IoT Device Management

Quick Start

Issue	01
Date	2019-06-25

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei 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.

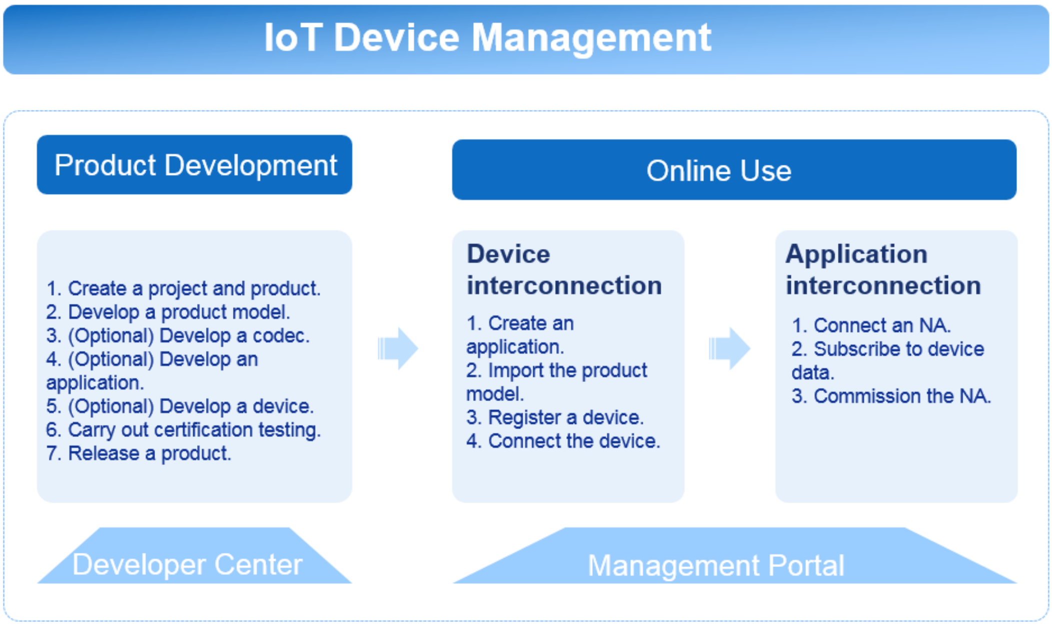
The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Service Usage Process

The following figure shows the process of using IoT Device Management. The process is divided into product development and online use phases.

- During product development, you develop a product on the Developer Center. Product development is the prerequisite for connecting physical devices to IoT Device Management.
- During online use, you connect physical devices and network applications (NAs) to IoT Device Management through the Management Portal for remote device monitoring and control.


IoT Device Management
Product Development
Online Use
1. Create a project and product.
2. Develop a product model.
3. (Optional) Develop a codec.
4. (Optional) Develop an application.
5. (Optional) Develop a device.
6. Carry out certification testing.
7. Release a product.
Device interconnection
1. Create an application.
2. Import the product model.
3. Register a device.
4. Connect the device.
Application interconnection
1. Connect an NA.
2. Subscribe to device data.
3. Commission the NA.
Developer Center
Management Portal

Phase	Key Operation	Description
Product development	**Create a project and product.**	You can develop IoT products and applications through projects on the Developer Center. A product is a collection of devices with the same properties and is an abstract model of physical devices on the IoT platform.

Phase	Key Operation	Description
	Develop a product model.	A product model is also called a profile file. It describes the capabilities and features of a device and is an abstract model of the device on the IoT platform.
	(Optional) Develop a codec.	The codec parses hexadecimal data reported by devices into JSON data based on the defined product model. It also converts JSON data delivered by the IoT platform into hexadecimal data and delivers this data to devices. NOTE If data reported by a device is in JSON format, codec development is not required. For example, a device that has AgentLite integration reports data in JSON format.
	(Optional) Develop an application.	You can create applications based on SDKs and APIs provided by the IoT platform to remotely monitor and control devices. NOTE If you do not want to develop applications for remote device monitoring and control, you can use the Management Portal of IoT Device Management instead.
	(Optional) Develop a device.	You can develop devices based on the AgentLite, LiteOS, or NB-IoT modules, allowing the devices to connect to the IoT platform, report data, and receive commands. NOTE If a device can connect to the IoT platform, you only need to set the IP address and port number for device connection.
	Carry out self-service testing.	During self-service testing, end-to-end test cases are provided to help developers test basic device capabilities, such as data reporting and command delivery.
	Release a product.	After the profile file and codec meet commercial use requirements, you can release the product to the Product Center.
Device interconnection	**Create an application.**	On the Management Portal, an application is created to manage devices and NAs. Application creation is prerequisite to device and NA access.
	Import the product model.	Through the Management Portal, import the product model released in the Product Center.
	Register a device.	Register a device on the Management Portal and deliver information required for device connection.
	Connect the device.	After a physical device is powered on, it can connect to the IoT platform for data collection and device management.
Application interconnection	**Connect an NA.**	An NA is connected to the IoT platform for remote device management.

Phase	Key Operation	Description
	Subscribe to device data.	An NA can subscribe to device data. If the subscribed-to data changes, the IoT platform pushes this data to the NA.
	Commission the NA.	Confirm that the data reported can be pushed to the NA and that the NA can remotely control the device.

2 Experiencing IoT Platform Integration

2.1 Before You Start

This document uses a smart street lamp as an example to describe how to use the Development Center's device and application simulators to simulate physical devices and applications during data reporting and command delivery.

This document assumes that the street lamp reports a data message carrying its light intensity (Light_Intensity) and status (Light_Status) and that the command (SWITCH_LIGHT) is used to remotely control the street lamp status. The data reported is in hexadecimal format.

2.2 Subscribing to the Developer Center and Creating a Project

Scenarios

When using the IoT platform cloud service for the first time, you must subscribe to the Developer Center. You can use the Developer Center to develop product models and codecs online. The Developer Center also provides device and application simulators for commissioning.

Prerequisites

You have registered a HUAWEI CLOUD account on the **registration page**.

Procedure

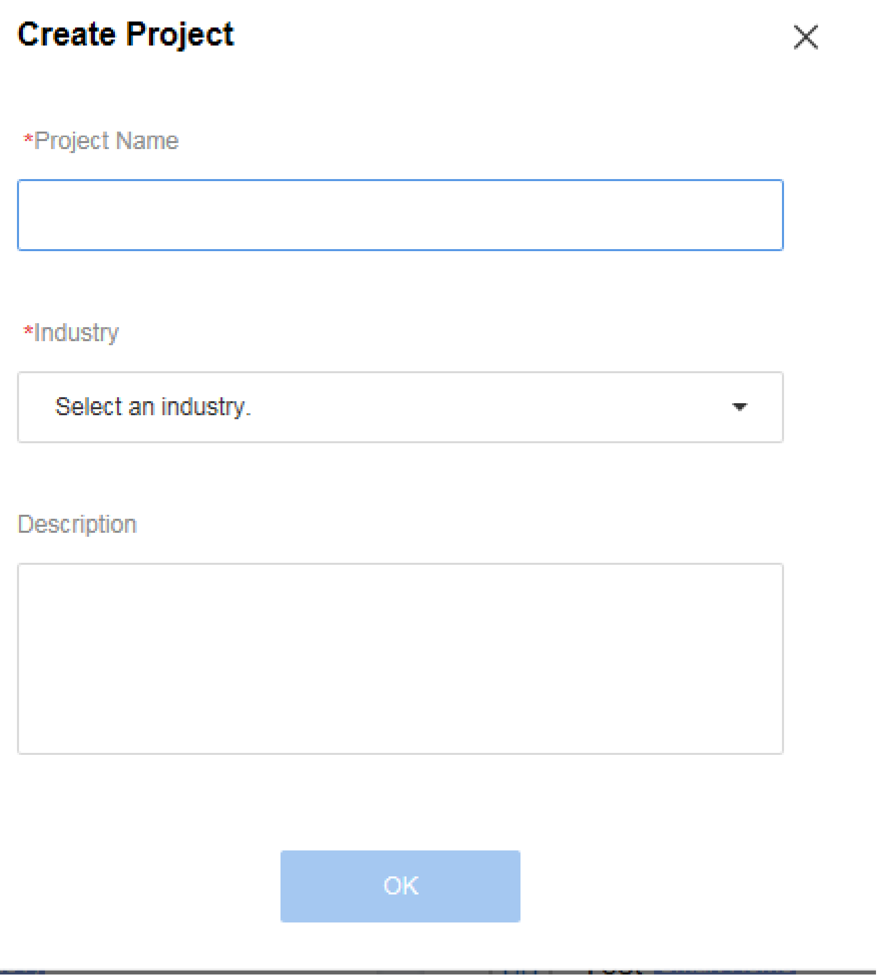
Step 1 Log in to HUAWEI CLOUD and visit IoT Device Management.

