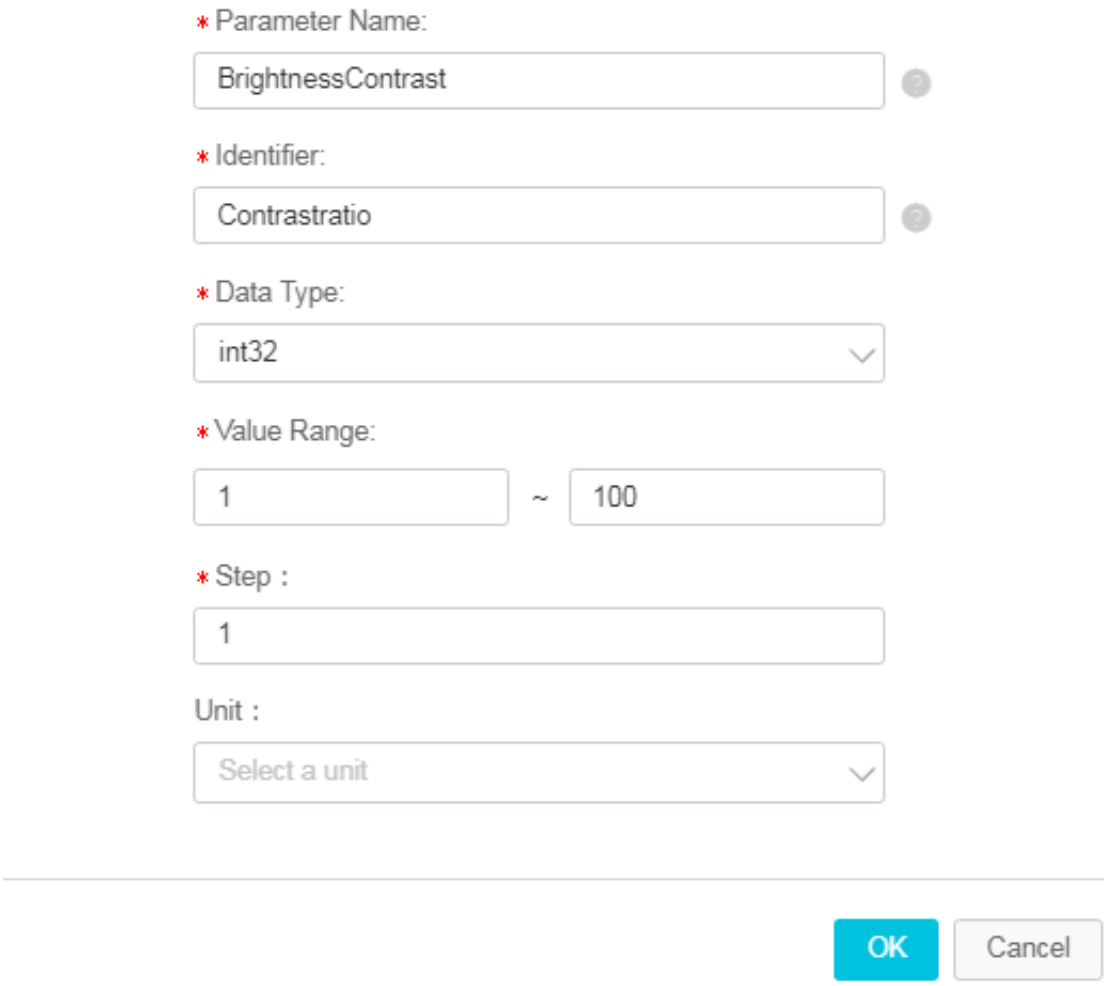
e) Define a service. For example, you can add an input parameter for adjusting the brightness of the bulb, and add an output parameter for the bulb to report the brightness contrast between the bulb and the room environment.



