API Reference

07 API Reference - International

Issue	01
Date	2024-11-06

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 API Reference on the Application Side

1.1 Before You Start

Overview

IoTDA provides abundant management capabilities through APIs, including product, device, device group, tag, device CA certificate, device shadow, device command, device message, device property, subscription, rule, and batch task management, which helps you quickly build applications based on the platform. You can use the platform services based on the APIs provided in this document. For details on all the APIs provided by the platform, see **API**.

Description

The platform supports RESTful APIs that allow HTTPS-based calling. For details on how to call an API, see **Calling APIs**.

Endpoints

An endpoint is the request address for calling an API. For endpoints of IoTDA, see **Platform Connection Information**.

Constraints

- Forward compatibility is supported. If an API is upgraded, the API of the earlier version can still be used, but its functions are not enhanced. The new functions are provided only in the new version.
- When receiving and processing response messages and push messages from the IoT platform, an application must support or ignore new parameters in the messages. It should not return an error because of the new parameters.
- For details on more restrictions on calling APIs, see **Limitations**.

Concepts

- Account

  An account is created upon successful registration with Huawei Cloud. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform routine management. For security purposes, create users and grant them permissions for routine management.

- Users

  An Identity and Access Management (IAM) user is created by an account to use cloud services. Each IAM user has its own identity credentials (password and access keys).

  An IAM user can view the account ID and user ID on the **My Credentials** page of the management console. The account name, username, and password will be required for API authentication.

- Region

  Regions are divided from the dimensions of geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

  For details, see **Region and AZ**.

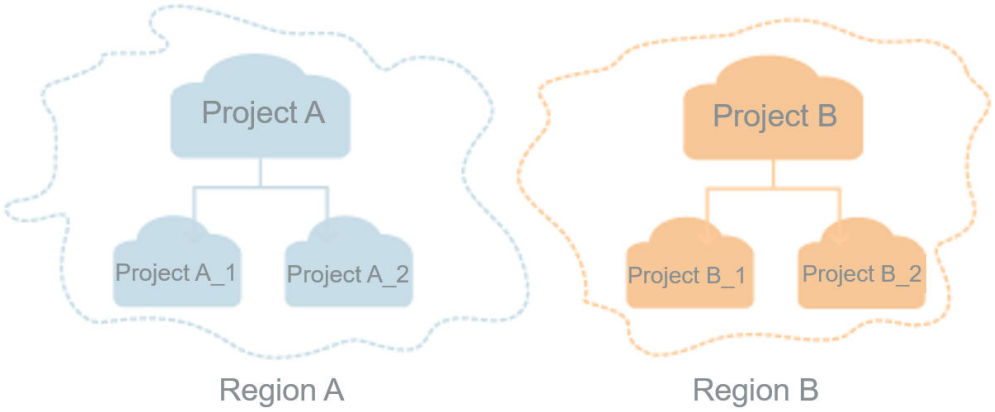
- Availability zone (AZ)

  An availability zone (AZ) contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proofing, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.

- Project

  Projects group and isolate resources (including compute, storage, network, and other resources) across physical regions. A default project is provided for each region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. If you need more refined access control, create subprojects under a default project and purchase resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated. It contains resources in multiple regions, and allows resources to be added or removed. For details about how to obtain enterprise project IDs and features, see **Applicable Scenarios**.
- Resource spaces

  Resource space is a space allocated for your applications. Resources (such as products and devices) created on the platform must belong to a resource space. You can use the resource space for domain-based management. For details, see **Resource Spaces**.
- Products

  A product model describes the capabilities and features of a device. Developers can define product models to build an abstract model of a device on IoT platform. For details, see **Product Model Definition**.
- Message delivery

  Message delivery does not rely on product models. The platform provides one-way notifications for devices and caches messages. For details, see **Message Delivery**.
- Command delivery

  A product model defines commands that can be delivered to the devices. Applications can call platform APIs to deliver commands to the devices to effectively manage these devices. For details, see **Command Delivery**.
- Property delivery

  Property delivery is used for property query or modification. An application or the platform can obtain device property information or modify the properties, and synchronize the modification result to the device. For details, see **Property Delivery**.
- AMQP queue management

  AMQP is short for Advanced Queuing Message Protocol. You can use the AMQP client to establish a connection with IoTDA to receive data. For details, see **AMQP Subscription/Push**.
- Data forwarding

  A device can connect to and communicate with the platform. The device reports data to the platform using custom topics or product models. After the

subscription/push configuration on the console is complete, the platform pushes messages about device lifecycle changes, reported device properties, reported device messages, device message status changes, device status changes, and batch task status changes to the application. For details, see **Overview of Subscription and Push**.

- Device shadow

  The IoT platform supports the creation of device shadows. A device shadow is a JSON file that stores the device status, latest device properties reported, and device configurations to deliver. Each device has only one shadow. A device can retrieve and set its shadow to synchronize properties, either from the shadow to the device or from the device to the shadow. For details, see **Device Shadow**.

- Device group

  A group is a collection of devices. You can create groups for all the devices in a resource space based on different rules, such as regions and types, and you can operate the devices by group. For example, you can perform a firmware upgrade on a group of water meters in the resource space. Devices in a group can be added, deleted, modified, and queried. A device can be bound to and unbound from multiple groups. For details, see **Group**.

- Tags

  You can add tags to cloud resources for quicker search. You can view, modify, and delete these tags in a unified manner, facilitating cloud resource management. For details, see **Tag Overview**.

- Linkage rules

  Linkage rules are classified into device-side rules and cloud rules. Device-side rules: Device-side rules are device linkage rules delivered to devices, where the device-side rule engine parses and executes the rules. Device-side rules can still run on devices when the network is interrupted or devices cannot communicate with the platform. Cloud rules: If you set a cloud rule, IoTDA determines whether the rule triggering condition is met. If the condition is met, IoTDA performs actions you set, such as alarm reporting, topic notification, and command delivery. For details, see **Cloud Rules**.

- Batch tasks

  You can use batch tasks to perform a batch operation on multiple devices. Supported batch operations: Upgrading software and firmware, creating, modifying, deleting, freezing, unfreezing, and updating devices, creating commands and messages, and setting device shadow.

- OTA upgrade

  OTA upgrade refers to software and firmware upgrade. Software upgrade refers to the upgrade of the system software and application software of the device. Firmware upgrade refers to the upgrade of the underlying driver 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. For details, see **About OTA Upgrade**.

1.2 Calling APIs

1.2.1 Constructing a Request

This topic describes the structure of a RESTful API request. It also uses the APIs for **querying a product** and **creating a product** as an example to describe how to call an API.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme**: Protocol used to transmit requests. All APIs use HTTPS.
- **Endpoint**: Domain name or IP address of the server bearing the REST service endpoint. Obtain the value from **Platform Connection Information**.
- **resource-path**: Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, **resource-path** of the API for querying products is **/v5/iot/{project_id}/products/{product_id}**.
- **query-string**: Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

For example, to obtain information about a specific product from the IoT platform in **CN North-Beijing4**, combine the endpoint (**iotda.cn-north-4.myhuaweicloud.com**) of the **CN North-Beijing4** region and **resource-path** (**/v5/iot/{project_id}/products/{product_id}**) in the URI of the API for **querying a product**. The following is an example:

```
https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/{project_id}/products/{product_id}
```

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Method

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET**: requests the server to return specified resources.
- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as **GET** except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to create an API group (see **1.4.1.3 Query a Product**), the request method is GET. The request is as follows:

```
GET https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/{project_id}/products/{product_id}
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json;charset=utf-8**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies the user token. This parameter is mandatory when token authentication is used. You can call the API used to **obtain a user token** to obtain the value of this parameter. In the response message header returned by the API, **X-Subject-Token** is the desired user token.

The API for **querying a product** requires authentication. Therefore, the **Content-Type** and **X-Auth-Token** fields need to be added to the header of the request for calling the API. An example of such a request is as follows:

```
GET https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/{project_id}/products/{product_id}
Content-Type: application/json
X-Auth-Token: ******
```

Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header. If the request body supports Chinese characters, the Chinese characters must be encoded in UTF-8 mode.

The request body varies according to APIs. Certain APIs do not require the request body, such as GET and DELETE.

For the API for **creating a product**, the request parameters and parameter description are available in the request. The following is an example request with the body included. Replace the italic fields in bold with the actual values. For example, ***name*** indicates the product name, ***device_type*** indicates the device type, and ***protocol_type*** indicates the protocol type used by the device.

```
POST https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/abab***cdcd/products
Content-Type: application/json
X-Auth-Token: ********

{
  "name" : "Thermometer",
  "device_type" : "Thermometer",
  "protocol_type" : "MQTT",
  "data_format" : "binary",
  "manufacturer_name" : "ABC",
  "industry" : "smartCity",
  "description" : "this is a thermometer produced by Huawei",
  "service_capabilities" : [ {
    "service_type" : "temperature",
    "service_id" : "temperature",
```



```
  "description" : "temperature",
  "properties" : [ {
    "unit" : "centigrade",
    "min" : "1",
    "method" : "R",
    "max" : "100",
    "data_type" : "decimal",
    "description" : "force",
    "step" : 0.1,
    "enum_list" : [ "string" ],
    "required" : true,
    "property_name" : "temperature",
    "max_length" : 100
  } ],
  "commands" : [ {
    "command_name" : "reboot",
    "responses" : [ {
      "response_name" : "ACK",
      "paras" : [ {
        "unit" : "km/h",
        "min" : "1",
        "max" : "100",
        "para_name" : "force",
        "data_type" : "string",
        "description" : "force",
        "step" : 0.1,
        "enum_list" : [ "string" ],
        "required" : false,
        "max_length" : 100
      } ]
    } ],
    "paras" : [ {
      "unit" : "km/h",
      "min" : "1",
      "max" : "100",
      "para_name" : "force",
      "data_type" : "string",
      "description" : "force",
      "step" : 0.1,
      "enum_list" : [ "string" ],
      "required" : false,
      "max_length" : 100
    } ]
  } ],
  "option" : "Mandatory"
 } ],
 "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
}
```


If all data required for the API request is available, you can send the request to call the API through **curl**, **Postman**, or coding.

1.2.2 Authentication

Requests for calling an API can be authenticated in either of the following methods:

- Authentication using tokens: General requests are authenticated using tokens.
- Access Key ID/Secret Access Key (AK/SK)-based authentication: Requests are authenticated by encrypting the request body using an AK/SK.

Token Authentication

NOTE

A token is a character string generated by the server and is used as a token for a client to send a request. After the first login, the server generates a token and returns the token to the client. The client only needs to carry the token to request data, and does not need to carry the username and password again. The validity period of a token is 24 hours, which starts from the time when the client obtains the token. If the same token needs to be used for authentication, it is recommended that the token be cached to avoid frequent calling. Before the token expires, you must update the token or obtain a new token. Otherwise, the authentication on the server will fail after the token expires.

Obtaining a new token does not affect the validity of the existing token.

A token is used to acquire temporary permissions. Token-based authentication adds a token to the request header during API calling to obtain permissions to operate APIs.

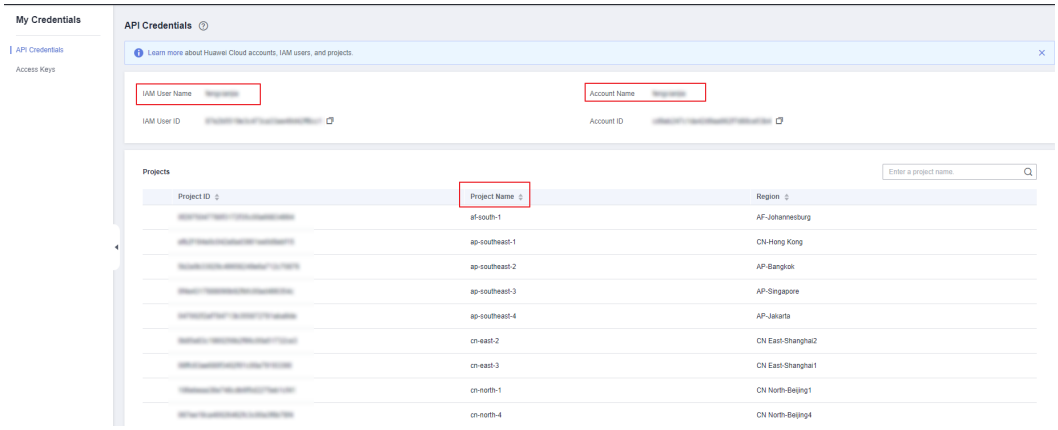
Call the API **Obtaining a User Token Through Password Authentication** to obtain the token. The following is an example:

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "projectname"
            }
        }
    }
}
```

Note: **username** indicates the IAM username, **password** indicates the password for logging in to Huawei Cloud, **domainname** indicates the account name, and **projectname** indicates the project name. You can obtain them from the **My Credentials** page.