Step 2 On the home page, click **Developer Center**.

Step 3 Enter relevant information. This document uses the smart street lamp solution as an example. Click **Subscribe Now**.

Step 4 (Optional) If you are using the Developer Center for the first time, click **Manufacturer** in the upper right corner, complete the manufacturer information, and return to the home page.

Step 5 To create a project, click + on the home page. In the dialog box displayed, enter the project name and industry, and click **OK**.



Create Project

*Project Name

*Industry

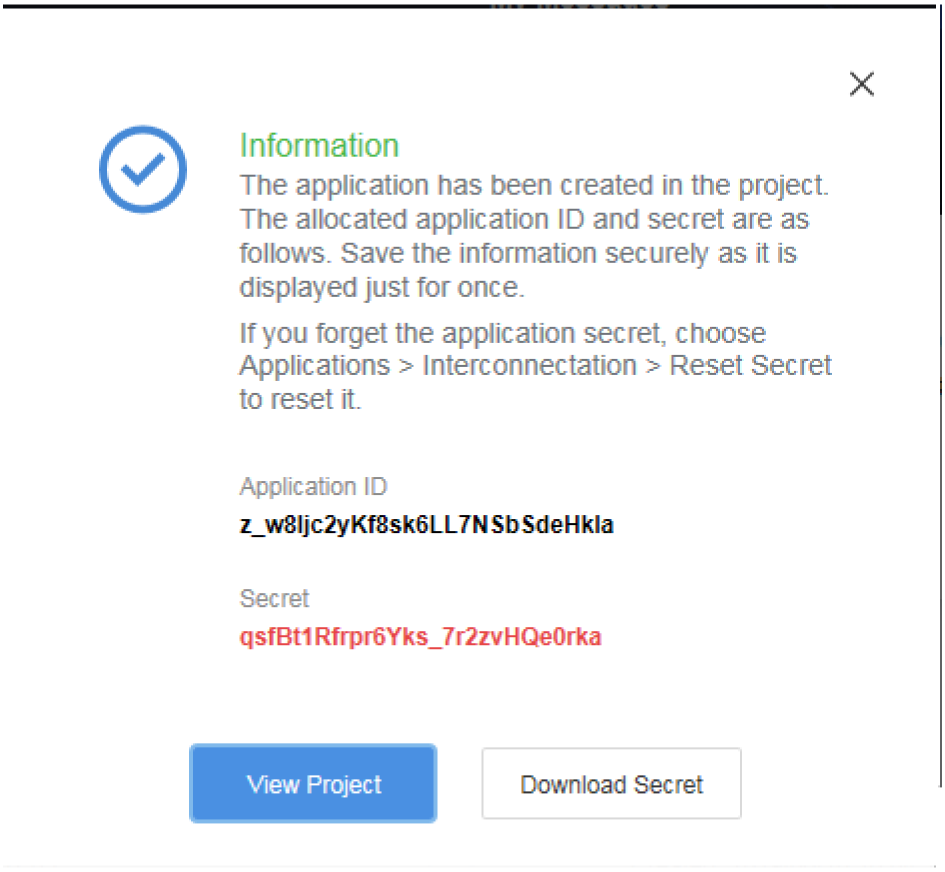
Select an industry.

Description

Step 6 After the project is created, its application ID and secret are generated. Download the secret to your local PC and keep it securely. Finally enter the project.

NOTE

A secret is used to authenticate NAs when they attempt to access the IoT platform. It is not displayed. Make sure to keep it secure. If you forget the secret, you can reset it on the **Interconnection** page.



Information

The application has been created in the project. The allocated application ID and secret are as follows. Save the information securely as it is displayed just for once.

If you forget the application secret, choose Applications > Interconnectation > Reset Secret to reset it.

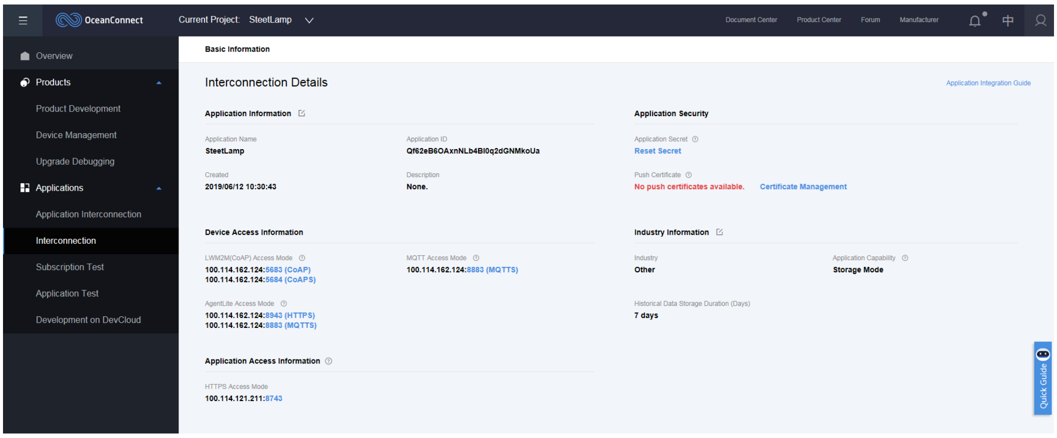
Application ID

z_w8ljc2yKf8sk6LL7NSbSdeHkla

Secret

qsfBt1Rfrpr6Yks_7r2zvHQe0rka

Step 7 After the project is created, you can view its basic information, such as device and application access information, and reset the secret on the **Interconnection** page.



Interconnection Details

Application Information

Application Name
SteetLamp

Application ID
Qf62eB6OAxnNLb4Bl0q2dGNMkoUa

Created
2019/06/12 10:30:43

Description
None.

Application Security

Application Secret
Reset Secret

Push Certificate
No push certificates available. Certificate Management

Device Access Information

LWM2M(CoAP) Access Mode
100.114.162.124:5683 (CoAP)
100.114.162.124:5684 (CoAPS)

MQTT Access Mode
100.114.162.124:8883 (MQTTS)

AgentLite Access Mode
100.114.162.124:8943 (HTTPS)
100.114.162.124:8883 (MQTTS)

Industry Information

Industry
Other

Application Capability
Storage Mode

Historical Data Storage Duration (Days)
7 days

Application Access Information

HTTPS Access Mode
100.114.121.211:8743

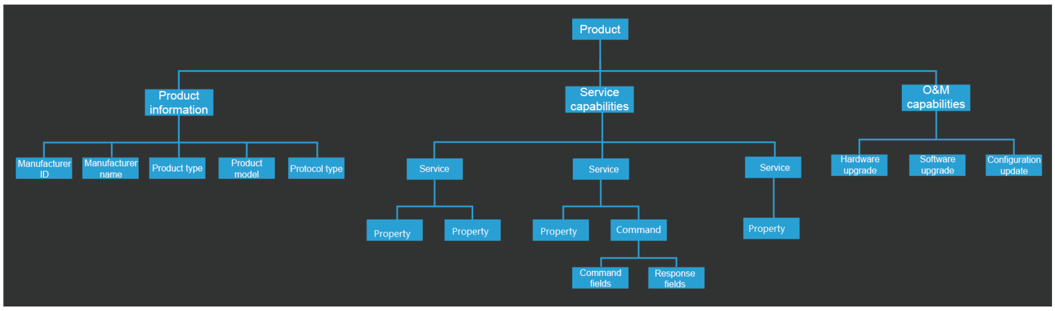
----End

2.3 Developing a Product Model Online

Scenarios

A product model (or profile file) is used to describe the capabilities and features of a device. Developers build an abstract model of a device by defining a profile file on the IoT platform, allowing it to understand the services, properties, and commands supported by the device, such as color, light intensity, and switches.

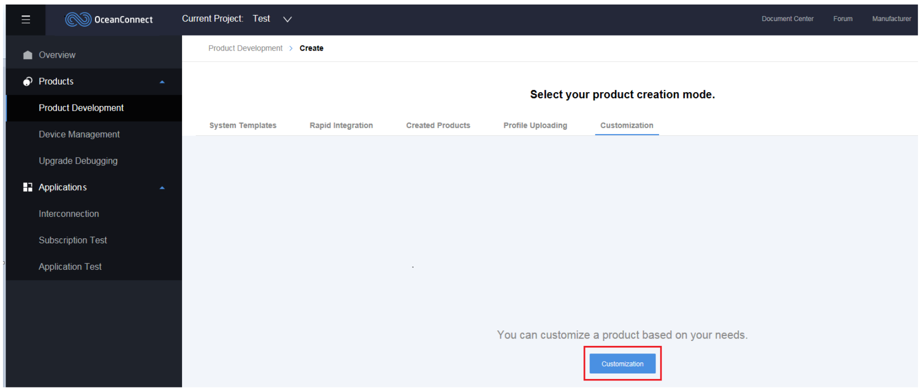
Figure 2-1 Profile file structure



This document uses a smart street lamp as an example. The street lamp reports a data message carrying its light intensity (Light_Intensity) and status (Light_Status). The command (SWITCH_LIGHT) can be used to remotely control the street lamp status.