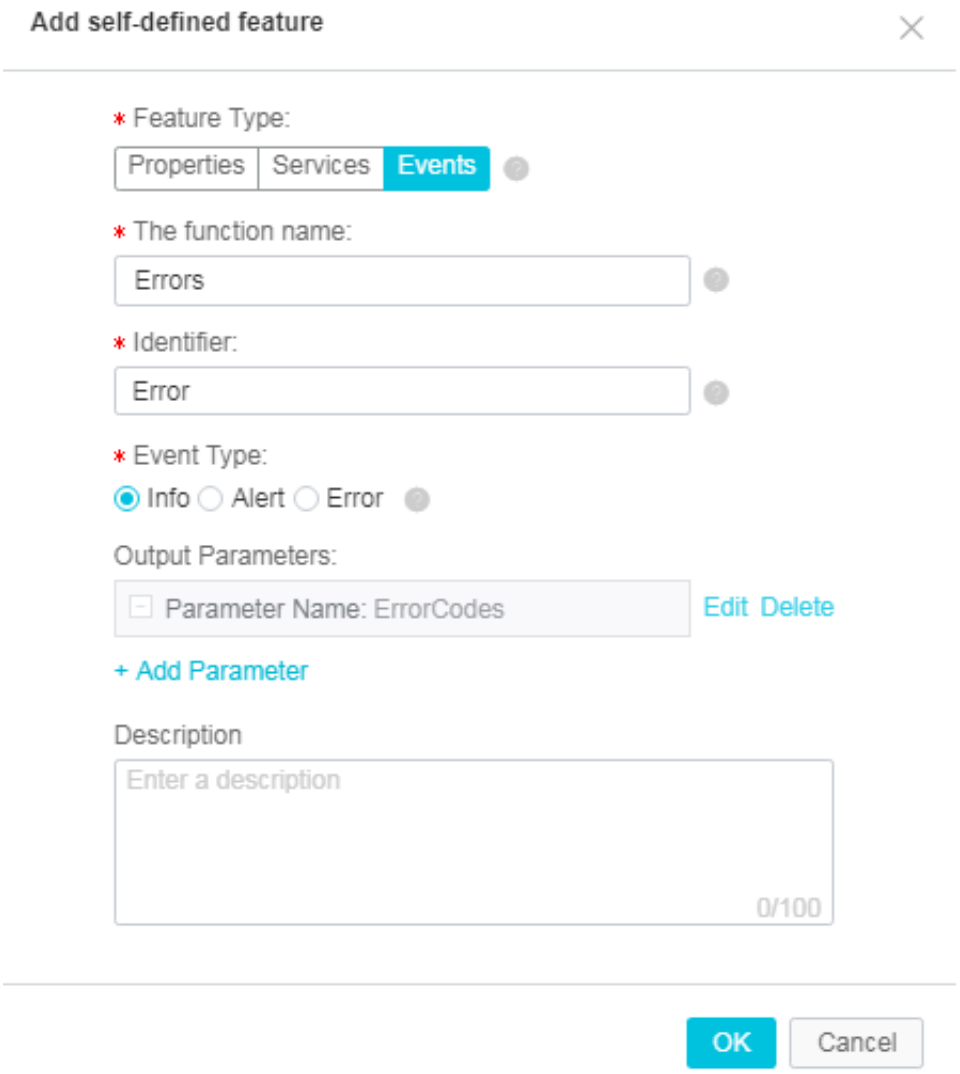
The following figure shows an example of input parameter.