Figure 1-2 API credential - obtaining credential information



In the response to the API used to obtain a user token, **X-Subject-Token** is the desired user token.

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
GET https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/{project_id}/products/{product_id}
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the request headers for authentication.

- AK: access key ID. It is a unique ID associated with an SK. AK is used together with SK to sign requests.
- SK: secret access key. It is used together with an AK to sign requests. They can identify request senders and prevent requests from being modified.

In AK/SK-based authentication, you can sign requests using an AK/SK based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see **AK/SK Signing and Authentication Guide**.

NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

1.2.3 Returning a Response

Status Code

After sending a request, you will receive a response that includes a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **1.7.1 Status Code**.

If **201** is returned for calling the API used to **create a product**, the request is successful.

Response Header Field

Similar to a request, a response also has a header, for example, **Content-Type**.

For the API used to **create a product**, the platform will return a response containing message headers, for example, **Content-Type** and **Date**.

Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

For the API used to **create a product**, the following message body is returned: The following is part of the response body for the API used to create a product:

```
{
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "name" : "Thermometer",
  "device_type" : "Thermometer",
  "protocol_type" : "LWM2M",
  "data_format" : "binary",
  "manufacturer_name" : "ABC",
  "industry" : "smartCity",
  "description" : "this is a thermometer produced by Huawei",
    ......
}
```

If an error occurs during API calling, an error code and error description will be displayed. The following shows an error response body:

```
{
    "error_msg": "The format of message is error",
    "error_code": "IOTDA.013005"
}
```

In the preceding information, **error_code** is an error code, and **error_msg** describes the error.

1.3 API Overview

Before calling all the listed APIs, complete authentication by following the instructions provided in **1.2.2 Authentication**.

Product Management APIs

API	Description
1.4.1.2 Query the Product List	This API is used to query the list of product models that have been imported to the platform to learn about the brief information about the product models.

API	Description
1.4.1.1 Create a Product	This API is used to create a product. It is used to create a product but not to create or install a plug-in. If you need to encode or decode data, develop and install a plug-in on the platform.
1.4.1.3 Query a Product	This API is used to query details about a specific product model that has been imported to the platform, including the services, properties, and commands of the product model.
1.4.1.4 Modify a Product	This API is used to modify a specific product model that has been imported to the platform, including the services, properties, and commands of the product model. This API is used to modify the product but not to modify or install a plug-in. If the service definition in the product is modified and the corresponding plug-in exists on the platform, modify and reinstall the plug-in.
1.4.1.5 Delete a Product	This API is used to delete a specific product model that has been imported to the platform.

Device Management APIs

API	Description
1.4.2.2 Query the Device List	This API is used to query the device list on the platform.
1.4.2.1 Create a Device	This API is used to register a device with the platform. The device can access the platform only after being registered.
1.4.2.3 Query a Device	This API is used to query the details about a specific device on the platform.
1.4.2.4 Modify a Device	This API is used to modify the basic information about a specific device on the platform.
1.4.2.5 Delete a Device	This API is used to delete a specific device from the platform. If the device is connected to an indirectly connected device, you must delete the indirectly connected device first.
1.4.2.6 Reset a Device Secret	This API is used by an application to reset a device secret. If the request contains a secret, the platform resets the device secret to the specified secret. If the request does not contain a secret, the platform automatically generates a new random secret and returns it.

API	Description
1.4.2.7 Freeze a Device	This API is used to freeze a device. After the device is frozen, the device cannot go online. You can unfreeze the device by calling the API used to unfreeze a device. Currently, only devices that are directly connected to the platform can be frozen.
1.4.2.8 Unfreeze a Device	This API is used to unfreeze a device. After the device is unfrozen, the device can go online.
1.4.2.9 Reset a Device Fingerprint	This API is used by an application to reset a device fingerprint. A device fingerprint will be reset to the specified one. If a device fingerprint is not specified, it is left empty.
1.4.2.10 Query Device List Flexibly	This API is used by an application to query a desired device list using SQL statements.
1.4.2.11 Query the List of Device Groups to Which a Specified Device Is Added	This API is used by an application to query the list of device groups to which a device is added.

Device Message APIs

API	Description
1.4.3.1 Deliver a Message to a Device	This API is used to query messages of a specific device. By default, the platform stores a maximum of 20 messages for each device. If the number of messages exceeds 20, the earliest messages will be overwritten by subsequent messages.
1.4.3.2 Query Device Messages	This API is used by an application to deliver a message to a specific device to control the device. After an application delivers a message to the platform, the platform returns a response to the application and then sends the message to the device.
1.4.3.3 Query a Message by Message ID	This API is used to query a message by message ID.

Device Command APIs

API	Description
1.4.4.1.1 Deliver a Command to a Device	A product model defines commands that the platform can deliver to devices. This API is used by an application to deliver synchronous commands to a specific device to control the device.
1.4.4.2.1 Deliver an Asynchronous Command	A product model defines commands that the platform can deliver to devices. This API is used by an application to deliver asynchronous commands to a specific device to control the device.
1.4.4.2.2 Query a Command with a Specific ID	This API is used to query a command by command ID.

Device Property APIs

API	Description
1.4.5.2 Query Device Properties	A product model defines properties that the platform can deliver to devices. This API is used by an application to query properties of a specific device.
1.4.5.1 Modify Device Properties	A product model defines properties that the platform can deliver to devices. This API is used by an application to modify properties of a device. The platform sends the properties to the device in synchronous mode and returns the execution result synchronously to the application.

AMQP Queue Management APIs

API	Description
1.4.6.2 Query the AMQP List	This API is used to query the AMQP queue list on the platform.
1.4.6.1 Create an AMQP Queue	This API is used to create an AMQP queue on the platform. You can call the data transfer rule management API to push data to the AMQP queue.
1.4.6.3 Query an AMQP Queue	This API is used to query the details of a specific AMQP queue on the platform.
1.4.6.4 Delete an AMQP Queue	This API is used to delete a specific AMQP queue from the platform.

Access Credential Management APIs

API	Description
1.4.7.1 Generate an Access Credential	An access code is an authentication credential used by a client to connect to the platform through a protocol such as AMQP. A pair of new access codes will be generated when this API is called.

Data Transfer Rule Management APIs

API	Description
1.4.8.2 Query the Rule Triggering Condition List	This API is used to query the rule condition list on the platform.
1.4.8.1 Create a Rule Triggering Condition	This API is used to create a rule condition on the platform.
1.4.8.3 Query a Rule Triggering Condition	This API is used to query the configuration of a specific rule condition on the platform.
1.4.8.4 Modify a Rule Triggering Condition	This API is used to modify the configuration of a specific rule condition on the platform.
1.4.8.5 Delete a Rule Triggering Condition	This API is used to delete a specific rule condition from the platform.
1.4.8.7 Query the Rule Action List	This API is used to query the rule action list on the platform.
1.4.8.6 Create a Rule Action	This API is used to create a rule action on the platform.
1.4.8.8 Query a Rule Action	This API is used to query the configuration of a specific rule action on the platform.
1.4.8.9 Modify a Rule Action	This API is used to modify a specific rule action on the platform.
1.4.8.10 Delete a Rule Action	This API is used to delete a specific rule action from the platform.

Data Transfer APIs

API	Description
1.4.9.1 Push a Device Status Change Notification	An application has created a rule about device status change notification on the platform. When the status of a device changes, the platform calls this API to push a notification to the application.

API	Description
1.4.9.2 Push a Device Property Reporting Notification	An application has created a rule about device property reporting notification on the platform. When a device reports its property data, the platform calls this API to push a notification to the application.
1.4.9.3 Push a Device Message Status Change Notification	An application has created a rule about message status change notification on the platform. When the status of a device message changes, the platform calls this API to push a notification to the application.
1.4.9.4 Push a Batch Task Status Change Notification	An application has created a rule about batch task status change notification on the platform. When the status of a batch task changes, the platform calls this API to push a notification to the application.
1.4.9.5 Push a Device Message Reporting Notification	An application has created a rule about device message reporting notification on the platform. When a device reports a message, the platform calls this API to push a notification to the application.
1.4.9.6 Push a Device Addition Notification	An application has created a rule about device addition notification on the platform. When a device is added on the platform, the platform calls this API to push a notification to the application.
1.4.9.7 Push a Device Update Notification	An application has created a rule about device update notification on the platform. When a device is updated on the platform, the platform calls this API to push a notification to the application.
1.4.9.8 Push a Device Deletion Notification	An application has created a rule about device deletion notification on the platform. When a device is deleted from the platform, the platform calls this API to push a notification to the application.
1.4.9.9 Push a Product Addition Notification	An application has created a rule about product addition notification on the platform. When a product is added on the platform, the platform calls this API to push a notification to the application.
1.4.9.10 Push a Product Update Notification	An application has created a rule about product update notification on the platform. When a product is updated on the platform, the platform calls this API to push a notification to the application.
1.4.9.11 Push a Product Deletion Notification	An application has created a rule about product deletion notification on the platform. When a product is deleted from the platform, the platform calls this API to push a notification to the application.

API	Description
1.4.9.12 Push an Asynchronous Device Command Status Change Notification	An application has created a rule about command status change notifications on the platform. When the status of a command changes, the platform calls this API to push a notification to the application.

Device Linkage Rule APIs

API	Description
1.4.10.2 Query the Rule List	This API is used to query the list of device linkage rules on the platform.
1.4.10.1 Creating a Rule	This API is used to create a device linkage rule on the platform.
1.4.10.4 Query a Rule	This API is used to query the configuration of a specific rule on the platform.
1.4.10.3 Modify a Rule	This API is used to modify the configuration of a specific rule on the platform.
1.4.10.5 Delete a Rule	This API is used to delete a specific rule from the platform.
1.4.10.6 Modify the Rule Status	This API is used to modify the status of a specific rule on the platform by activating or deactivating the rule.

Device Shadow APIs

API	Description
1.4.11.1 Query a Device Shadow	This API is used to query the device shadow of a specific device, including the desired device configurations (in the desired section) and the latest configurations reported by the device (in the reported section). Only LwM2M-based devices support the device shadow function, and only properties defined by LwM2M can be modified. User-defined properties cannot be modified.

API	Description
1.4.11.2 Configure Desired Properties in a Device Shadow	This API is used by an application to configure the desired data (in the desired section) of the device shadow. When the device goes online, the application delivers the data to the device. The device shadow properties are coupled with the product model. The desired properties must be defined in the product model and can be delivered only when the method has the Write property. Only LwM2M-based devices support the device shadow function, and only properties defined by LwM2M can be modified. User-defined properties cannot be modified.

Device Group Management APIs

API	Description
1.4.12.2 Query the Device Group List	This API is used to query the device group list on the platform.
1.4.12.1 Create a Device Group	This API is used to create a device group. A Huawei Cloud account can have a maximum of 1,000 groups, including parent and child groups.
1.4.12.3 Query a Device Group	This API is used to query details about a device group.
1.4.12.4 Modify a Device Group	This API is used to modify a specific device group.
1.4.12.5 Delete a Device Group	This API is used to delete a specific device group.
1.4.12.6 Manage Devices in a Device Group	This API is used to manage devices in a device group, including adding and deleting devices to and from a device group. A maximum of 20,000 devices can be added to a device group. A device can be bound to a maximum of 10 device groups.
1.4.12.7 Query Devices in a Device Group	This API is used to query the list of devices in a specific device group.

Tag Management APIs

API	Description
1.4.13.1 Bind Tags	This API is used to bind a tag to a specific resource. Currently, only devices can be bound with tags.

API	Description
1.4.13.2 Unbind Tags	This API is used to unbind a tag from a specific resource. Currently, only devices can be unbound from tags.
1.4.13.3 Query Resources by Tag	This API is used to query resources bound with a specific tag.

Resource Space Management APIs

API	Description
1.4.15.1 Query the Resource Space List	A resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that in the application. The only difference relies on the name. This API is used by an application to query the resource space list.
1.4.15.2 Create a Resource Space	A resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that in the application. The only difference relies on the name. This API is used by an application to create a resource space.
1.4.15.3 Query a Resource Space	A resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that in the application. The only difference relies on the name. This API is used by an application to query details about a specific resource space.
1.4.15.4 Delete a Resource Space	This API is used to delete a specific resource space. Deleting a resource space is a high-risk operation. After the resource space is deleted, resources such as products and devices in the space will be unavailable. Exercise caution when performing this operation.