Procedure

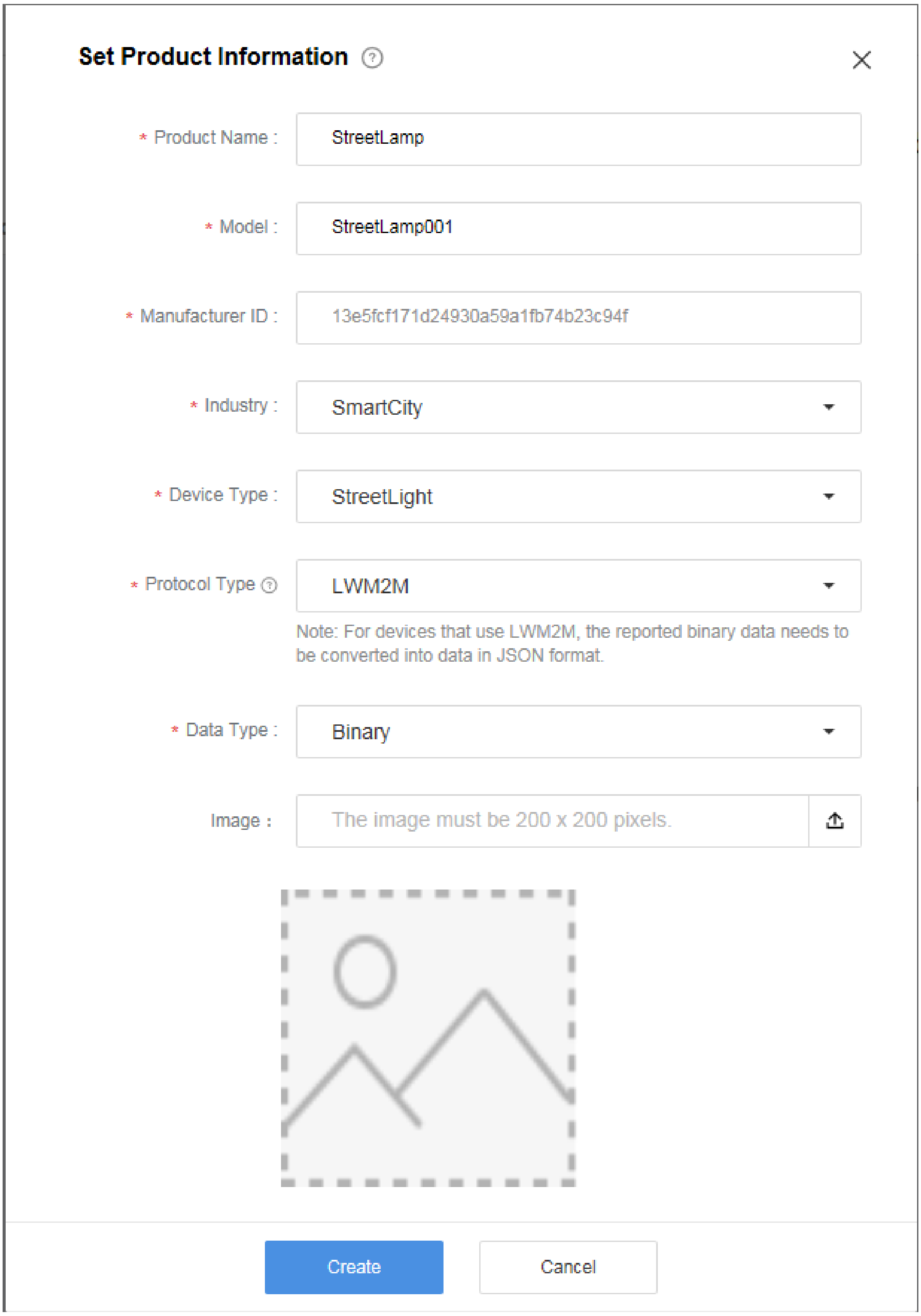
Step 1 On the **Product Development** page, click **Create Product**, click the **Customization** tab, and click **Customization**.



Select your product creation mode.

You can customize a product based on your needs.

Step 2 In the dialog box displayed, define the product information and click **Create**.



Set Product Information

* Product Name : StreetLamp

* Model : StreetLamp001

* Manufacturer ID : 13e5fcf171d24930a59a1fb74b23c94f

* Industry : SmartCity

* Device Type : StreetLight

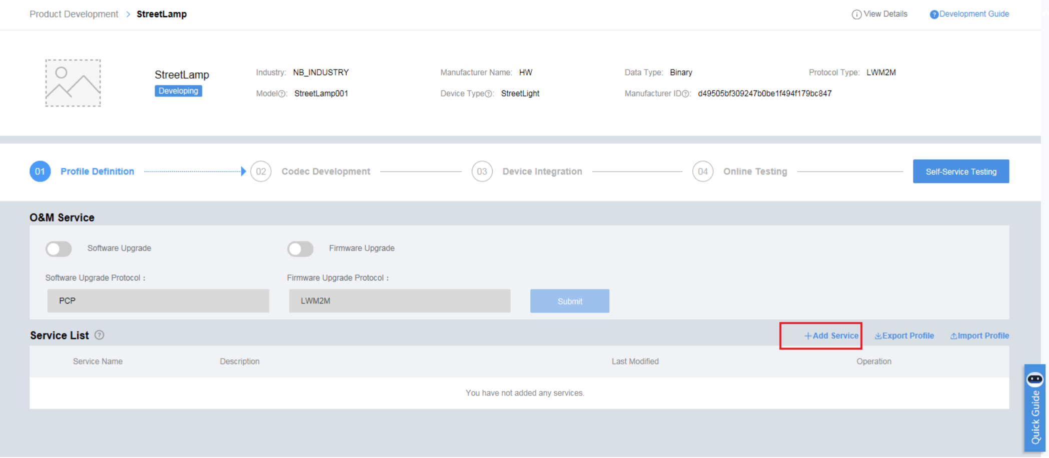
* Protocol Type : LWM2M

Note: For devices that use LWM2M, the reported binary data needs to be converted into data in JSON format.

* Data Type : Binary

Image : The image must be 200 x 200 pixels.

Step 3 On the **Profile Definition** page, click **Add Service**.



StreetLamp
Developing

Industry: NB_INDUSTRY
Model: StreetLamp001
Manufacturer Name: HW
Device Type: StreetLight
Data Type: Binary
Manufacturer ID: d49505bf309247b0be1f494f179bc847
Protocol Type: LWM2M

O&M Service

Software Upgrade
Firmware Upgrade

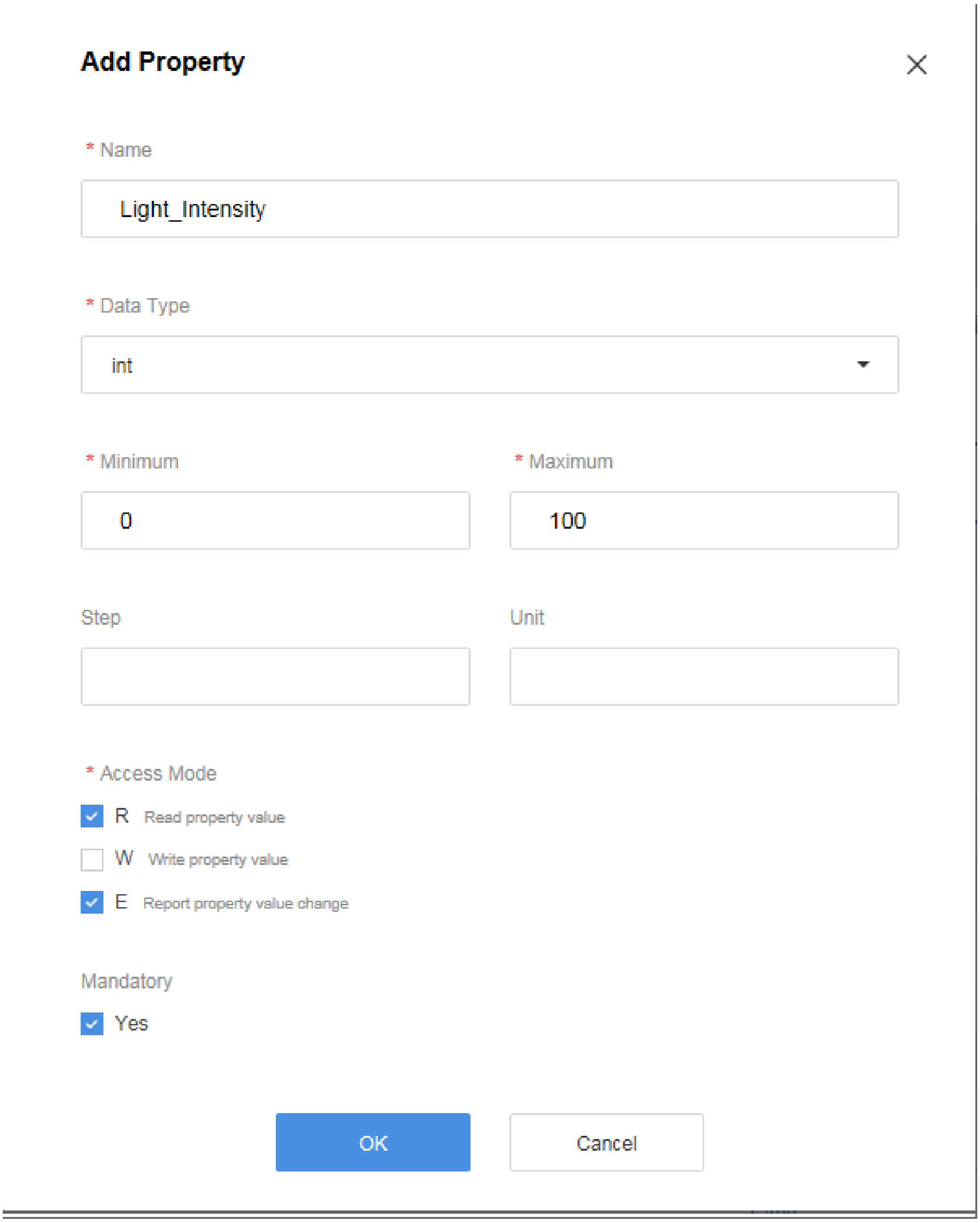
Software Upgrade Protocol : PCP
Firmware Upgrade Protocol : LWM2M