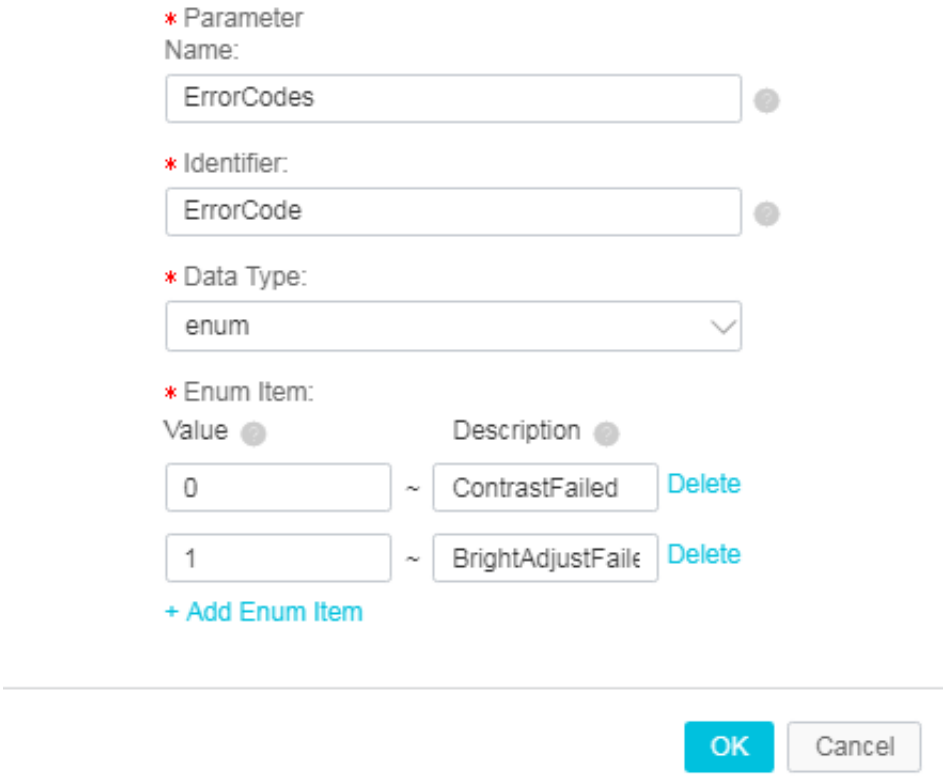
The following figure shows an example of output parameter.



f) Define an event. You can define an event for devices to report errors.



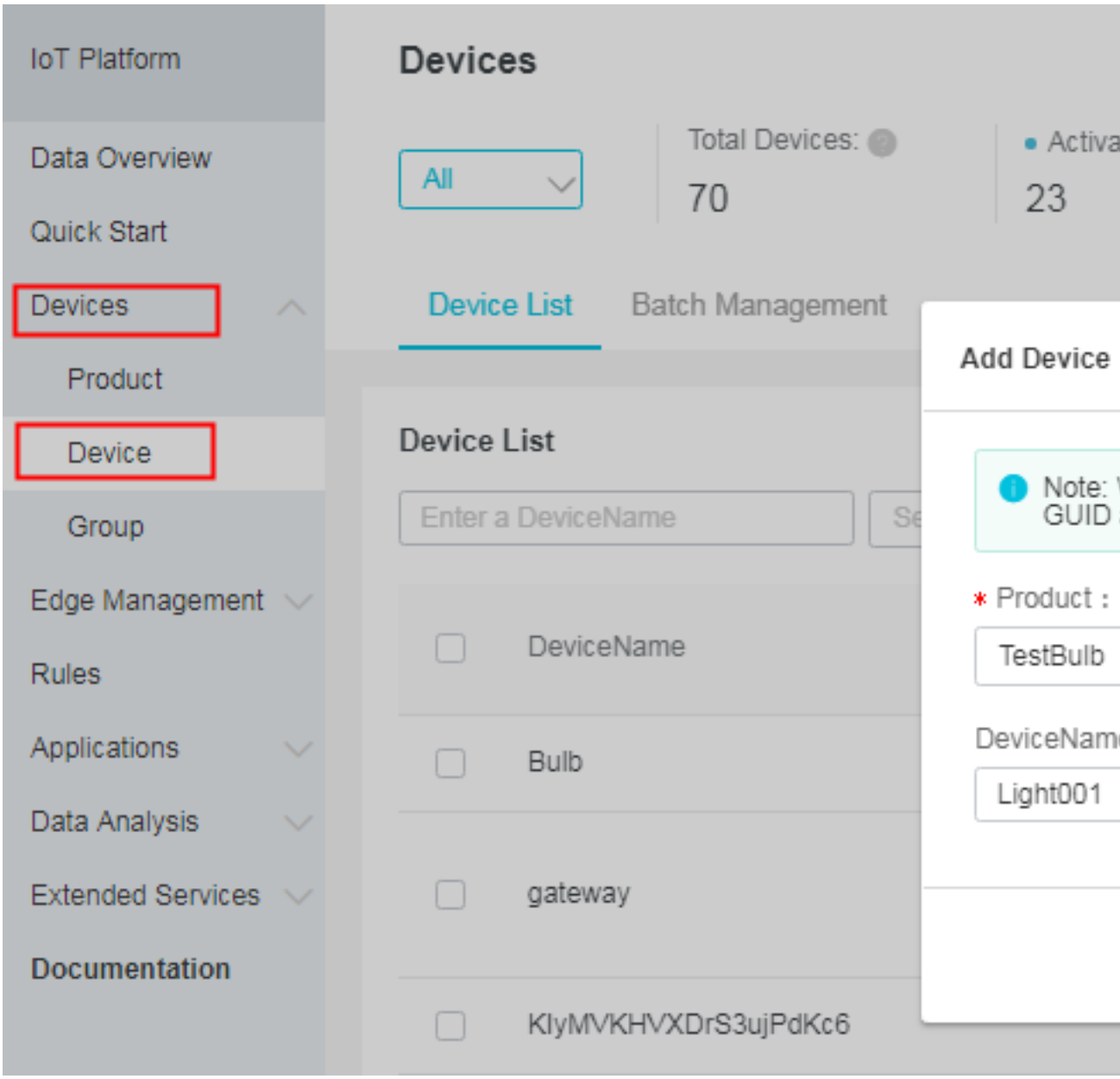
The following figure shows an example of output parameter.