Batch Task APIs

API	Description
1.4.16.2 Query the Batch Task List	This API is used to query the batch task list on the platform. Each task includes the task content, status, and completion statistics.

API	Description
1.4.16.1 Create a Batch Task	This API is used to create a batch task to perform batch operations on multiple devices. Supported batch operations: Upgrading software and firmware, creating, modifying, deleting, freezing, unfreezing, and updating devices, creating commands and messages, and setting device shadow.
1.4.16.3 Query a Batch Task	This API is used to query information about a specific batch task on the platform, including the task content, status, completion statistics, and subtask list.
1.4.16.4 Delete a Batch Task	This API is used by an application to delete a completed batch task (successful, failed, partially successful, or stopped) from the platform.
1.4.16.5 Re-Execute a Batch Task	This API is used by an application to retry a batch task. Currently, **task_type** can only be set to **firmwareUpgrade** or **softwareUpgrade**. If the task specified by **task_id** is successful, stopped, being stopped, waiting, or being initialized, this API cannot be called.
1.4.16.6 Stop a Batch Task	This API is used by an application to stop a batch task. Currently, **task_type** can only be set to **firmwareUpgrade** or **softwareUpgrade**. If the task specified by **task_id** has been completed (successful, failed, partially successful, or stopped) or is being stopped, this API cannot be called.
1.4.16.7.2 Query the List of Batch Task Files	This API is used to query the batch task file list.
1.4.16.7.1 Upload a Batch Task File	This API is used to upload a batch task file to create a batch task. You can upload files of batch device creation, deletion, freezing, unfreezing tasks, and updating tasks.
1.4.16.7.3 Delete a Batch Task File	This API is used to delete a batch task file.

Device CA Certificate Management APIs

API	Description
1.4.17.2 Obtain the Device CA Certificate List	This API is used to obtain the device CA certificate list.
1.4.17.1 Upload a Device CA Certificate	This API is used to upload a device CA certificate.

API	Description
1.4.17.3 Delete a Device CA Certificate	This API is used to delete a device CA certificate.
1.4.17.5 Verify a Device CA Certificate	This API is used to verify the CA certificate of a device. CA certificate verification checks whether the user has the private key of the CA certificate of the device.
1.4.17.4 Update a CA Certificate	This API is used to update a device CA certificate.

OTA Upgrade Package Management

API	Description
1.4.18.1 Create an OTA Upgrade Package	This API is used to create an OBS object associated with an upgrade package.
1.4.18.2 Query the OTA Upgrade Package List	This API is used to query the upgrade package associated with an OBS object.
1.4.18.3 Obtain OTA Upgrade Package Details	This API is used to query details about the upgrade package associated with an OBS object.
1.4.18.4 Delete an OTA Upgrade Package	This API is used to delete the upgrade package information associated with an OBS object. The OBS object will not be deleted.

Message Broadcasting

API	Description
1.4.19.1 Broadcasting a Message	This API is used by an application to broadcast a message to all online devices that subscribe to a specified topic.

Device Tunnel Management

API	Description
1.4.20.1 Create a Device Tunnel	This API is used by an application to create a tunnel.
1.4.20.2 Query All Tunnels of a Device	This API is used by an application to query all device tunnels.

API	Description
1.4.20.3 Query Device Tunnels	This API is used by an application to query device tunnel details.
1.4.20.4 Disable a Device Tunnel	This API is used by an application to close a tunnel.
1.4.20.5 Delete a Device Tunnel	This API is used by an application to delete a tunnel.

Device Proxy

API	Description
1.4.23.1 Create a Device Proxy	This API is used by an application to call a device proxy.
1.4.23.2 Query Device Proxy List	This API is used by an application to query all device proxies.
1.4.23.3 Query Device Proxy Details	This API is used by an application to query device proxy details.
1.4.23.4 Modify a Device Proxy	This API is used by an application to modify a device proxy.
1.4.23.5 Delete a Device Proxy	This API is used by an application to delete a device proxy.

Bridge Management

API	Description
1.4.24.1 Create a Bridge	This API is used by an application to create a bridge.
1.4.24.2 Query the Bridge List	This API is used by an application to query all bridges.
1.4.24.3 Delete a Bridge	This API is used by an application to delete a bridge.
1.4.24.4 Reset the Bridge Secret	This API is used by an application to reset the bridge password.

Device policy management

API	Description
1.4.25.1 Create a Device Policy	This API is used by an application to create a device policy.
1.4.25.2 Query the Device Policy List	This API is used by an application to query all device policies.
1.4.25.3 Delete a Device Policy	This API is used by an application to delete a device policy.
1.4.25.4 Query Device Policy Details	This API is used by an application to query device policy details.
1.4.25.5 Update Device Policy Information	This API is used by an application to update device policy information.
1.4.25.6 Bind a Device Policy	This API is used by an application to bind a device policy.
1.4.25.7 Unbind a Device Policy	This API is used by an application to unbind a device policy.
1.4.25.8 Query the List of the Targets Bound to a Device Policy	This API is used by an application to query the list of targets bound to a device policy.

Pre-provisioning template management

API	Description
1.4.26.1 Create a Pre-provisioning Template	This API is used by an application to create a pre-provisioning template.
1.4.26.2 Query the Pre-provisioning Template List	This API is used by an application to query all created pre-provisioning templates.
1.4.26.3 Delete a Pre-provisioning Template	This API is used by an application to delete a pre-provisioning template.
1.4.26.4 Query Pre-provisioning Template Details	This API is used by an application to query details a pre-provisioning template.
1.4.26.5 Update Information About a Pre-provisioning Template with a Specified ID	This API is used by an application to update information about a pre-provisioning template with a specified ID.

Pre-provisioning template management

API	Description
1.4.27.1 Create a Custom Authenticator	This API is used by an application to create a custom authenticator.
1.4.27.2 Query the Custom Authenticator List	This API is used by an application to query all custom authorizers.
1.4.27.3 Delete a Custom Authenticator	This API is used by an application to delete a custom authenticator.
1.4.27.4 Query Custom Authenticator Details	This API is used by an application to query details about a custom authenticator.
1.4.27.5 Update a Custom Authenticator with a Specified ID	This API is used by an application to update a custom authorizer with a specified ID.

1.4 API

1.4.1 Product Management

A product model defines the capabilities or features of all devices under a product. Product management APIs are used to manage product models on the platform.

1.4.1.1 Create a Product

Function

This API is used by an application to create a product. This API is used to create a product but not to create or install a codec. If you need to encode or decode data, develop and install a codec on the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/products

Table 1-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-3 Request body parameters

Parameter	Mandatory	Type	Description
product_id	No	String	** Parameter description**: Product ID, which is unique in the resource space. Uniquely identifies a product in the resource space. If this parameter is specified, the platform uses the specified product ID. If this parameter is not specified, the platform allocates a product ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
name	Yes	String	**Parameter description**: product name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
device_type	Yes	String	**Parameter description**: device type. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
protocol_type	Yes	String	**Parameter description**: protocol used by the device. **Options**: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, **ONVIF**, **OPC-UA**, **OPC-DA**, and **Other**.
data_format	Yes	String	**Parameter description**: format of the data reported by the device. **Options**: • **json**: JSON format • **binary**: binary code stream format The default value is **json**. Default: **json**

Parameter	Mandatory	Type	Description
service_capabilities	Yes	Array of **ServiceCapability** objects	**Parameter description**: list of service capabilities of the device. **Value**: The array length cannot exceed 500, and the content size cannot exceed 500 KB.
manufacturer_name	No	String	**Parameter description**: manufacturer name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
industry	No	String	**Parameter description**: industry to which the device belongs. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
description	No	String	**Parameter description**: product description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the product to create will belong. If this parameter is not specified, the product to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 1-4 ServiceCapability

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: service ID of the device. The value must be unique in a product. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
service_type	Yes	String	**Parameter description**: service type of the device. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
properties	No	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service. **Value**: The array length cannot exceed 500.
commands	No	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service. **Value**: The array length cannot exceed 500.
events	No	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service. Currently, event customization is not supported. You do not need to define this field when creating or modifying a product. **Value**: The array length cannot exceed 500.
description	No	String	**Parameter description**: device service description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Parameter	Mandatory	Type	Description
option	No	String	**Parameter description**: whether the device service is mandatory. Currently, this field is not a functional field and is used only for identification. **Options**: • **Master**: master service • **Mandatory**: mandatory service • **Optional**: optional service The default value is **Optional**. Default: **Optional**

Table 1-5 ServiceProperty

Parameter	Mandatory	Type	Description
property_name	Yes	String	**Parameter description**: device property name. The value must be unique in the device service. If device property names are used as keys in the device shadow JSON file, the names cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). A device shadow file that contains these special characters cannot be updated. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	Yes	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	No	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**

Parameter	Mandatory	Type	Description
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	No	String	**Parameter description**: minimum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the device property.
step	No	Double	**Parameter description**: step of the device property.
unit	No	String	**Parameter description**: unit of the device property. **Value length**: a maximum of 16 characters Maximum: **16**
method	Yes	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: device property description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
default_value	No	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-6 ServiceCommand

Parameter	Mandatory	Type	Description
command_name	Yes	String	**Parameter description**: device command name. The value must be unique in the device service. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.
responses	No	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-7 ServiceCommandResponse

Parameter	Mandatory	Type	Description
response_name	Yes	String	**Parameter description**: name of the device command response. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.

Table 1-8 ServiceEvent

Parameter	Mandatory	Type	Description
event_type	Yes	String	**Parameter description**: device event type. The value must be unique in the device service. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-9 ServiceCommandPara

Parameter	Mandatory	Type	Description
para_name	Yes	String	**Parameter description**: parameter name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.

Parameter	Mandatory	Type	Description
data_type	Yes	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	No	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the parameter.
min	No	String	**Parameter description**: minimum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the parameter.
step	No	Double	**Parameter description**: step of the parameter.
unit	No	String	**Parameter description**: unit of the parameter. **Value length**: a maximum of 16 characters Maximum: **16**
description	No	String	**Parameter description**: parameter description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Response Parameters

Status code: 201

Table 1-10 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID.
app_name	String	Resource space name.
product_id	String	Product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	String	Product name.
device_type	String	Device type.
protocol_type	String	Protocol used by a device. Options: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, **ONVIF**, **OPC-UA**, **OPC-DA**, and **Other**.
data_format	String	Format of the data reported by the device. Options: **json** and **binary**.
manufacturer_name	String	Manufacturer name.
industry	String	Industry to which the device belongs.
description	String	Product description.
service_capabilities	Array of **ServiceCapability** objects	List of service capabilities of the device.
create_time	String	Time when a product was created on the platform. The format is yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-11 ServiceCapability

Parameter	Type	Description
service_id	String	**Parameter description**: service ID of the device. The value must be unique in a product. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.

Parameter	Type	Description
service_type	String	**Parameter description**: service type of the device. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
properties	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service. **Value**: The array length cannot exceed 500.
commands	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service. **Value**: The array length cannot exceed 500.
events	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service. Currently, event customization is not supported. You do not need to define this field when creating or modifying a product. **Value**: The array length cannot exceed 500.
description	String	**Parameter description**: device service description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
option	String	**Parameter description**: whether the device service is mandatory. Currently, this field is not a functional field and is used only for identification. **Options**: • **Master**: master service • **Mandatory**: mandatory service • **Optional**: optional service The default value is **Optional**. Default: **Optional**

Table 1-12 ServiceProperty

Parameter	Type	Description
property_name	String	**Parameter description**: device property name. The value must be unique in the device service. If device property names are used as keys in the device shadow JSON file, the names cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). A device shadow file that contains these special characters cannot be updated. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**
enum_list	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	String	**Parameter description**: minimum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	String	**Parameter description**: maximum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	Integer	**Parameter description**: maximum length of the device property.
step	Double	**Parameter description**: step of the device property.
unit	String	**Parameter description**: unit of the device property. **Value length**: a maximum of 16 characters Maximum: **16**

Parameter	Type	Description
method	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.
description	String	**Parameter description**: device property description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
default_value	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-13 ServiceCommand

Parameter	Type	Description
command_name	String	**Parameter description**: device command name. The value must be unique in the device service. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.
responses	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-14 ServiceCommandResponse

Parameter	Type	Description
response_name	String	**Parameter description**: name of the device command response. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.