Service List

Service Name	Description	Last Modified	Operation
You have not added any services.			

Step 4 Define the properties of light intensity (Light_Intensity) and status (Light_Status). Set **Service Name** to **StreetLamp**, and click **Add** under **Property List**.

Step 5 Define a property as follows: Set **Property Name** to **Light_Intensity**, **Data Type** to **int**, **Minimum** to **0**, and **Maximum** to **100**. Click **OK**.



Add Property

* Name: Light_Intensity

* Data Type: int

* Minimum: 0

* Maximum: 100

Step:

Unit:

* Access Mode

☑ R Read property value

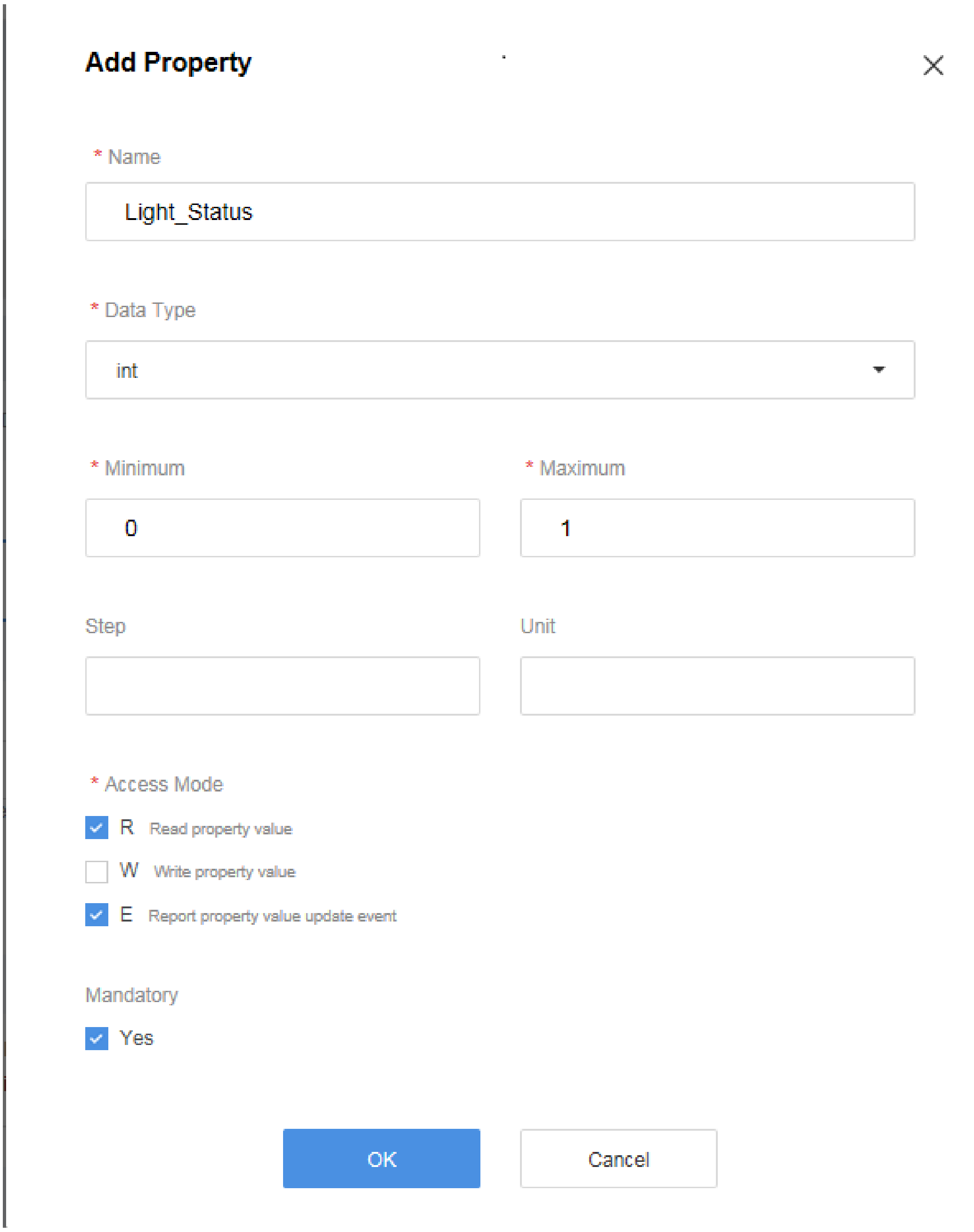
□ W Write property value

☑ E Report property value change

Mandatory

☑ Yes

Step 6 Click **Add** to define another property as follows: Set **Property Name** to **Light_Status**, **Data Type** to **int**, **Minimum** to **0** (off), and **Maximum** to **1** (on). Click **OK**.



Add Property

* Name: Light_Status

* Data Type: int

* Minimum: 0

* Maximum: 1

Step:

Unit:

* Access Mode

☑ R Read property value

□ W Write property value

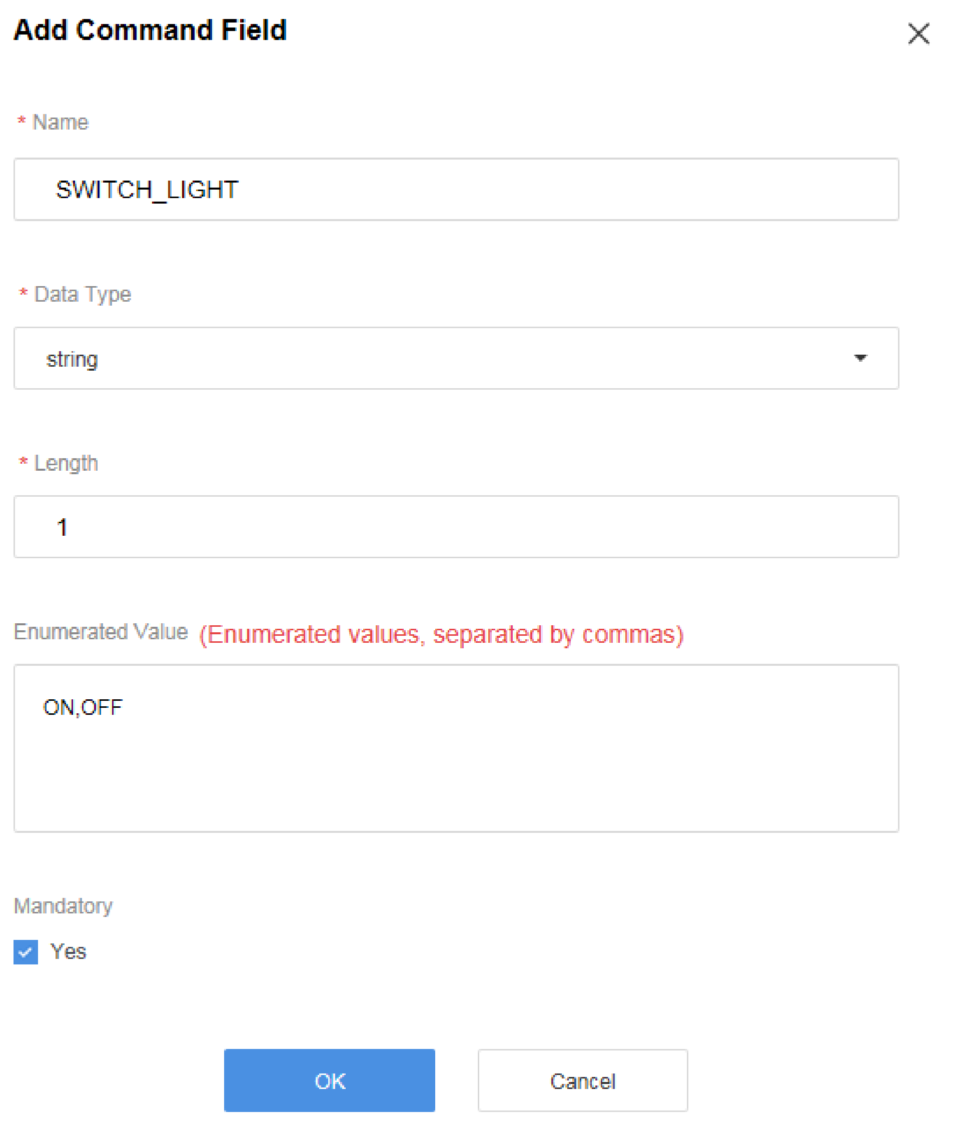
☑ E Report property value update event

Mandatory

☑ Yes

Step 7 Define the command used to remotely control the switch status. Click **Add** under **Command List**. In the dialog box displayed, set **Command Name** to **SWITCH_LIGHT**.

Step 8 Click **Add** under **Command Fields**, and define a command field as follows: Set **Property Name** to **SWITCH_LIGHT**, **Data Type** to **string**, **Length** to **1**, and **Enumerated Value** to **ON,OFF**. Click **OK**.



Add Command Field

* Name

SWITCH_LIGHT

* Data Type

string

* Length

1

Enumerated Value (Enumerated values, separated by commas)

ON,OFF

Mandatory

☑ Yes

Step 9 Click **Save** to complete the creation of the profile file.

----End

2.4 Developing a Codec Online

Scenarios

Generally, devices report hexadecimal data to reduce power consumption. The codec converts the hexadecimal data into JSON data based on properties defined in the profile file so that the IoT platform and NAs can identify the data. In addition, when an NA remotely delivers a control command, the IoT platform converts the command in JSON format into hexadecimal and delivers the data to the device.

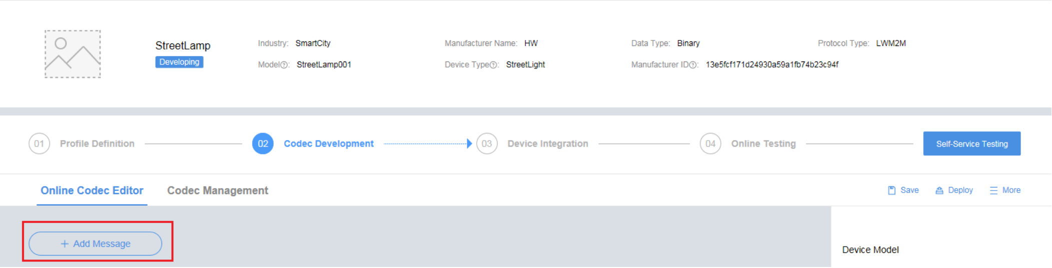
NOTE

If a device (for example, an AgentLite device) reports data in JSON format, you do not need to define a codec.

Procedure

Step 1 On the **Product Development** page, enter the created product and click **Codec Development**.

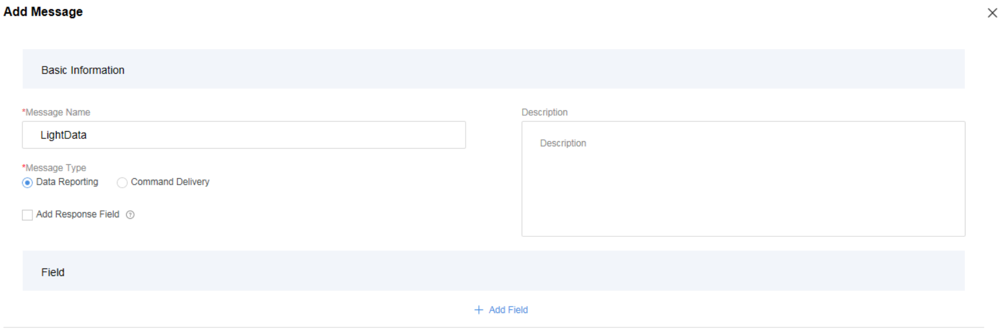
Step 2 On the **Online Codec Editor** tab page, click **Add Message**.



StreetLamp
Developing

Industry: SmartCity
Model: StreetLamp001
Manufacturer Name: HW
Device Type: StreetLight
Data Type: Binary
Manufacturer ID: 13e5fcf171d24930a59a1fb74b23c94f
Protocol Type: LWM2M

Step 3 Set **Message Name** to **LightData**, select **Data Reporting** for **Message Type**, and click **Add Field**.



Add Message

Basic Information

*Message Name: LightData

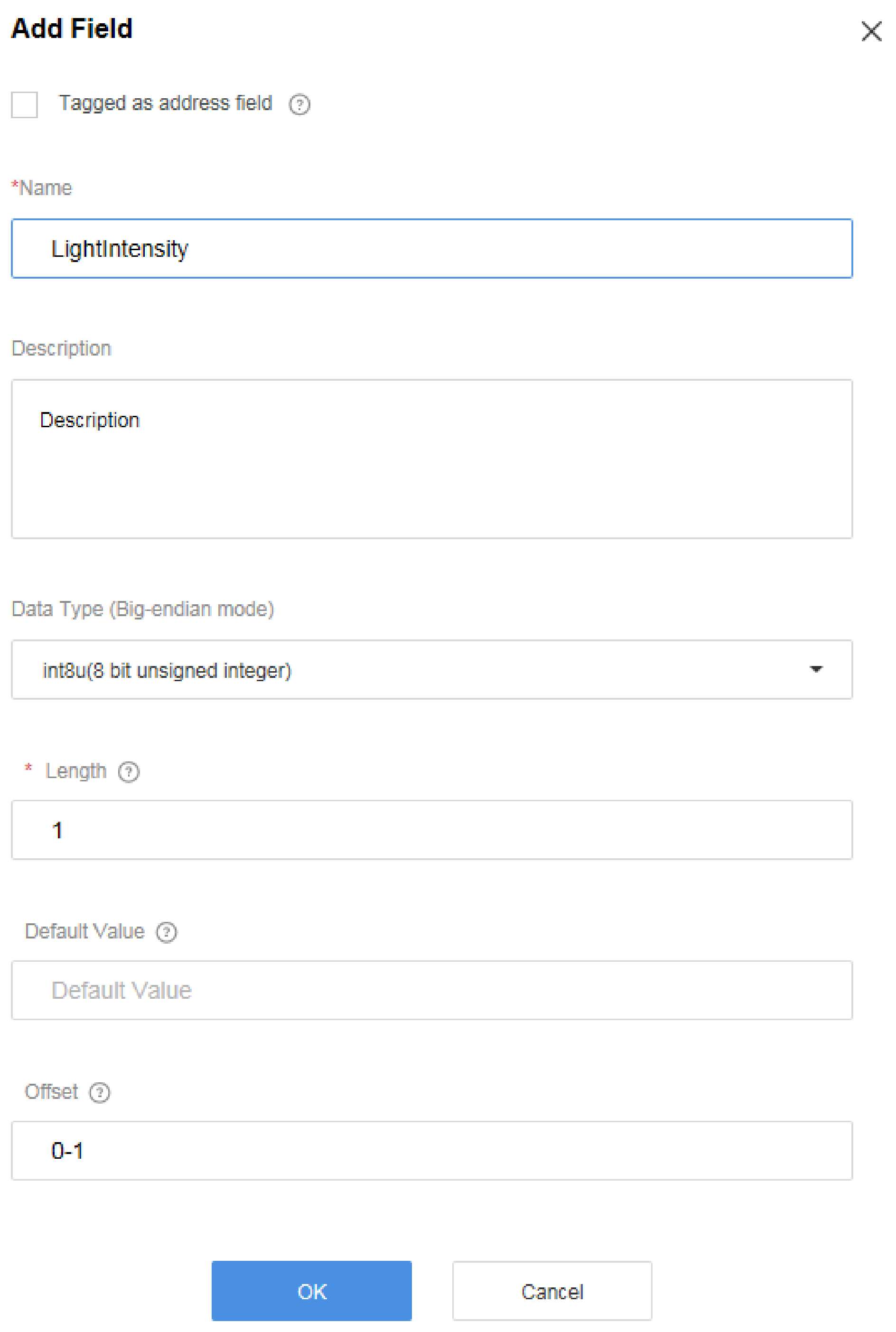
*Message Type: ● Data Reporting ○ Command Delivery

□ Add Response Field

Description: Description

Field

Step 4 In the dialog box displayed, set **Name** to **LightIntensity**, **Data Type** to **int8u(8 bit unsigned integer)**, and **Length** to **1**, and click **OK**.