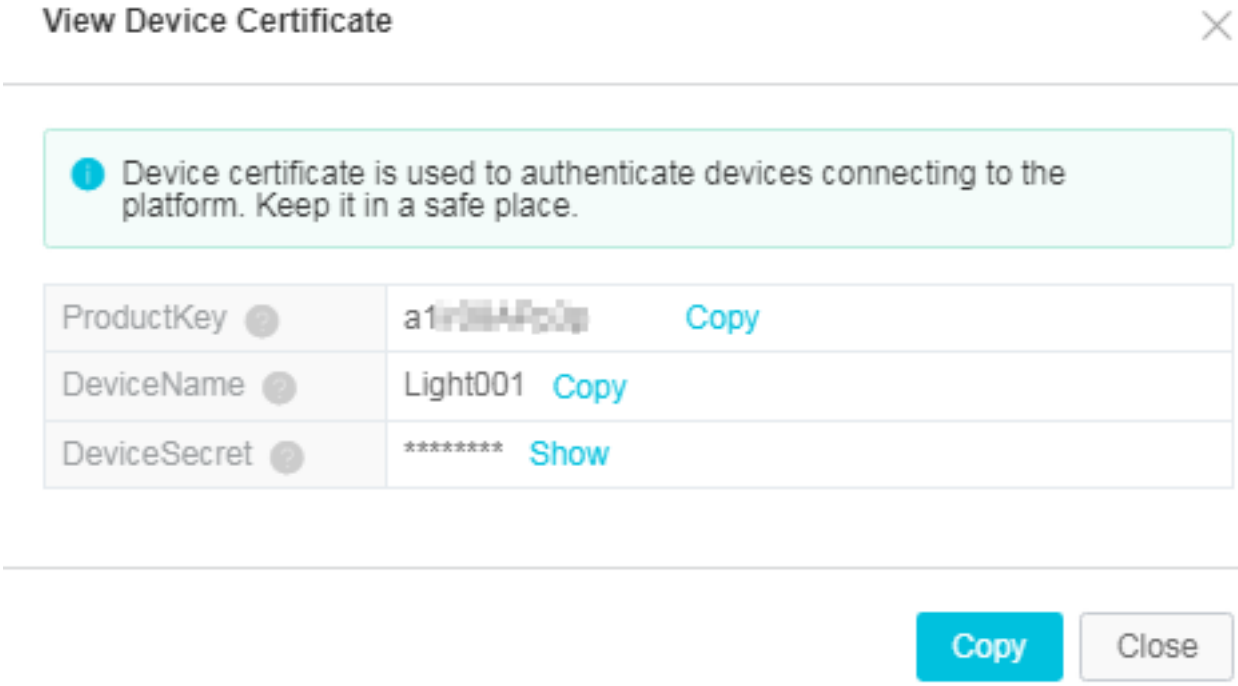
4. Create a device.

   a) In the left-side navigation pane, click **Devices** > **Device**.

   b) On the device management page, click **Add Device**. Select a product to which the device to be created belongs, and then enter a name for the device (DeviceName). Click **OK**.



c) Save the device certificate information. The certificate information includes ProductKey, DeviceName, and DeviceSecret. Keep this information confidential, because it is the certificate that will be used for device authentication when the device is connecting to IoT Platform.