Table 1-15 ServiceEvent

Parameter	Type	Description
event_type	String	**Parameter description**: device event type. The value must be unique in the device service. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-16 ServiceCommandPara

Parameter	Type	Description
para_name	String	**Parameter description**: parameter name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**
enum_list	Array of strings	**Parameter description**: list of enumerated values of the parameter.

Parameter	Type	Description
min	String	**Parameter description**: minimum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	String	**Parameter description**: maximum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	Integer	**Parameter description**: maximum length of the parameter.
step	Double	**Parameter description**: step of the parameter.
unit	String	**Parameter description**: unit of the parameter. **Value length**: a maximum of 16 characters Maximum: **16**
description	String	**Parameter description**: parameter description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Example Requests

Creates a product whose name is **Thermometer** and service is **temperature**. The product contains the **temperature** property and the **reboot** command.

```
POST https://{endpoint}/v5/iot/{project_id}/products

{
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "name" : "Thermometer.",
  "device_type" : "Thermometer.",
  "protocol_type" : "MQTT",
  "data_format" : "json",
  "service_capabilities" : [ {
    "service_id" : "Temperature.",
    "service_type" : "temperature",
    "properties" : [ {
      "property_name" : "temperature",
      "data_type" : "decimal",
      "required" : true,
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "centigrade",
      "method" : "RW",
      "description" : "force",
      "default_value" : {
        "color" : "red",
        "size" : 1
```



```
    }
   } ],
   "commands" : [ {
    "command_name" : "reboot",
    "paras" : [ {
     "para_name" : "force",
     "data_type" : "string",
     "required" : false,
     "enum_list" : null,
     "min" : "1",
     "max" : "100",
     "max_length" : 100,
     "step" : 0.1,
     "unit" : "km/h",
     "description" : "force"
    } ],
    "responses" : [ {
     "response_name" : "ACK",
     "paras" : [ {
      "para_name" : "force",
      "data_type" : "string",
      "required" : false,
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
     } ]
    } ]
   } ],
   "description" : "temperature",
   "option" : "Mandatory"
  } ],
  "manufacturer_name" : "ABC",
  "industry" : "smartCity",
  "description" : "this is a thermometer produced by Huawei",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
}
```

Example Responses

Status code: 201

Created

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name" : "testAPP01",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "name" : "Thermometer",
  "device_type" : "Thermometer",
  "protocol_type" : "MQTT",
  "data_format" : "json",
  "manufacturer_name" : "ABC",
  "industry" : "smartCity",
  "description" : "this is a thermometer produced by Huawei",
  "service_capabilities" : [ {
    "service_id" : "temperature",
    "service_type" : "temperature",
    "properties" : [ {
      "property_name" : "temperature",
      "required" : true,
      "data_type" : "decimal",
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
```



```
    "step" : 0.1,
    "unit" : "centigrade",
    "method" : "RW",
    "description" : "force",
    "default_value" : {
      "color" : "red",
      "size" : 1
    }
  } ],
  "commands" : [ {
    "command_name" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "required" : false,
      "data_type" : "string",
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ],
    "responses" : [ {
      "response_name" : "ACK",
      "paras" : [ {
        "para_name" : "force",
        "required" : false,
        "data_type" : "string",
        "enum_list" : null,
        "min" : "1",
        "max" : "100",
        "max_length" : 100,
        "step" : 0.1,
        "unit" : "km/h",
        "description" : "force"
      } ]
    } ]
  } ],
  "description" : "temperature",
  "option" : "Mandatory"
 } ],
 "create_time" : "20190303T081011Z"
}
```


Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.1.2 Query the Product List

Function

This API is used by an application to query the list of product models that have been imported to the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/products

Table 1-17 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-18 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. It is used by users who have multiple resource spaces to query the product list of a specified resource space. If this parameter is not carried, the list of all products of the user will be returned. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_name	No	String	**Parameter description**: product name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-19 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-20 Response body parameters

Parameter	Type	Description
products	Array of **ProductSummary** objects	Product information list.
page	**Page** object	Pagination information of the query results.

Table 1-21 ProductSummary

Parameter	Type	Description
app_id	String	Resource space ID.
app_name	String	Resource space name.
product_id	String	Product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	String	Product name.
device_type	String	Device type.
protocol_type	String	Protocol used by a device. Options: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, **ONVIF**, **OPC-UA**, **OPC-DA**, and **Other**.
data_format	String	Format of the data reported by the device. Options: **json** and **binary**.

Parameter	Type	Description
manufacturer_name	String	Manufacturer name.
industry	String	Industry to which the device belongs.
description	String	Product description.
create_time	String	Time when a product was created on the platform. The format is yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-22 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries all products in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/products
```

Example Responses

Status code: 200

Successful response

```
{
  "products" : [ {
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
    "app_name" : "testAPP01",
    "product_id" : "5ba24f5ebbe8f56f5a14f605",
    "name" : "Thermometer",
    "device_type" : "Thermometer",
    "protocol_type" : "MQTT",
    "data_format" : "json",
    "manufacturer_name" : "ABC",
    "industry" : "smartCity",
    "description" : "this is a thermometer produced by Huawei",
    "create_time" : "20190303T081011Z"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.1.3 Query a Product

Function

This API is used by an application to query details of a product model that has been created on the IoT platform, including the services, properties, and commands in the product model.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/products/{product_id}

Table 1-23 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
product_id	Yes	String	**Parameter description**: product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 1-24 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space that the product to be queried belongs to. If this parameter is not specified, the product in the **default resource space** will be queried. If there is no product in the default resource space, the earliest created product will be queried. If you have multiple resource spaces and do not want to specify this parameter, contact Huawei technical engineers to combine your resource spaces. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-25 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-26 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID.
app_name	String	Resource space name.
product_id	String	Product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	String	Product name.
device_type	String	Device type.

Parameter	Type	Description
protocol_type	String	Protocol used by a device. Options: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, **ONVIF**, **OPC-UA**, **OPC-DA**, and **Other**.
data_format	String	Format of the data reported by the device. Options: **json** and **binary**.
manufacturer_name	String	Manufacturer name.
industry	String	Industry to which the device belongs.
description	String	Product description.
service_capabilities	Array of **ServiceCapability** objects	List of service capabilities of the device.
create_time	String	Time when a product was created on the platform. The format is yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-27 ServiceCapability

Parameter	Type	Description
service_id	String	**Parameter description**: service ID of the device. The value must be unique in a product. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
service_type	String	**Parameter description**: service type of the device. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
properties	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service. **Value**: The array length cannot exceed 500.
commands	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service. **Value**: The array length cannot exceed 500.

Parameter	Type	Description
events	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service. Currently, event customization is not supported. You do not need to define this field when creating or modifying a product. **Value**: The array length cannot exceed 500.
description	String	**Parameter description**: device service description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
option	String	**Parameter description**: whether the device service is mandatory. Currently, this field is not a functional field and is used only for identification. **Options**: • **Master**: master service • **Mandatory**: mandatory service • **Optional**: optional service The default value is **Optional**. Default: **Optional**

Table 1-28 ServiceProperty

Parameter	Type	Description
property_name	String	**Parameter description**: device property name. The value must be unique in the device service. If device property names are used as keys in the device shadow JSON file, the names cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). A device shadow file that contains these special characters cannot be updated. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**

Parameter	Type	Description
enum_list	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	String	**Parameter description**: minimum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	String	**Parameter description**: maximum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	Integer	**Parameter description**: maximum length of the device property.
step	Double	**Parameter description**: step of the device property.
unit	String	**Parameter description**: unit of the device property. **Value length**: a maximum of 16 characters Maximum: **16**
method	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.
description	String	**Parameter description**: device property description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
default_value	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-29 ServiceCommand

Parameter	Type	Description
command_name	String	**Parameter description**: device command name. The value must be unique in the device service. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.
responses	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-30 ServiceCommandResponse

Parameter	Type	Description
response_name	String	**Parameter description**: name of the device command response. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.

Table 1-31 ServiceEvent

Parameter	Type	Description
event_type	String	**Parameter description**: device event type. The value must be unique in the device service. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-32 ServiceCommandPara

Parameter	Type	Description
para_name	String	**Parameter description**: parameter name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**
enum_list	Array of strings	**Parameter description**: list of enumerated values of the parameter.
min	String	**Parameter description**: minimum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	String	**Parameter description**: maximum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	Integer	**Parameter description**: maximum length of the parameter.
step	Double	**Parameter description**: step of the parameter.
unit	String	**Parameter description**: unit of the parameter. **Value length**: a maximum of 16 characters Maximum: **16**
description	String	**Parameter description**: parameter description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Example Requests

Queries details about a specified product.

```
GET https://{endpoint}/v5/iot/{project_id}/products/{product_id}
```


Example Responses

Status code: 200

Successful response


```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name" : "testAPP01",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "name" : "Thermometer",
  "device_type" : "Thermometer",
  "protocol_type" : "MQTT",
  "data_format" : "json",
  "manufacturer_name" : "ABC",
  "industry" : "smartCity",
  "description" : "this is a thermometer produced by Huawei",
  "service_capabilities" : [ {
    "service_id" : "temperature",
    "service_type" : "temperature",
    "properties" : [ {
      "property_name" : "temperature",
      "required" : true,
      "data_type" : "decimal",
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "centigrade",
      "method" : "RW",
      "description" : "force",
      "default_value" : {
        "color" : "red",
        "size" : 1
      }
    } ],
    "commands" : [ {
      "command_name" : "reboot",
      "paras" : [ {
        "para_name" : "force",
        "required" : false,
        "data_type" : "string",
        "enum_list" : null,
        "min" : "1",
        "max" : "100",
        "max_length" : 100,
        "step" : 0.1,
        "unit" : "km/h",
        "description" : "force"
      } ],
      "responses" : [ {
        "response_name" : "ACK",
        "paras" : [ {
          "para_name" : "force",
          "required" : false,
          "data_type" : "string",
          "enum_list" : null,
          "min" : "1",
          "max" : "100",
          "max_length" : 100,
          "step" : 0.1,
          "unit" : "km/h",
          "description" : "force"
        } ]
      } ]
    } ],
    "events" : [ {
      "event_type" : "reboot",
```