Add Field

□ Tagged as address field

*Name: LightIntensity

Description: Description

Data Type (Big-endian mode): int8u(8 bit unsigned integer)

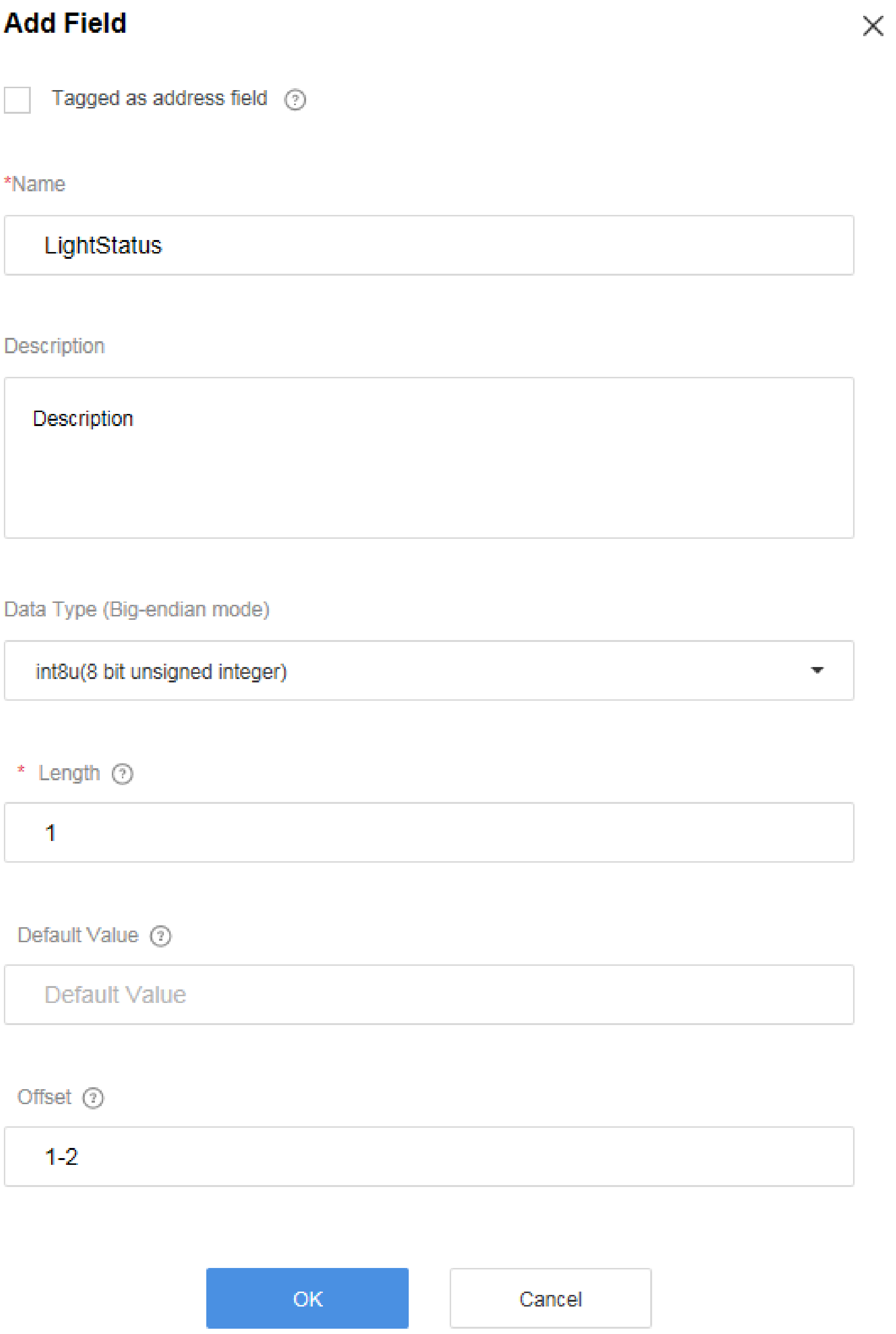
* Length: 1

Default Value: Default Value

Offset: 0-1

Step 5 Click **Add Field** again to add switch status data reported.

Step 6 In the dialog box displayed, set **Name** to **LightStatus**, **Data Type** to **int8u(8 bit unsigned integer)**, and **Length** to **1**, and click **OK**.



Add Field

□ Tagged as address field

*Name: LightStatus

Description: Description

Data Type (Big-endian mode): int8u(8 bit unsigned integer)

* Length: 1

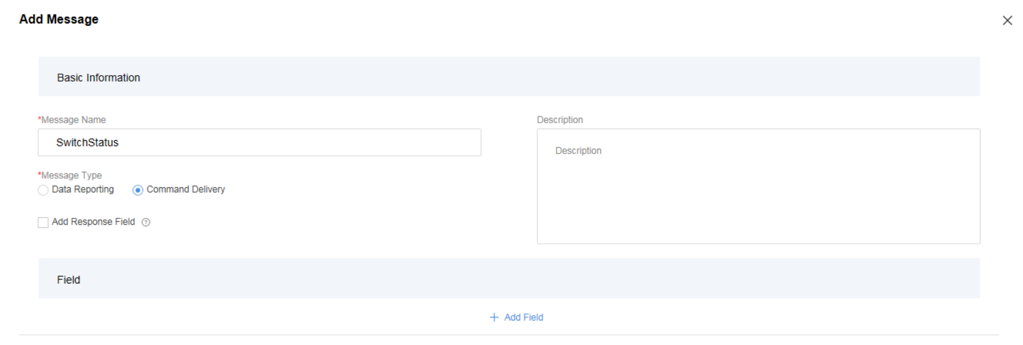
Default Value: Default Value

Offset: 1-2

Step 7 Click **OK** to complete the codec for data reporting messages.

Step 8 Click **Add Message** again to develop a codec for command delivery messages.

Step 9 Set **Message Name** to **SwitchStatus**, select **Command Delivery** for **Message Type**, and click **Add Field**.



Add Message

Basic Information

*Message Name: SwitchStatus

Description: Description

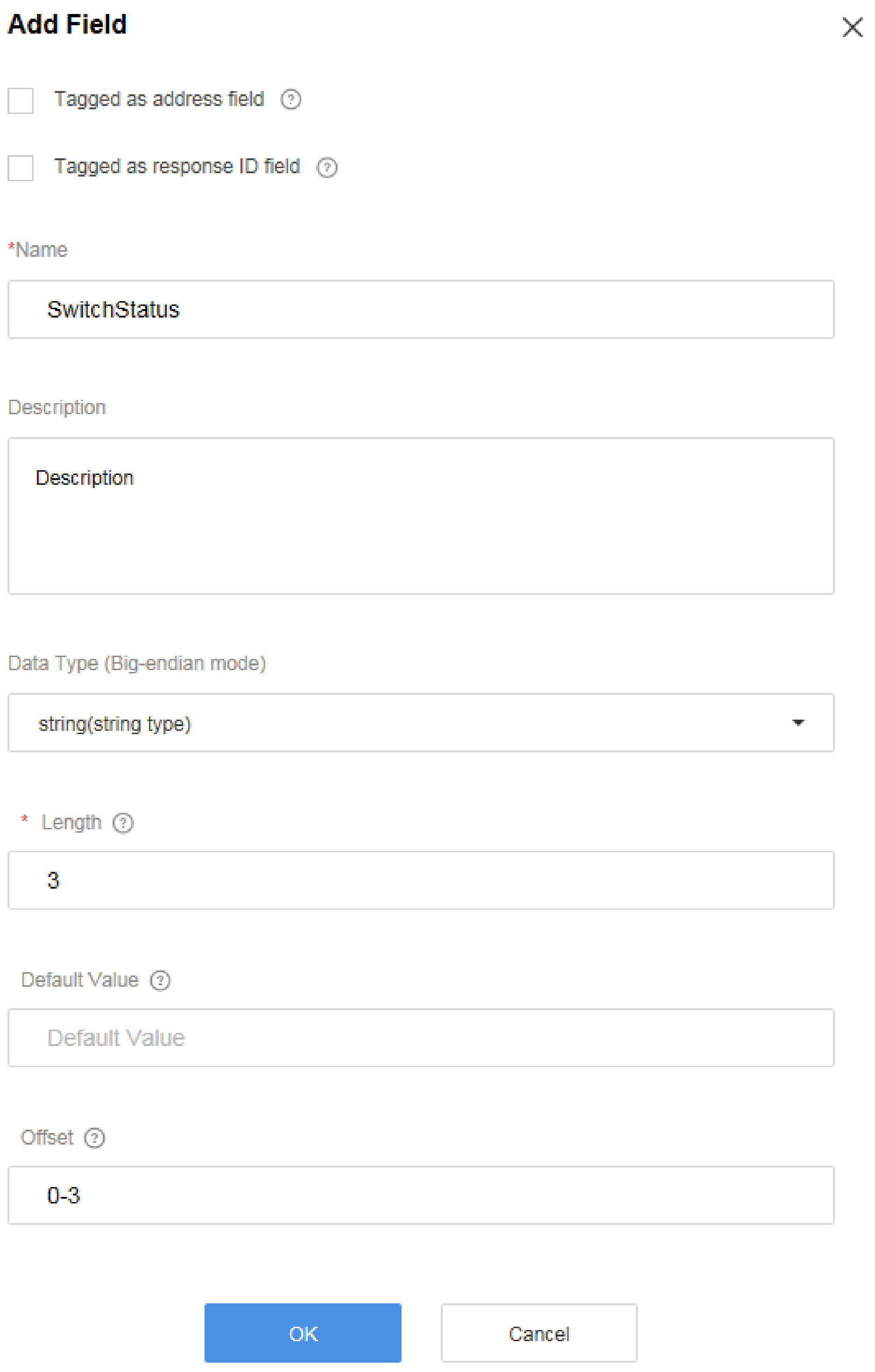
*Message Type: ○ Data Reporting ● Command Delivery