View Device Certificate
Device certificate is used to authenticate devices connecting to the platform. Keep it in a safe place.
ProductKey
a1
Copy
DeviceName
Light001
Copy
DeviceSecret
****
Show
Copy
Close

2 Establish a connection between a device and IoT Platform

Alibaba Cloud IoT Platform provides device SDKs that allow devices to connect to IoT Platform. This article uses a sample program provided by IoT Platform to introduce how to connect the device to IoT Platform using the provided SDK.

Prerequisites

- The SDK used in this example is a C SDK for Linux system. We recommend that you develop this SDK on Ubuntu16.04 (64-bit)
- Software used in the development of the SDK: `make-4.1`, `git-2.7.4`, `gcc-5.4.0`, `gcov -5.4.0`, `lcov-1.12`, `bash-4.3.48`, `tar-1.28`, and `mingw-5.3.1` Using the following command to install the software:

  ```
  apt-get install -y build-essential make git gcc
  ```

Procedure

1. Log on to your Linux VM instance.
2. Download the C SDK 2.3.0.

   ```
   wget https://github.com/aliyun/iotkit-embedded/archive/v2.3.0.zip?spm=a2c4g. 11186623.2.13.1f41492b5WHpzV&file=v2.3.0.zip
   ```

3. Use the unzip command to extract files from the package.
4. Open the demo program

   ```
   vi iotkit-embedded-2.3.0/examples/linkkit/linkkit_example_solo.c
   ```

5. Change the values of ProductKey, DeviceName, and DeviceSecret in the demo to be your device certificate information, and then save the file.

   See the following example:

   ```
   // for demo only
   #define PRODUCT_KEY      "a1I1nn8vPf4"
   #define DEVICE_NAME      "Light00"
   #define DEVICE_SECRET    "n27gKXTxrUx*********QZEmoUX8TceM"
   ```

6. In the top level directory, use make command to compile the sample program.

   ```
   $ make distclean
   ```

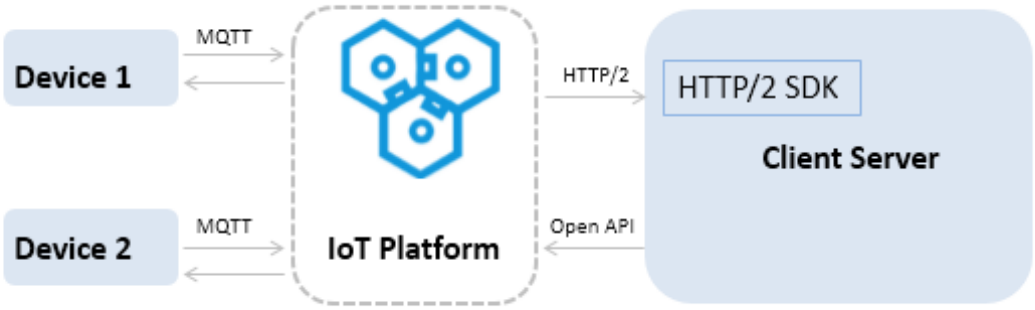
```
$ make
```

7. Run the sample program to connect the device to IoT Platform. In the IoT Platform console, you see that the device status is online, indicating that the device has been connected to IoT Platform successfully.

   Once the device has been connected to IoT Platform, it automatically report messages to IoT Platform. You see the device logs for message contents.

3 Servers subscribe to device messages

When devices are connected to IoT Platform, they report data to the platform. Data in the platform can be pushed to your server through a HTTP/2 channel. Set the service subscription through HTTP/2 and configure for HTTP/2 SDKs. You can then connect your server to an HTTP/2 SDK and the server can receive device data.