```
    "paras" : [ {
      "para_name" : "force",
      "required" : false,
      "data_type" : "string",
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ]
  } ],
  "description" : "temperature",
  "option" : "Mandatory"
 } ],
 "create_time" : "20190303T081011Z"
}
```


Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.1.4 Modify a Product

Function

This API is used by an application to modify details of a product model that has been imported to the IoT platform, including the services, properties, and commands in the product model. This API is used to modify the product but not to modify or install a codec. If the service definition in the product is modified and the corresponding codec exists on the platform, modify and reinstall the codec.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/products/{product_id}

Table 1-33 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
product_id	Yes	String	**Parameter description**: product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-34 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-35 Request body parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space that the product to be modified belongs to. If this parameter is not specified, the product in the **default resource space** will be modified. If there is no product in the default resource space, the earliest created product will be modified. If you have multiple resource spaces and do not want to specify this parameter, contact Huawei technical engineers to combine your resource spaces. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
name	No	String	**Parameter description**: product name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.

Parameter	Mandatory	Type	Description
device_type	No	String	**Parameter description**: device type. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
protocol_type	No	String	**Parameter description**: protocol used by the device. Do not change other protocol types to **CoAP**. **Options**: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, **ONVIF**, **OPC-UA**, **OPC-DA**, and **Other**.
data_format	No	String	**Parameter description**: format of the data reported by the device. **Options**: • **json**: JSON format • **binary**: binary code stream format
service_capabilities	No	Array of **ServiceCapability** objects	**Parameter description**: list of service capabilities of the device.
manufacturer_name	No	String	**Parameter description**: manufacturer name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
industry	No	String	**Parameter description**: industry to which the device belongs. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
description	No	String	**Parameter description**: product description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Table 1-36 ServiceCapability

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: service ID of the device. The value must be unique in a product. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
service_type	Yes	String	**Parameter description**: service type of the device. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
properties	No	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service. **Value**: The array length cannot exceed 500.
commands	No	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service. **Value**: The array length cannot exceed 500.
events	No	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service. Currently, event customization is not supported. You do not need to define this field when creating or modifying a product. **Value**: The array length cannot exceed 500.
description	No	String	**Parameter description**: device service description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Parameter	Mandatory	Type	Description
option	No	String	**Parameter description**: whether the device service is mandatory. Currently, this field is not a functional field and is used only for identification. **Options**: • **Master**: master service • **Mandatory**: mandatory service • **Optional**: optional service The default value is **Optional**. Default: **Optional**

Table 1-37 ServiceProperty

Parameter	Mandatory	Type	Description
property_name	Yes	String	**Parameter description**: device property name. The value must be unique in the device service. If device property names are used as keys in the device shadow JSON file, the names cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). A device shadow file that contains these special characters cannot be updated. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	Yes	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	No	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**

Parameter	Mandatory	Type	Description
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	No	String	**Parameter description**: minimum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the device property.
step	No	Double	**Parameter description**: step of the device property.
unit	No	String	**Parameter description**: unit of the device property. **Value length**: a maximum of 16 characters Maximum: **16**
method	Yes	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: device property description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
default_value	No	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-38 ServiceCommand

Parameter	Mandatory	Type	Description
command_name	Yes	String	**Parameter description**: device command name. The value must be unique in the device service. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.
responses	No	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-39 ServiceCommandResponse

Parameter	Mandatory	Type	Description
response_name	Yes	String	**Parameter description**: name of the device command response. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.

Table 1-40 ServiceEvent

Parameter	Mandatory	Type	Description
event_type	Yes	String	**Parameter description**: device event type. The value must be unique in the device service. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-41 ServiceCommandPara

Parameter	Mandatory	Type	Description
para_name	Yes	String	**Parameter description**: parameter name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.

Parameter	Mandatory	Type	Description
data_type	Yes	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	No	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the parameter.
min	No	String	**Parameter description**: minimum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the parameter.
step	No	Double	**Parameter description**: step of the parameter.
unit	No	String	**Parameter description**: unit of the parameter. **Value length**: a maximum of 16 characters Maximum: **16**
description	No	String	**Parameter description**: parameter description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Response Parameters

Status code: 200

Table 1-42 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID.
app_name	String	Resource space name.
product_id	String	Product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	String	Product name.
device_type	String	Device type.
protocol_type	String	Protocol used by a device. Options: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, **ONVIF**, **OPC-UA**, **OPC-DA**, and **Other**.
data_format	String	Format of the data reported by the device. Options: **json** and **binary**.
manufacturer_name	String	Manufacturer name.
industry	String	Industry to which the device belongs.
description	String	Product description.
service_capabilities	Array of **ServiceCapability** objects	List of service capabilities of the device.
create_time	String	Time when a product was created on the platform. The format is yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-43 ServiceCapability

Parameter	Type	Description
service_id	String	**Parameter description**: service ID of the device. The value must be unique in a product. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.

Parameter	Type	Description
service_type	String	**Parameter description**: service type of the device. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-$) are allowed.
properties	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service. **Value**: The array length cannot exceed 500.
commands	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service. **Value**: The array length cannot exceed 500.
events	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service. Currently, event customization is not supported. You do not need to define this field when creating or modifying a product. **Value**: The array length cannot exceed 500.
description	String	**Parameter description**: device service description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
option	String	**Parameter description**: whether the device service is mandatory. Currently, this field is not a functional field and is used only for identification. **Options**: • **Master**: master service • **Mandatory**: mandatory service • **Optional**: optional service The default value is **Optional**. Default: **Optional**

Table 1-44 ServiceProperty

Parameter	Type	Description
property_name	String	**Parameter description**: device property name. The value must be unique in the device service. If device property names are used as keys in the device shadow JSON file, the names cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). A device shadow file that contains these special characters cannot be updated. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**
enum_list	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	String	**Parameter description**: minimum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	String	**Parameter description**: maximum value of the device property. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	Integer	**Parameter description**: maximum length of the device property.
step	Double	**Parameter description**: step of the device property.
unit	String	**Parameter description**: unit of the device property. **Value length**: a maximum of 16 characters Maximum: **16**

Parameter	Type	Description
method	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.
description	String	**Parameter description**: device property description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.
default_value	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-45 ServiceCommand

Parameter	Type	Description
command_name	String	**Parameter description**: device command name. The value must be unique in the device service. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.
responses	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-46 ServiceCommandResponse

Parameter	Type	Description
response_name	String	**Parameter description**: name of the device command response. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.

Table 1-47 ServiceEvent

Parameter	Type	Description
event_type	String	**Parameter description**: device event type. The value must be unique in the device service. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
paras	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-48 ServiceCommandPara

Parameter	Type	Description
para_name	String	**Parameter description**: parameter name. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
data_type	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
required	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**
enum_list	Array of strings	**Parameter description**: list of enumerated values of the parameter.

Parameter	Type	Description
min	String	**Parameter description**: minimum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max	String	**Parameter description**: maximum value of the parameter. **Value length**: 1–16 characters Minimum: **1** Maximum: **16**
max_length	Integer	**Parameter description**: maximum length of the parameter.
step	Double	**Parameter description**: step of the parameter.
unit	String	**Parameter description**: unit of the parameter. **Value length**: a maximum of 16 characters Maximum: **16**
description	String	**Parameter description**: parameter description. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_?'#().,;&%@!-:$![]~/) are allowed.

Example Requests

Changes the product name to **Thermometer** and service to **temperature**.

```
PUT https://{endpoint}/v5/iot/{project_id}/products/{product_id}

{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "name" : "Thermometer.",
  "device_type" : "Thermometer",
  "protocol_type" : "MQTT",
  "data_format" : "json",
  "service_capabilities" : [ {
    "service_id" : "Temperature.",
    "service_type" : "temperature",
    "properties" : [ {
      "property_name" : "temperature",
      "data_type" : "decimal",
      "required" : true,
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "centigrade",
      "method" : "RW",
      "description" : "force",
      "default_value" : {
        "color" : "red",
        "size" : 1
      }
```



```
  } ],
  "commands" : [ {
    "command_name" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "data_type" : "string",
      "required" : false,
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ],
    "responses" : [ {
      "response_name" : "ACK",
      "paras" : [ {
        "para_name" : "force",
        "data_type" : "string",
        "required" : false,
        "enum_list" : null,
        "min" : "1",
        "max" : "100",
        "max_length" : 100,
        "step" : 0.1,
        "unit" : "km/h",
        "description" : "force"
      } ]
    } ]
  } ],
  "events" : [ {
    "event_type" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "data_type" : "string",
      "required" : false,
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ]
  } ],
  "description" : "temperature",
  "option" : "Mandatory"
 } ],
 "manufacturer_name" : "ABC",
 "industry" : "smartCity",
 "description" : "this is a thermometer produced by Huawei"
}
```

Example Responses

Status code: 200

Successful response

```
{
 "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
 "app_name" : "testAPP01",
 "product_id" : "5ba24f5ebbe8f56f5a14f605",
 "name" : "Thermometer",
 "device_type" : "Thermometer",
 "protocol_type" : "MQTT",
 "data_format" : "json",
 "manufacturer_name" : "ABC",
```



```
"industry" : "smartCity",
"description" : "this is a thermometer produced by Huawei",
"service_capabilities" : [ {
  "service_id" : "temperature",
  "service_type" : "temperature",
  "properties" : [ {
    "property_name" : "temperature",
    "required" : true,
    "data_type" : "decimal",
    "enum_list" : null,
    "min" : "1",
    "max" : "100",
    "max_length" : 100,
    "step" : 0.1,
    "unit" : "centigrade",
    "method" : "RW",
    "description" : "force",
    "default_value" : {
      "color" : "red",
      "size" : 1
    }
  } ],
  "commands" : [ {
    "command_name" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "required" : false,
      "data_type" : "string",
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ],
    "responses" : [ {
      "response_name" : "ACK",
      "paras" : [ {
        "para_name" : "force",
        "required" : false,
        "data_type" : "string",
        "enum_list" : null,
        "min" : "1",
        "max" : "100",
        "max_length" : 100,
        "step" : 0.1,
        "unit" : "km/h",
        "description" : "force"
      } ]
    } ]
  } ],
  "events" : [ {
    "event_type" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "required" : false,
      "data_type" : "string",
      "enum_list" : null,
      "min" : "1",
      "max" : "100",
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ]
  } ],
  "description" : "temperature",
  "option" : "Mandatory"
```



```
 } ],
 "create_time" : "20190303T081011Z"
}
```

Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.1.5 Delete a Product

Function

This API is used by an application to delete a specific product model that has been imported to the IoT platform. This API can delete only the product but not the associated codec. If the product has devices, the product cannot be deleted.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/products/{product_id}

Table 1-49 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
product_id	Yes	String	**Parameter description**: product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 1-50 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space that the product to be deleted belongs to. If this parameter is not specified, the product in the **default resource space** will be deleted. If there is no product in the default resource space, the earliest created product will be deleted. If you have multiple resource spaces and do not want to specify this parameter, contact Huawei technical engineers to combine your resource spaces. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-51 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a specified product.

```
DELETE https://{endpoint}/v5/iot/{project_id}/products/{product_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2 Device Management

1.4.2.1 Create a Device

Function

This API is used by an application to create a device on the IoT platform. Only the created device can connect to the IoT platform.

- You can specify the **gateway_id** parameter to create a device under a parent device. Currently, up to two levels of child devices are supported.
- This API supports initial device configuration. The API reads the product details provided in the **product_id** parameter in the request. If the default value of a product property is defined, the default value is written to the device shadow.
- You can also use the **shadow** parameter to specify the initial configuration for the device. After the initial configuration is specified, the system compares the property values specified by **service_id** and the desired section with the default values of the corresponding properties in the product. If they are different, the property values specified by the **shadow** parameter are written to the device shadow.
- This API can be used to create a single device. For details about how to register devices in batches, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices

Table 1-52 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-53 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-54 Request body parameters

Parameter	Mandatory	Type	Description
device_id	No	String	** Parameter description**: device ID, which is globally unique and uniquely identifies a device. If this parameter is carried, the platform will use the parameter value as the device ID. Otherwise, the platform will allocate a device ID, which is in the format of *product_idnode_id. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (*), and hyphens (-) are allowed. You are advised to use at least 4 characters.
node_id	Yes	String	**Parameter description**: device identifier. This parameter is set to the IMEI, MAC address, or serial number. The device node ID contains 1 to 64 characters, including letters, digits, hyphens (-), and underscores (*). Note: Information cannot be modified once it is hardcoded to NB-IoT modules. Therefore, the node ID of an NB-IoT device must be globally unique. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (*), and hyphens (-) are allowed. You are advised to use at least 4 characters.

Parameter	Mandatory	Type	Description
device_name	No	String	** Parameter description**: device name, which uniquely identifies a device in a resource space. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. You are advised to use at least 4 characters. Minimum: **1** Maximum: **256**
product_id	Yes	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
auth_info	No	**AuthInfo** object	**Parameter description**: access authentication information about the device.
description	No	String	**Parameter description**: device description. **Value**: The value can contain a maximum of 2048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**

Parameter	Mandatory	Type	Description
gateway_id	No	String	**Parameter description**: gateway ID, which is the device ID of the parent device. If this parameter is carried, a child device is created under the parent device, and the child device is not directly connected to the platform. In this case, the parent device must already exist on the platform. After the child device is created, the gateway ID of the child device is the value of this parameter. If this parameter is not carried, a device that is directly connected to the platform is created. After the device is created, **device_id** and **gateway_id** of the device are the same. Note: The platform supports up to two levels of child devices. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the device to create will belong. If this parameter is not specified, the device to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
extension_info	No	Object	**Parameter description**: extended information, which can be customized. If this parameter is specified for a child device during device creation, the platform will notify the gateway of the extended information using the MQTT API for notifying a gateway of new child device connection. The maximum size of the value is 1 KB.
shadow	No	Array of **InitialDesired** objects	**Parameter description**: initial configuration of the device. You can use this parameter to specify the initial configuration for the device. The system compares the property values specified by **service_id** and the desired section with the default values of the corresponding properties in the product. If they are different, the property values specified by the **shadow** parameter are written to the device shadow. The value of **service_id** and the properties in the desired section must be defined in the product model.

Table 1-55 AuthInfo

Parameter	Mandatory	Type	Description
auth_type	No	String	**Parameter description**: authentication type. If **auth_type** is not specified, the secret authentication is used. **Options**: • **SECRET**: The secret authentication is used. • **CERTIFICATES**: The certificate authentication is used.