□ Add Response Field

Field

+ Add Field

Step 10 In the dialog box displayed, set **Name** to **SwitchStatus**, **Data Type** to **string(string type)**, and **Length** to **3**, and click **OK**.



Add Field

□ Tagged as address field

□ Tagged as response ID field

*Name: SwitchStatus

Description: Description

Data Type (Big-endian mode): string(string type)

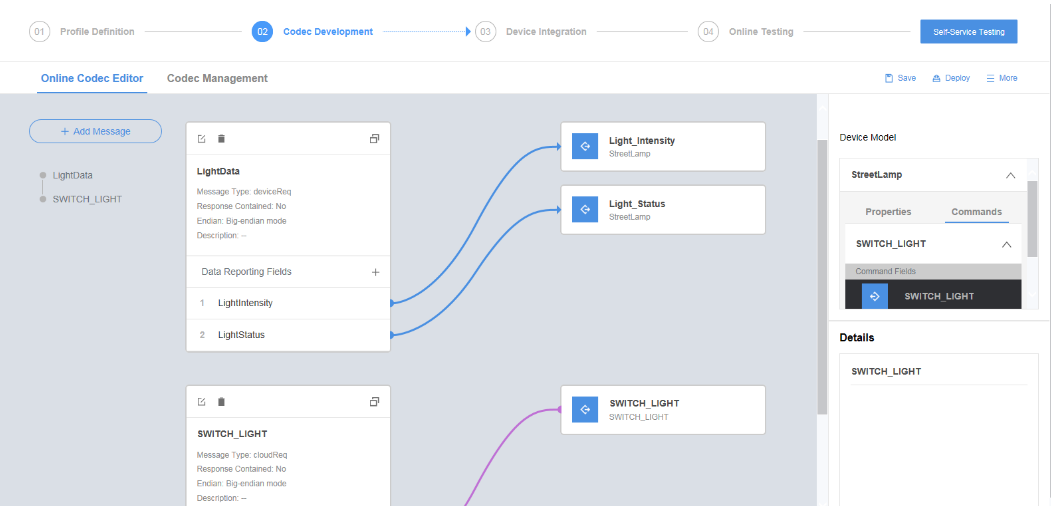
* Length: 3

Default Value: Default Value

Offset: 0-3

Step 11 Click **OK** to complete the codec for command delivery messages.

Step 12 Drag the property and command fields (defined in the profile file) in the **Device Model** area on the right to map the fields in the data report and command delivery messages defined by the codec.



LightData
Message Type: deviceReq
Response Contained: No
Endian: Big-endian mode
Description: --

1 LightIntensity
2 LightStatus

SWITCH_LIGHT
Message Type: cloudReq
Response Contained: No
Endian: Big-endian mode
Description: --

Step 13 Click **Save** in the upper right corner and click **Deploy** to complete codec deployment.

----End

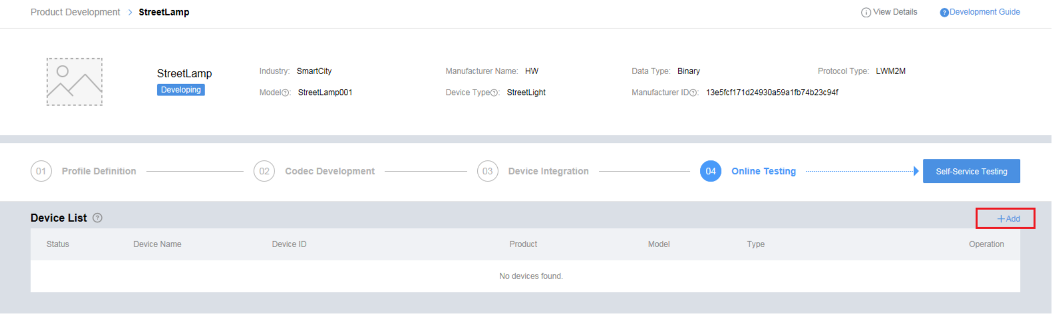
2.5 Using Simulators for Online Testing

Scenarios

Simulators commission the functions of devices and applications. You can commission the defined profile file and codec through the simulation of data reporting and command delivery.

Procedure

Step 1 On the **Product Development** page, enter the created product, click **Online Testing**, and click **Add**.

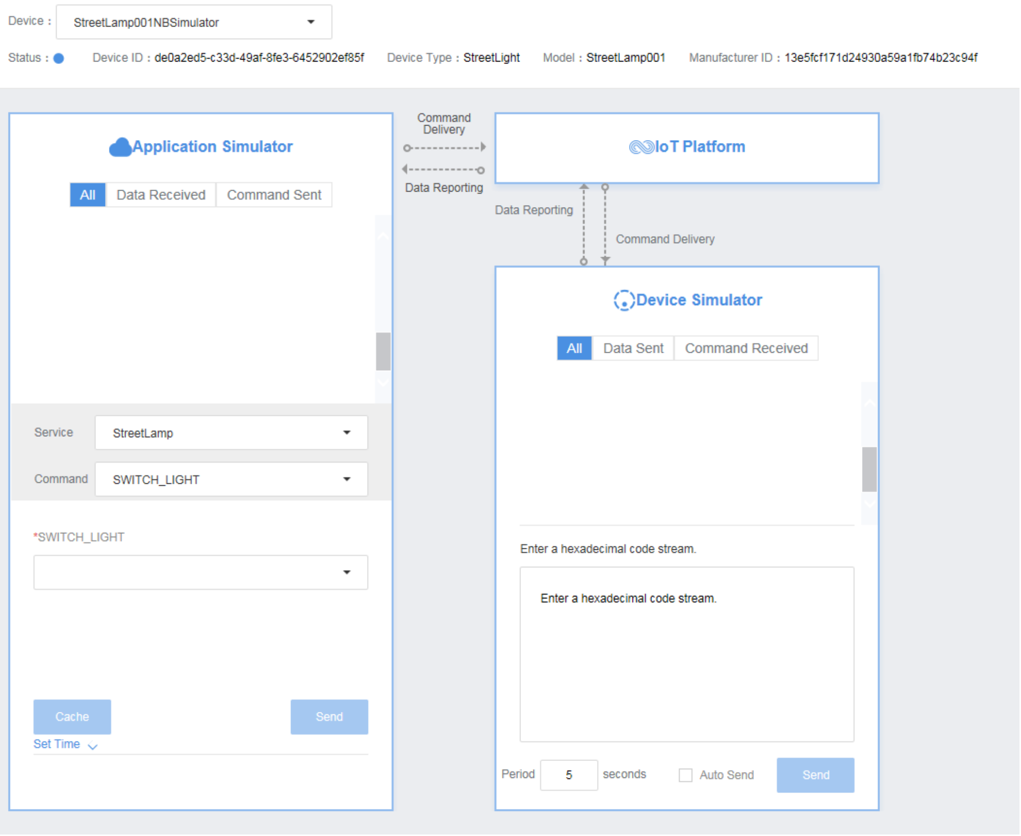


Industry: SmartCity
Model: StreetLamp001
Manufacturer Name: HW
Device Type: StreetLight
Data Type: Binary
Manufacturer ID: 13e5fcf171d24930a59a1fb74b23c94f
Protocol Type: LWM2M

Status	Device Name	Device ID	Product	Model	Type	Operation

No devices found.