Procedure

1. You configure service subscription for your products in the *IoT Platform console*.
   a) On the **Products** page, find the product for which you want to configure the service subscription and click **View**.
   b) On the product details page, click **Service Subscription**, and then click **Set**.
   c) Select the types of notifications which you want to push to your server (HTTP/2 SDK) and click **Save**.
      - Device Upstream Notification: Includes the custom data, property data, and event data that are reported by devices, responses to property setting requests, and responses to service callings.
      - Device Status Change Notification: Indicates the notifications that are sent by the system when the statuses of devices change. For example, the notifications upon devices going online or going offline.

   The subscription configuration in the console takes effect about one minute after the configuration is completed.
2. Connect to the HTTP/2 SDK.

   If you use Apache Maven to manage Java projects, you add the following dependency content to the pom.xml file.

Note:

Currently, only SDKs in Java 8 are supported.

```
<dependency>
    <groupId>com.aliyun.openservices</groupId>
    <artifactId>iot-client-message</artifactId>
    <version>1.1.3</version>
</dependency>

<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>3.7.1</version>
</dependency>
```

3. Connect the SDK and IoT Platform using the AccessKey information of your Alibaba Cloud account for identity authentication.

```
// Your AccessKey
        String accessKey = "xxxxxxxxxxxxxx";
        // Your account AccessKeySecret
        String accessSecret = "xxxxxxxxxxxxxx";
        // Region ID of your IoT Platform service
        String regionId = "cn-shanghai";
        // Your account ID.
        String uid = "xxxxxxxxxxx";
        // endPoint:  https://${uid}.iot-as-http2.${region}.aliyuncs
.com
        String endPoint = "https://" + uid + ".iot-as-http2." +
regionId + ".aliyuncs.com";

        // Connection configuration
        Profile profile = Profile.getAccessKeyProfile(endPoint,
regionId, accessKey, accessSecret);

        // Construct the client
        MessageClient client = MessageClientFactory.messageClient(
profile);

        // Receive data
        client.connect(messageToken -> {
            Message m = messageToken.getMessage();
            System.out.println("receive message from " + m);
            return MessageCallback.Action.CommitSuccess;
        });
```

The parameters are introduced as follows:

- **accessKey** and **accessSecret**: Log on to the console, move the pointer to your account image, and click **AccessKey**. You are directed to the **User Management** page and you can create a new AccessKey or view the AccessKey ID and AccessKey Secret of an existing AccessKey on this page.

- **uid**: Log on to the console, move the pointer to your account image, and click **Security Settings**. You are directed to the **Account Management** page and you can view your account ID on this page.
- **regionId**: The region of your IoT Platform service. For information about RegionId expressions, see *Regions and zones*.

4. Test and make sure that the HTTP/2 SDK can successfully receive messages from devices.

   If the message is successfully received, you can obtain the following data from the message callback of the SDK.

Parameter	Description
messageId	A 19-bit message ID generated by IoT Platform .
topic	The topic from which the message is sent. For example, `/a1wmrZPO8o9/cbgiotkJ4O4WW59ivysa/data`. If the message is a device status change notification, the topic format is `/as/mqtt/status/${productKey}/${deviceName}`.
payload	• The binary data that a Pro Edition device publishes to a topic. • If the message is a device status change notification, the format is as follows: `{ "status":"online" (or offline), //the device status "productKey":"xxxxxxxxxx", //In the above example, the ProductKey is a1wmrZPO8o9 "deviceName":"xxxxxxxxxx", //In the above example, the DeviceName is cbgiotkJ4O4WW59ivysa "time":"2018-08-31 15:32:28.205", //The time when the notification is sent "utcTime":"2018-08-31T07:32:28.205Z",//The UTC time when the notification is sent "lastTime":"2018-08-31 15:32:28.195", //The time when the last message communication occurred before the status change "utcLastTime":"2018-08-31T07:32:28.195Z",//The UTC time when the last message communication occurred before the status change "clientIp":"xxx.xxx.xxx.xxx" //the Internet IP address of the device. }` **Note:** We recommend that you maintain your device status according to the value of the parameter lastTime.

Parameter	Description
generateTime	The timestamp when the message is generated, in millisecond.
qos	• 0: The message is only pushed one time. • 1: The message is pushed at least one time.