Parameter	Mandatory	Type	Description
secret	No	String	**Parameter description**: device secret. This parameter is mandatory when **auth_type** is set to **SECRET**. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**
fingerprint	No	String	**Parameter description**: certificate fingerprint. This parameter is mandatory when **auth_type** is set to **CERTIFICATES**. If this parameter is not specified during device registration, the certificate fingerprint used during the first device access is used. **Value**: The value is a string of 40 or 64 hexadecimal characters.
secure_access	No	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. **Options**: • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Devices connected to the platform using insecure protocols may be spoofed. You are advised not to use this value. Default: **true**

Parameter	Mandatory	Type	Description
timeout	No	Integer	**Parameter description**: validity period for device access, in seconds. The default value is **0**. If the device has not been connected to the IoT platform and activated within the validity period, the IoT platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the platform does not delete the device registration information. Note: This parameter takes effect only for directly connected devices. Minimum: **0** Maximum: **2147483647** Default: **0**

Table 1-56 InitialDesired

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: device service ID, which is defined in the product model associated with the device. **Value**: The value can contain a maximum of 32 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.

Parameter	Mandatory	Type	Description
desired	Yes	Object	**Parameter description**: initial device properties in JSON format. The data is in key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model. Currently, only one-layer structure is supported, as shown in the example request. The property value is compared with the default value of the corresponding property of the product. If they are different, the property value set in this parameter is written to the device shadow. To delete the entire **desired** section, enter an empty object (for example, **"desired":{}**). To delete the desired value of a property, set the property value to **null** (for example, **{"temperature":null}**).

Response Parameters

Status code: 201

Table 1-57 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID. Maximum: **36**
app_name	String	Resource space name.
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.
node_id	String	Device identifier. This parameter is set to the IMEI, MAC address, or serial number.

Parameter	Type	Description
gateway_id	String	Gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates.
device_name	String	Device name.
node_type	String	Device node type. • **ENDPOINT**: an indirectly connected device. • **GATEWAY**: a directly connected device or gateway. • **UNKNOWN**: The device node type is unknown.
description	String	Device description.
fw_version	String	Firmware version of the device.
sw_version	String	Software version of the device.
device_sdk_version	String	Device SDK information.
auth_info	**AuthInfoRes** object	Access authentication information about the device.
product_id	String	Unique ID of the product associated with the device.
product_name	String	Name of the product associated with the device.
status	String	Device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FROZEN**: The device is frozen.
create_time	String	Time when the device was registered on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
connection_status_update_time	String	Time of the last device connection status change. Such statuses include **ONLINE**, **OFFLINE**, and **ABNORMAL**. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2015-12-12T12:12:122Z.

Parameter	Type	Description
active_time	String	Time when a device was activated. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2015-12-12T12:12:122Z.
tags	Array of **TagV5DTO** objects	List of device tags.
extension_info	Object	Extended device information, which can be customized. If this parameter is specified for a child device during device creation, the platform will notify the gateway of the extended information using the MQTT API for notifying a gateway of new child device connection.

Table 1-58 AuthInfoRes

Parameter	Type	Description
auth_type	String	**Parameter description**: authentication type. If **auth_type** is not specified, the secret authentication is used. **Options**: • **SECRET**: The secret authentication is used. • **CERTIFICATES**: The certificate authentication is used.
secret	String	**Parameter description**: device secret. This parameter is mandatory when **auth_type** is set to **SECRET**. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**

Parameter	Type	Description
secondary_secret	String	** Parameter description**: secondary device secret used when the master secret fails to pass the authentication. This parameter is valid when **auth_type** is set to **SECRET**. Unavailable for devices accessed using CoAP. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**
fingerprint	String	**Parameter description**: certificate fingerprint. This parameter is valid when **auth_type** is set to **CERTIFICATES**. If this parameter is not specified during device registration, the certificate fingerprint used during the first device access is used. **Value**: The value is a string of 40 or 64 hexadecimal characters.
secondary_fingerprint	String	** Parameter description**: secondary certificate fingerprint used when the master fingerprint fails to pass the authentication. This parameter is valid when **auth_type** is set to **CERTIFICATES**. **Value**: The value is a string of 40 or 64 hexadecimal characters.
secure_access	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. **Options**: • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Devices connected to the platform using insecure protocols may be spoofed. You are advised not to use this value. Default: **true**

Parameter	Type	Description
timeout	Integer	**Parameter description**: validity period for device access, in seconds. The default value is **0**. If the device has not been connected to the IoT platform and activated within the validity period, the IoT platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the platform does not delete the device registration information. Note: This parameter is only available for directly connected devices. Minimum: **0** Maximum: **2147483647** Default: **0**

Table 1-59 TagV5DTO

Parameter	Type	Description
tag_key	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
tag_value	String	**Parameter description**: tag value. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Example Requests

- Creates a device and sets the authentication type to secret authentication.

```
POST https://{endpoint}/v5/iot/{project_id}/devices

{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "device_name" : "dianadevice",
  "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
  "auth_info" : {
    "auth_type" : "SECRET",
    "secret" : "3b935a250c50dc2c6d481d048cefdc3c",
    "secure_access" : true
  },
  "description" : "watermeter device",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "extension_info" : {
    "aaa" : "xxx",
```



```
    "bbb" : 0
  },
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "temperature" : "60"
    }
  } ]
}
```

- Creates a device and set the authentication type to certificate authentication.

```
POST https://{endpoint}/v5/iot/{project_id}/devices

{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "device_name" : "dianadevice",
  "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
  "auth_info" : {
    "auth_type" : "CERTIFICATES",
    "fingerprint" : "dc0f1016f495157344ac5f1296335cff725ef22f",
    "secure_access" : true
  },
  "description" : "watermeter device",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "extension_info" : {
    "aaa" : "xxx",
    "bbb" : 0
  },
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "temperature" : "60"
    }
  } ]
}
```

- Creates a non-directly connected device.

```
POST https://{endpoint}/v5/iot/{project_id}/devices

{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "device_name" : "dianadevice",
  "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
  "description" : "watermeter device",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "gateway_id" : "5a332c1a-d14f-489c-a8e7-4753b1bb0872",
  "extension_info" : {
    "aaa" : "xxx",
    "bbb" : 0
  },
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "temperature" : "60"
    }
  } ]
}
```

Example Responses

Status code: 201

Created

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
```



```
 "app_name" : "testAPP01",
 "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
 "node_id" : "ABC123456789",
 "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
 "device_name" : "dianadevice",
 "node_type" : "ENDPOINT",
 "description" : "watermeter device",
 "fw_version" : "1.1.0",
 "sw_version" : "1.1.0",
 "auth_info" : {
   "auth_type" : "SECRET",
   "secret" : "3b93****fdc3c",
   "fingerprint" : "dc0f****f22f",
   "secure_access" : true,
   "timeout" : 0
 },
 "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
 "product_name" : "Thermometer",
 "status" : "INACTIVE",
 "create_time" : "20190303T081011Z",
 "connection_status_update_time" : null,
 "active_time" : null,
 "tags" : [ {
   "tag_key" : "testTagName",
   "tag_value" : "testTagValue"
 } ],
 "extension_info" : {
   "aaa" : "xxx",
   "bbb" : 0
 }
}
```


Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.2 Query the Device List

Function

This API is used by an application to query the device list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices

Table 1-60 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-61 Query Parameters

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
gateway_id	No	String	**Parameter description**: gateway ID, which is the device ID of the parent device. If this parameter is carried, child devices under the parent device are queried. By default, the child devices at the nearest lower level are queried. To query all levels of child devices, set **is_cascade_query** to **true**. If this parameter is not carried, all your devices are queried. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
is_cascade_query	No	Boolean	**Parameter description**: whether cascading query is performed. This parameter is valid only when **gateway_id** is carried. The default value is **false**. **Options**: • **true**: All child devices under the device whose ID is the value of **gateway_id** are queried. • **false**: The child devices at the nearest lower level under the device whose ID is the value of **gateway_id** are queried. Default: **false**
node_id	No	String	**Parameter description**: device identifier. This parameter is set to the IMEI, MAC address, or serial number. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
device_name	No	String	** Parameter description**: device name, which uniquely identifies a device in a resource space. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**
start_time	No	String	**Parameter description**: Records of devices registered after the value of this parameter are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
end_time	No	String	**Parameter description**: Records of devices registered before the value of this parameter are queried. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can specify this parameter to query devices in the specified resource space. If this parameter is not carried, devices in all resource spaces are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-62 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-63 Response body parameters

Parameter	Type	Description
devices	Array of **QueryDeviceSimplify** objects	Device information list.
page	**Page** object	Pagination information of the query results.

Table 1-64 QueryDeviceSimplify

Parameter	Type	Description
app_id	String	Resource space ID.
app_name	String	Resource space name.
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.
node_id	String	Device identifier. This parameter is set to the IMEI, MAC address, or serial number.
gateway_id	String	Gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates.
device_name	String	Device name.
node_type	String	Device node type. • **ENDPOINT**: an indirectly connected device. • **GATEWAY**: a directly connected device or gateway. • **UNKNOWN**: The device node type is unknown.
description	String	Device description.
fw_version	String	Firmware version of the device.

Parameter	Type	Description
sw_version	String	Software version of the device.
device_sdk_version	String	Device SDK information.
product_id	String	Unique ID of the product associated with the device.
product_name	String	Name of the product associated with the device.
status	String	Device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FROZEN**: The device is frozen.
tags	Array of **TagV5DTO** objects	List of device tags.

Table 1-65 TagV5DTO

Parameter	Type	Description
tag_key	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
tag_value	String	**Parameter description**: tag value. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Table 1-66 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.

Parameter	Type	Description
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries all devices in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/devices
```

Example Responses

Status code: 200

OK

```
{
  "devices" : [ {
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
    "app_name" : "testAPP01",
    "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "node_id" : "ABC123456789",
    "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "device_name" : "dianadevice",
    "node_type" : "ENDPOINT",
    "description" : "watermeter device",
    "fw_version" : "1.1.0",
    "sw_version" : "1.1.0",
    "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
    "product_name" : "Thermometer",
    "status" : "INACTIVE",
    "tags" : [ {
      "tag_key" : "testTagName",
      "tag_value" : "testTagValue"
    } ]
  } ],
  "page" : {
    "count" : 100,
    "marker" : "5c8f3d2d3df1f10d803adbda"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.3 Query a Device

Function

This API is used by an application to query the details of a specific device on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices/{device_id}

Table 1-67 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id][node_id]*. ***Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (*_*), and hyphens (-) are allowed.*

Request Parameters

Table 1-68 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-69 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID. Maximum: **36**
app_name	String	Resource space name.
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.

Parameter	Type	Description
node_id	String	Device identifier. This parameter is set to the IMEI, MAC address, or serial number.
gateway_id	String	Gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates.
device_name	String	Device name.
node_type	String	Device node type. • **ENDPOINT**: an indirectly connected device. • **GATEWAY**: a directly connected device or gateway. • **UNKNOWN**: The device node type is unknown.
description	String	Device description.
fw_version	String	Firmware version of the device.
sw_version	String	Software version of the device.
device_sdk_version	String	Device SDK information.
auth_info	**AuthInfoRes** object	Access authentication information about the device.
product_id	String	Unique ID of the product associated with the device.
product_name	String	Name of the product associated with the device.
status	String	Device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FROZEN**: The device is frozen.
create_time	String	Time when the device was registered on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Type	Description
connection_status_update_time	String	Time of the last device connection status change. Such statuses include **ONLINE**, **OFFLINE**, and **ABNORMAL**. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2015-12-12T12:12:122Z.
active_time	String	Time when a device was activated. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2015-12-12T12:12:122Z.
tags	Array of **TagV5DTO** objects	List of device tags.
extension_info	Object	Extended device information, which can be customized. If this parameter is specified for a child device during device creation, the platform will notify the gateway of the extended information using the MQTT API for notifying a gateway of new child device connection.

Table 1-70 AuthInfoRes

Parameter	Type	Description
auth_type	String	**Parameter description**: authentication type. If **auth_type** is not specified, the secret authentication is used. **Options**: • **SECRET**: The secret authentication is used. • **CERTIFICATES**: The certificate authentication is used.
secret	String	**Parameter description**: device secret. This parameter is mandatory when **auth_type** is set to **SECRET**. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**

Parameter	Type	Description
secondary_secret	String	** Parameter description**: secondary device secret used when the master secret fails to pass the authentication. This parameter is valid when **auth_type** is set to **SECRET**. Unavailable for devices accessed using CoAP. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**
fingerprint	String	**Parameter description**: certificate fingerprint. This parameter is valid when **auth_type** is set to **CERTIFICATES**. If this parameter is not specified during device registration, the certificate fingerprint used during the first device access is used. **Value**: The value is a string of 40 or 64 hexadecimal characters.
secondary_fingerprint	String	** Parameter description**: secondary certificate fingerprint used when the master fingerprint fails to pass the authentication. This parameter is valid when **auth_type** is set to **CERTIFICATES**. **Value**: The value is a string of 40 or 64 hexadecimal characters.
secure_access	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. **Options**: • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Devices connected to the platform using insecure protocols may be spoofed. You are advised not to use this value. Default: **true**

Parameter	Type	Description
timeout	Integer	**Parameter description**: validity period for device access, in seconds. The default value is **0**. If the device has not been connected to the IoT platform and activated within the validity period, the IoT platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the platform does not delete the device registration information. Note: This parameter is only available for directly connected devices. Minimum: **0** Maximum: **2147483647** Default: **0**

Table 1-71 TagV5DTO

Parameter	Type	Description
tag_key	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
tag_value	String	**Parameter description**: tag value. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Example Requests

Queries details about a specified device.