Step 2 Select **No** for **Is Physical Device Available**, and click **OK** to use the simulator for testing. The Developer Center creates a simulator by default and displays the testing page.



Device : StreetLamp001NBSimulator

Status : ● Device ID : de0a2ed5-c33d-49af-8fe3-6452902ef85f Device Type : StreetLight Model : StreetLamp001 Manufacturer ID : 13e5fcf171d24930a59a1fb74b23c94f

Service: StreetLamp

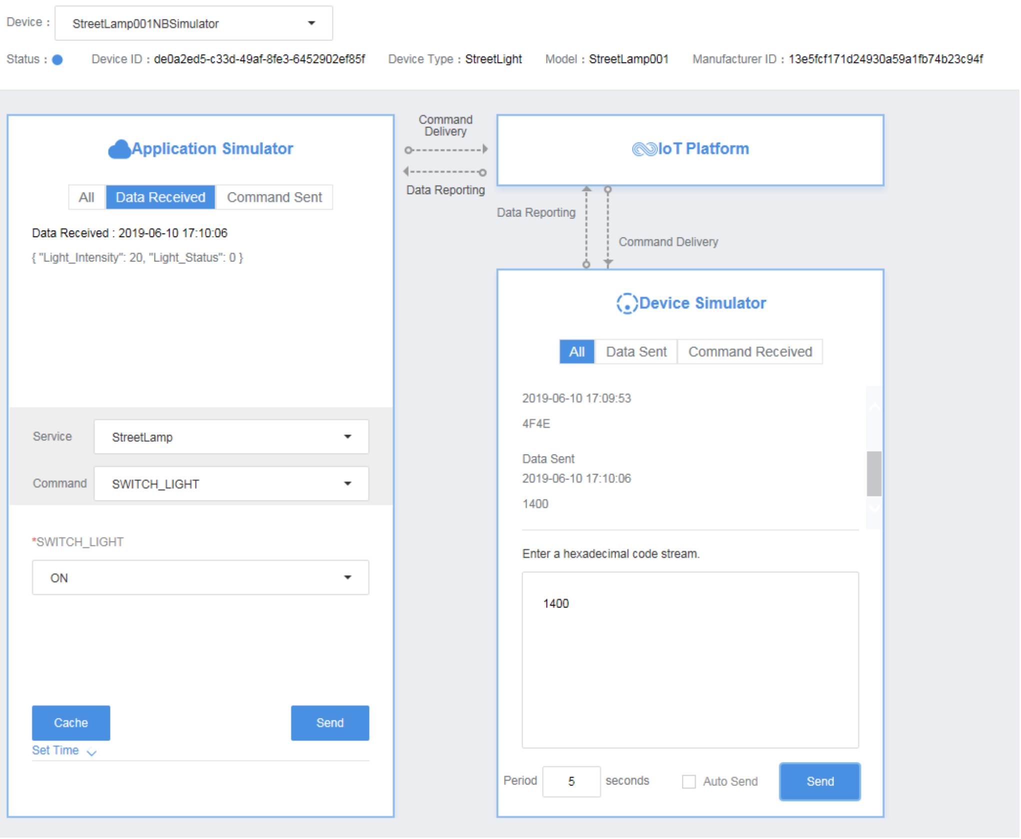
Command: SWITCH_LIGHT

*SWITCH_LIGHT

Enter a hexadecimal code stream.

Period 5 seconds □ Auto Send

Step 3 Simulate a data reporting scenario. Assume that the light intensity collected by the street lamp is 20 and the status of the street lamp switch is 0 (off). Under **Device Simulator**, enter the hexadecimal code stream **1400**, and click **Send**. (The first byte indicates light intensity, corresponding to hexadecimal code stream 14, and the second indicates switch status, corresponding to hexadecimal code stream 00.) Under **Application Simulator**, **"Light_Intensity": 20, "Light_Status": 0** should be displayed.



Device : StreetLamp001NBSimulator

Status : ● Device ID : de0a2ed5-c33d-49af-8fe3-6452902ef85f Device Type : StreetLight Model : StreetLamp001 Manufacturer ID : 13e5fcf171d24930a59a1fb74b23c94f

Data Received : 2019-06-10 17:10:06
{ "Light_Intensity": 20, "Light_Status": 0 }

Service: StreetLamp

Command: SWITCH_LIGHT

*SWITCH_LIGHT: ON

2019-06-10 17:09:53
4F4E

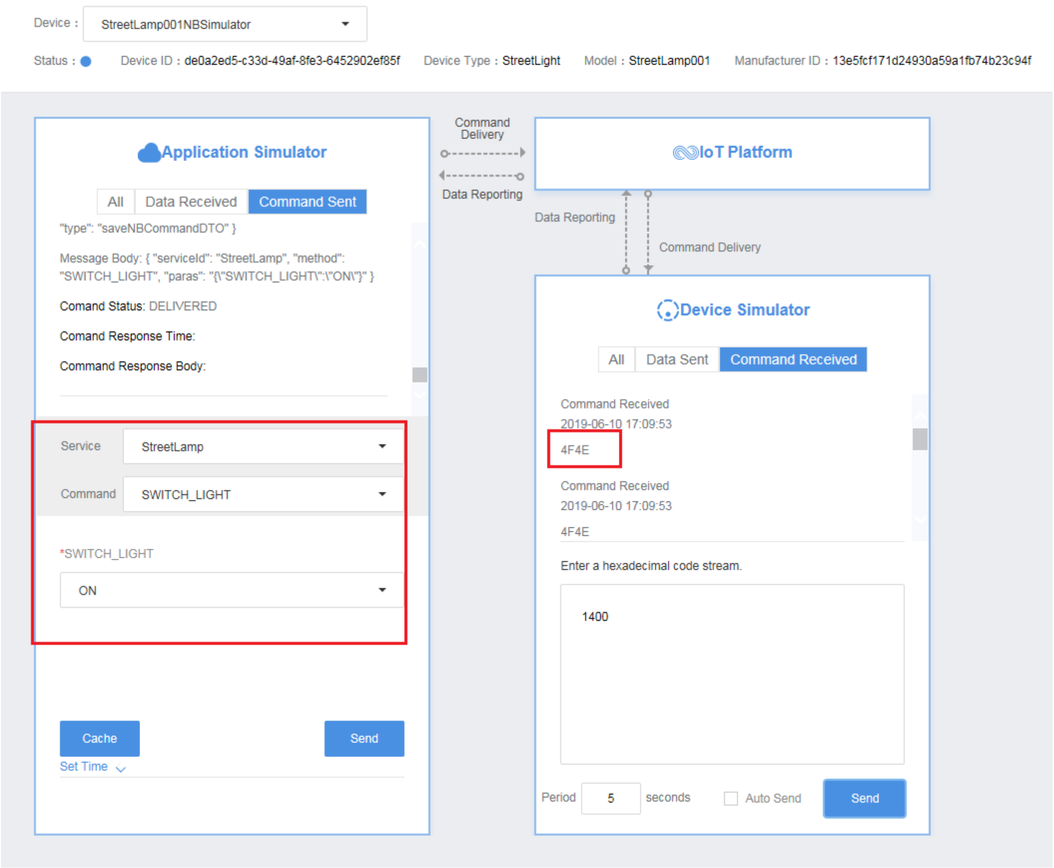
Data Sent
2019-06-10 17:10:06
1400

Enter a hexadecimal code stream.

1400

Period 5 seconds □ Auto Send

Step 4 Simulate a command delivery scenario. Under **Application Simulator**, select **StreetLamp** for **Service**, **SWITCH_LIGHT** for **Command**, **ON** for **SWITCH_LIGHT**, and click **Send**. Under **Application Simulator**, **4F4E** should be displayed.



Device : StreetLamp001NBSimulator

Status : ● Device ID : de0a2ed5-c33d-49af-8fe3-6452902ef85f Device Type : StreetLight Model : StreetLamp001 Manufacturer ID : 13e5fcf171d24930a59a1fb74b23c94f

"type": "saveNBCommandDTO" }

Message Body: { "serviceId": "StreetLamp", "method": "SWITCH_LIGHT", "paras": "{\"SWITCH_LIGHT\":\"ON\"}" }

Comand Status: DELIVERED

Comand Response Time:

Command Response Body:

Service: StreetLamp

Command: SWITCH_LIGHT

*SWITCH_LIGHT: ON

Command Received
2019-06-10 17:09:53
4F4E

Command Received
2019-06-10 17:09:53
4F4E

Enter a hexadecimal code stream.

1400

Period 5 seconds □ Auto Send

----End