```
GET https://{endpoint}/v5/iot/{project_id}/devices/{device_id}
```

Example Responses

Status code: 200

OK

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name" : "testAPP01",
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
```



```
 "device_name" : "dianadevice",
 "node_type" : "ENDPOINT",
 "description" : "watermeter device",
 "fw_version" : "1.1.0",
 "sw_version" : "1.1.0",
 "auth_info" : {
   "auth_type" : "SECRET",
   "secret" : "3b935a250c50dc2c6d481d048cefdc3c",
   "fingerprint" : "dc0f1016f495157344ac5f1296335cff725ef22f",
   "secure_access" : true,
   "timeout" : 0
 },
 "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
 "product_name" : "Thermometer",
 "status" : "ONLINE",
 "create_time" : "20190303T081011Z",
 "connection_status_update_time" : "2019-03-03T08:10:111Z",
 "active_time" : "2019-03-03T08:10:111Z",
 "tags" : [ {
   "tag_key" : "testTagName",
   "tag_value" : "testTagValue"
 } ],
 "extension_info" : {
   "aaa" : "xxx",
   "bbb" : 0
 }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.4 Modify a Device

Function

This API is used by an application to modify the basic information of a specific device on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/devices/{device_id}

Table 1-72 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-73 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-74 Request body parameters

Parameter	Mandatory	Type	Description
device_name	No	String	** Parameter description**: device name, which uniquely identifies a device in a resource space. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. You are advised to use at least 4 characters. Minimum: **1** Maximum: **256**
description	No	String	**Parameter description**: device description. **Value**: The value can contain a maximum of 2048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**
extension_info	No	Object	**Parameter description**: extended information, which can be customized. If child device details are modified, the modified details are not delivered to the gateway.
auth_info	No	**AuthInfoWithoutSecret** object	**Parameter description**: access authentication information about the device.

Table 1-75 AuthInfoWithoutSecret

Parameter	Mandatory	Type	Description
secure_access	No	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. **Options**: • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Devices connected to the platform using insecure protocols may be spoofed. You are advised not to use this value. Default: **true**
timeout	No	Integer	**Parameter description**: validity period for device access, in seconds. The default value is **0**. If the device has not been connected to the IoT platform and activated within the validity period, the IoT platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the platform does not delete the device registration information. Note: This parameter takes effect only for directly connected devices. Minimum: **0** Maximum: **2147483647** Default: **0**

Response Parameters

Status code: 200

Table 1-76 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID. Maximum: **36**
app_name	String	Resource space name.
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.
node_id	String	Device identifier. This parameter is set to the IMEI, MAC address, or serial number.
gateway_id	String	Gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates.
device_name	String	Device name.
node_type	String	Device node type. • **ENDPOINT**: an indirectly connected device. • **GATEWAY**: a directly connected device or gateway. • **UNKNOWN**: The device node type is unknown.
description	String	Device description.
fw_version	String	Firmware version of the device.
sw_version	String	Software version of the device.
device_sdk_version	String	Device SDK information.
auth_info	**AuthInfoRes** object	Access authentication information about the device.
product_id	String	Unique ID of the product associated with the device.
product_name	String	Name of the product associated with the device.

Parameter	Type	Description
status	String	Device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FROZEN**: The device is frozen.
create_time	String	Time when the device was registered on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
connection_status_update_time	String	Time of the last device connection status change. Such statuses include **ONLINE**, **OFFLINE**, and **ABNORMAL**. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2015-12-12T12:12:122Z.
active_time	String	Time when a device was activated. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2015-12-12T12:12:122Z.
tags	Array of **TagV5DTO** objects	List of device tags.
extension_info	Object	Extended device information, which can be customized. If this parameter is specified for a child device during device creation, the platform will notify the gateway of the extended information using the MQTT API for notifying a gateway of new child device connection.

Table 1-77 AuthInfoRes

Parameter	Type	Description
auth_type	String	**Parameter description**: authentication type. If **auth_type** is not specified, the secret authentication is used. **Options**: • **SECRET**: The secret authentication is used. • **CERTIFICATES**: The certificate authentication is used.

Parameter	Type	Description
secret	String	**Parameter description**: device secret. This parameter is mandatory when **auth_type** is set to **SECRET**. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**
secondary_secret	String	** Parameter description**: secondary device secret used when the master secret fails to pass the authentication. This parameter is valid when **auth_type** is set to **SECRET**. Unavailable for devices accessed using CoAP. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values. This parameter is not returned when you query a device list. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**
fingerprint	String	**Parameter description**: certificate fingerprint. This parameter is valid when **auth_type** is set to **CERTIFICATES**. If this parameter is not specified during device registration, the certificate fingerprint used during the first device access is used. **Value**: The value is a string of 40 or 64 hexadecimal characters.
secondary_fingerprint	String	** Parameter description**: secondary certificate fingerprint used when the master fingerprint fails to pass the authentication. This parameter is valid when **auth_type** is set to **CERTIFICATES**. **Value**: The value is a string of 40 or 64 hexadecimal characters.

Parameter	Type	Description
secure_access	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. **Options**: • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Devices connected to the platform using insecure protocols may be spoofed. You are advised not to use this value. Default: **true**
timeout	Integer	**Parameter description**: validity period for device access, in seconds. The default value is **0**. If the device has not been connected to the IoT platform and activated within the validity period, the IoT platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the platform does not delete the device registration information. Note: This parameter is only available for directly connected devices. Minimum: **0** Maximum: **2147483647** Default: **0**

Table 1-78 TagV5DTO

Parameter	Type	Description
tag_key	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
tag_value	String	**Parameter description**: tag value. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Example Requests

Changes the device name to **device** and access type to secure access.


```
PUT https://{endpoint}/v5/iot/{project_id}/devices/{device_id}

{
  "device_name" : "device",
  "description" : "watermeter device",
  "extension_info" : {
    "aaa" : "xxx",
    "bbb" : 0
  },
  "auth_info" : {
    "secure_access" : true
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name" : "testAPP01",
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "device_name" : "dianadevice",
  "node_type" : "ENDPOINT",
  "description" : "watermeter device",
  "fw_version" : "1.1.0",
  "sw_version" : "1.1.0",
  "auth_info" : {
    "auth_type" : "SECRET",
    "secret" : "3b935a250c50dc2c6d481d048cefdc3c",
    "fingerprint" : "dc0f1016f495157344ac5f1296335cff725ef22f",
    "secure_access" : true,
    "timeout" : 0
  },
  "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
  "product_name" : "Thermometer",
  "status" : "ONLINE",
  "create_time" : "20190303T081011Z",
  "connection_status_update_time" : "2019-03-03T08:10:111Z",
  "active_time" : "2019-03-03T08:10:111Z",
  "tags" : [ {
    "tag_key" : "testTagName",
    "tag_value" : "testTagValue"
  } ],
  "extension_info" : {
    "aaa" : "xxx",
    "bbb" : 0
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized

Status Code	Description
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.5 Delete a Device

Function

This API is used by an application to delete a specific device from the IoT platform. If the device is connected to an indirectly connected device, you must delete the indirectly connected device first. This API can be used to delete a single device. For details about how to delete devices in batches, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/devices/{device_id}

Table 1-79 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-80 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a specified device.

```
DELETE https://{endpoint}/v5/iot/{project_id}/devices/{device_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.6 Reset a Device Secret

Function

This API is used by an application to reset a device secret. If the request contains a secret, the platform resets the device secret to the specified secret. If the request does not contain a secret, the platform automatically generates a new random secret and returns it.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/action

Table 1-81 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 1-82 Query Parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	**Parameter description**: operation performed on a device. **Options**: • **resetSecret**: resetting the secret. Note: Due to protocol restrictions, NB-IoT device secrets must be hexadecimal values.

Request Parameters

Table 1-83 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-84 Request body parameters

Parameter	Mandatory	Type	Description
secret	No	String	**Parameter description**: device secret. When this parameter is specified, the platform resets the device secret to the specified value. If this parameter is not specified, the platform automatically generates a device secret. **Value**: The value can contain 8 to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **8** Maximum: **32**
force_disconnect	No	Boolean	**Parameter description**: whether to forcibly disconnect a device. Only long-connection devices are supported. The default value is **false**. Default: **false**

Parameter	Mandatory	Type	Description
secret_type	No	String	**Parameter description**: type of the device secret to reset. **Options**: • **PRIMARY**: Resets the master secret. Primary secret used for secret-based device access authentication. • **SECONDARY**: Resets the sub secret. Secondary secret used when the master secret fails to pass the authentication. Unavailable for devices accessed using CoAP. Default: **PRIMARY**

Response Parameters

Status code: 200

Table 1-85 Response body parameters

Parameter	Type	Description
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. Maximum: **256**
secret	String	Device secret. Minimum: **8** Maximum: **32**

Parameter	Type	Description
secret_type	String	**Parameter description**: type of the device secret to reset. **Options**: • **PRIMARY**: Resets the master secret. Primary secret used for secret-based device access authentication. • **SECONDARY**: Resets the sub secret. Secondary secret used when the master secret fails to pass the authentication. Unavailable for devices accessed using CoAP. Default: **PRIMARY**

Example Requests

Resets the secret of a specified device. The new secret is **3b93****dc3c**. Re-establishing the connection from the device is not mandatory.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/action?action_id=resetSecret

{
  "secret" : "3b93****dc3c",
  "force_disconnect" : false
}
```

Example Responses

Status code: 200

OK

```
{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "secret" : "3b93****dc3c"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.7 Freeze a Device

Function

This API is used by an application to freeze a device. After a device is frozen, it cannot connect to the platform. You can use the API "Unfreezing a Device" to unfreeze the device. Currently, only devices that are directly connected to the platform can be frozen. This API can be used to freeze a single device. For details about how to freeze devices in batches, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/freeze

Table 1-86 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-87 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Freezes a specified device.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/freeze
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.8 Unfreeze a Device

Function

This API is used by an application to unfreeze a device. After the device is unfrozen, the device can connect to the platform and go online. This API can be used to unfreeze a single device. For details about how to unfreeze devices in batches, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/unfreeze

Table 1-88 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-89 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Unfreezes a specified device.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/unfreeze
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.9 Reset a Device Fingerprint

Function

This API is used by an application to reset a device fingerprint. A device fingerprint will be reset to the specified one. If a device fingerprint is not specified, it is set to the certificate fingerprint used when the device is connected to the platform for the first time.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/reset-fingerprint

Table 1-90 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-91 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-92 Request body parameters

Parameter	Mandatory	Type	Description
fingerprint	No	String	**Parameter description**: device fingerprint. If this parameter is specified, the platform resets the device fingerprint to the specified value. If this parameter is not specified, the device fingerprint is set to the certificate fingerprint used when the device is connected to the platform for the first time. **Value**: The value is a string of 40 or 64 hexadecimal characters.
force_disconnect	No	Boolean	**Parameter description**: whether to forcibly disconnect a device. Only long-connection devices are supported. The default value is **false**. Default: **false**
fingerprint_type	No	String	** Parameter description**: type of the device certificate fingerprint to reset. **Options**: • **PRIMARY**: Resets the master fingerprint. Primary fingerprint used for certificate-based device access authentication. • **SECONDARY**: Resets the sub fingerprint. Secondary fingerprint used when the master fingerprint fails to pass the authentication. Unavailable for devices accessed using CoAP. Default: **PRIMARY**

Response Parameters

Status code: 200

Table 1-93 Response body parameters

Parameter	Type	Description
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. Maximum: **256**
fingerprint	String	Device fingerprint.
fingerprint_type	String	** Parameter description**: type of the device certificate fingerprint to reset. **Options**: • **PRIMARY**: Resets the master fingerprint. • **SECONDARY**: Resets the sub fingerprint. Default: **PRIMARY**

Example Requests

Resets the device fingerprint. The new fingerprint is **dc0f****f22f**.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/reset-fingerprint

{
  "fingerprint" : "dc0f****f22f"
}
```

Example Responses

Status code: 200

OK

```
{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "fingerprint" : "dc0f****f22f"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found

Status Code	Description
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.10 Query Device List Flexibly

Function

API description

This API is used by an application to query a desired device list using SQL statements.

Constraints

- This API is supported only by **Standard and Enterprise editions**.
- The maximum TPS for an account to call this API is 1 (one request per second).

SQL-like syntax description

SQL-like statements include **select**, **from**, **where** (optional), **order by** (optional), and **limit** clauses (optional). The maximum length is 400 characters. The content in the clause is case sensitive, but keywords in SQL statements are case insensitive.

Example:

```
select * from device where device_id = 'as********' limit 0,5
```

SELECT clause

```
select [field]/[count(*)/count(1)] from device
```

field indicates the parameter to be obtained. Replace it with the response parameter name. You can also enter an asterisk (*) to obtain all parameters.

To obtain the number of queried devices, use **count(*)** or **count(1)**.

FROM clause

```
from device
```

The parameter following **from** indicates the name of the resource to be queried. Currently, **device** is supported.

WHERE clause (optional)

```
WHERE [condition1] AND [condition2]
```

Up to five conditions are supported. Conditions cannot be nested. For details about the parameters that support query, see **Description of query condition parameters** and **Supported operators**.

AND and **OR** are supported. For details about the priority, see the standard SQL syntax. By default, the priority of **AND** is higher than that of **OR**.

LIMIT clause (optional)

```
limit [offset,] rows
```

offset indicates the query offset. **rows** indicates the maximum number of rows in the query result. Examples:

- limit n ; Example: (select * from device limit 10)

Up to *n* records can be returned.

- limit m,n; Example: (select * from device limit 20,10) The query offset is **m**. Up to *n* records can be returned.

Constraints

The maximum value of **offset** is **500**, and the maximum value of **rows** is **50**. If the **limit** clause is not specified, **limit 10** is used by default.

ORDER BY clause (optional)

This clause is used for customizing sorting. Currently, the **marker** parameter supports custom sorting.

```
order by marker [asc]/[desc]
```

If the clause is not specified, random sorting is used by default.

Description of query condition parameters

Parameter	Type	Description	Value Range
app_id	string	Resource space ID.	The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
device_id	string	Device ID.	The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description	Value Range
gateway_id	string	Gateway ID.	The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	string	ID of the product associated with the device.	The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
device_name	string	Device name.	The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#(),&%@!-) are allowed.
node_id	string	Node ID.	The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
status	string	Device status.	**ONLINE**, **OFFLINE**, **ABNORMAL**, **INACTIVE**, and **FROZEN**
node_type	string	Device node type.	**GATEWAY** (directly connected device or gateway) and **ENDPOINT** (indirectly connected device)

Parameter	Type	Description	Value Range
tag_key	string	Tag key.	The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
tag_value	string	Tag value.	The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
sw_version	string	Software version.	The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
fw_version	string	Firmware version.	The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
group_id	string	Group ID.	The value can contain a maximum of 36 characters. Only hexadecimal characters and hyphens (-) are allowed.

Parameter	Type	Description	Value Range
create_time	string	Device registration time.	Format: yyyy-MM-dd'T'HH:mm:ss.SSS'Z'. Example: **2015-06-06T12:10:10.000Z**
marker	string	Result record ID.	The value is a string of 24 hexadecimal characters. Example: **ffffffffffffffffffffffff**

Supported operators

Operator	Parameter that Support the Operator
=	All
!=	All
>	**create_time** and **marker**
<	**create_time** and **marker**
like	**device_name**, **node_id**, **tag_key**, and **tag_value**
in	Fields except **tag_key** and **tag_value**
not in	Fields except **tag_key** and **tag_value**

SQL restrictions

- like: Only prefix match is supported. Suffix match or wildcard match is not supported. At least four characters must be contained for prefix match. Special characters cannot be contained. Only letters, digits, underscores (_), and hyphens (-) are allowed. The prefix must end with %.
- Only **count(*)** or **count(1)** is supported.
- Other SQL statements, such as nested SQL statements, **union**, **join**, and **alias**, are not supported.
- The SQL statement can contain up to 400 characters. Up to five request conditions are supported.
- The condition value cannot be null or an empty string.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/search/query-devices

Table 1-94 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-95 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-96 Request body parameters

Parameter	Mandatory	Type	Description
sql	Yes	String	SQL statement. For details, see the SQL-like syntax description. Minimum: **1** Maximum: **400**

Response Parameters

Status code: 200

Table 1-97 Response body parameters

Parameter	Type	Description
devices	Array of **SearchDevice** objects	Device query result list.
count	Long	Total number of records that meet query conditions. This parameter is returned only when **select count(*)/count(1)** is used.

Table 1-98 SearchDevice

Parameter	Type	Description
app_id	String	Resource space ID. Maximum: **36**
app_name	String	Resource space name.
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. Maximum: **256**
node_id	String	Device identifier. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **64**

Parameter	Type	Description
gateway_id	String	Gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates. Maximum: **256**
device_name	String	Device name. Maximum: **256**
node_type	String	Device node type. • **ENDPOINT**: an indirectly connected device. • **GATEWAY**: a directly connected device or gateway. • **UNKNOWN**: The device node type is unknown.
fw_version	String	Firmware version of the device. Maximum: **256**
sw_version	String	Software version of the device. Maximum: **256**
device_sdk_version	String	Device SDK information. Maximum: **256**
product_id	String	Unique ID of the product associated with the device.
product_name	String	Name of the product associated with the device.
groups	Object	Device group list.
status	String	Device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FROZEN**: The device is frozen.
tags	Object	List of device tags.
marker	String	Query result record ID.

Example Requests

Runs the SQL statement to query all devices.

```
POST https://{endpoint}/v5/iot/{project_id}/search/query-devices

{
  "sql" : "select * from device"
}
```

Example Responses

Status code: 200

OK

```
{
  "devices" : [ {
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
    "marker" : "5c8f3d2d3df1f10d803adbda",
    "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "node_id" : "ABC123456789",
    "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "device_name" : "dianadevice",
    "node_type" : "ENDPOINT",
    "fw_version" : "1.1.0",
    "sw_version" : "1.1.0",
    "device_sdk_version" : "1.1.0",
    "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
    "status" : "INACTIVE",
    "tags" : [ {
      "tag_key" : "testTagName",
      "tag_value" : "testTagValue"
    } ]
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.2.11 Query the List of Device Groups to Which a Specified Device Is Added

Function

This API is used by an application to query the list of device groups to which a device in the IoT platform is added. This API is supported only by standard and enterprise editions.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/list-device-group

Table 1-99 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id][node_id]*. ***Value****: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_)*, and hyphens (-) are allowed.

Request Parameters

Table 1-100 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-101 Response body parameters

Parameter	Type	Description
device_groups	Array of **ListDeviceGroupSummary** objects	Device group information list.

Table 1-102 ListDeviceGroupSummary

Parameter	Type	Description
group_id	String	Device group ID. The ID is unique and is allocated by the platform during device group creation.
name	String	Device group name, which must be unique in a single resource space.
description	String	Device group description.
super_group_id	String	ID of the parent device group.
group_type	String	**Parameter description**: device group type. A static device group is used by default. When a dynamic device group is used, enter the dynamic device group rule.

Example Requests

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/list-device-group
```

Example Responses

Status code: 200

OK

```
{
  "device_groups" : [ {
    "group_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
    "name" : "GroupA",
    "description" : "Group A",
    "super_group_id" : "04ed32dc1b0025b52fe3c01a27c2b0a8",
    "group_type" : "STATIC"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.3 Device Message

1.4.3.1 Deliver a Message to a Device

Function

This API is used by an application to deliver a message to a device. After an application delivers a message to the platform, the platform returns a response to the application and then sends the message to the device. The response returned by the platform may not be the device receiving result. Call the API **Pushing a Device Message Status Change Notification** for the platform to push the device receiving result to the application. Notes:

- This API is only used to deliver messages to MQTT devices.
- This API supports message delivery to a single device. For details about how to deliver messages to multiple devices, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/messages

Table 1-103 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: ID of the device to which the message is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-104 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-105 Request body parameters

Parameter	Mandatory	Type	Description
message_id	No	String	**Parameter description**: message ID, which is user-defined (UUID is recommended). The ID must be unique under a device. Otherwise, an error is returned. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Maximum: **128**

Parameter	Mandatory	Type	Description
name	No	String	**Parameter description**: message name. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **128**
message	Yes	Object	**Parameter description**: message content. String and JSON formats are supported.
properties	No	**PropertiesDTO** object	**Parameter description**: property parameter of the message delivered to the device.
encoding	No	String	**Parameter description**: encoding format for the message content. The default value is **none**. **Options**: • none • base64 Default: **none**
payload_format	No	String	**Parameter description**: valid payload format. This parameter is valid when the encoding format for the message content is **none**. The default value is **standard**, indicating that the message content is encapsulated in the standard format. **Options**: • standard • raw: The message content is directly delivered as the payload. Default: **standard**

Parameter	Mandatory	Type	Description
topic	No	String	**Parameter description**: (optional) suffix of the custom topic through which the message is delivered to the device. This parameter is applicable only to MQTT devices. You can only enter topics configured on the tenant product page. Otherwise, the verification will fail. The prefix that the platform adds to a message topic is $oc/devices/*{device_id}*/user/. You can add a part to the prefix. For example, if you add **messageDown** to the prefix, the topic is $oc/devices/*{device_id}*/user/messageDown. Replace *device_id* with the actual gateway ID of the device. If you specify the topic, messages are delivered to the device through this topic. If you do not specify the topic, messages are delivered to the device through the default topic. The default topic is $oc/devices/*{device_id}*/sys/messages/down. Either this parameter or the **topic_full_name** parameter can be specified. Maximum: **128**

Parameter	Mandatory	Type	Description
topic_full_name	No	String	**Parameter description**: (optional) name of the topic through which the message is delivered to the device. You can specify a custom topic. The platform does not check whether the topic is defined on the platform and transparently transmits the topic to the device. The device needs to subscribe to the specified topic first. Either this field or the **topic** parameter can be specified. Maximum: **128**
ttl	No	Integer	**Parameter description**: aging time of the caches of delivered messages on the platform, in minutes. The default value is **1440**. The value of **ttl** must be a multiple of 5, that is, the granularity is 5 minutes. If this parameter is set to **0**, messages are not cached. The maximum cache duration is 1,440 minutes. For longer cache duration, you need to submit a request in advance, or the delivery will fail. Minimum: **0** Default: **1440**

Table 1-106 PropertiesDTO

Parameter	Mandatory	Type	Description
correlation_data	No	String	**Parameter description**: data in the request and response modes of MQTT 5.0. This parameter is optional. You can use this parameter to configure data in the request and response modes of MQTT. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
response_topic	No	String	**Parameter description**: response topic in the request and response modes of MQTT 5.0. This parameter is optional. You can use this parameter to configure the response topic in the request and response modes of MQTT. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_-?=$#+/) are allowed. Maximum: **128**
user_properties	No	Array of **UserPropDTO** objects	**Parameter description**: custom property. This parameter is optional. You can use this parameter to configure custom properties. A maximum of 20 customized attributes can be configured.

Table 1-107 UserPropDTO

Parameter	Mandatory	Type	Description
prop_key	No	String	**Parameter description**: key of the custom property. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
prop_value	No	String	**Parameter description**: value of the custom property. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-.

Response Parameters

Status code: 201

Table 1-108 Response body parameters

Parameter	Type	Description
message_id	String	Message ID, which is specified by you (UUID is recommended). The message ID must be unique under a device. If the ID is not specified, the platform automatically generates one.
result	**MessageResult** object	Result of delivering the message. The value is in JSON format.

Table 1-109 MessageResult

Parameter	Type	Description
status	String	Message status, including **PENDING**, **DELIVERED**, **FAILED**, and **TIMEOUT**. If the device is offline, the platform caches the message and returns **PENDING**. The platform delivers the message after the device reports data. If the device is online, the platform directly delivers the message. If the message is successfully delivered, the platform returns **DELIVERED**. If the message fails to be delivered, the platform returns **FAILED**. If the message is not delivered to the device within the default time (one day), the platform sets the message status to **TIMEOUT**. In addition, an application can subscribe to the message execution result. The platform pushes the result to the application.

Parameter	Type	Description
created_time	String	UTC time when the message was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
finished_time	String	UTC time when the message was delivered to the device, failed to be delivered, or timed out. The value is in the format of yyyyMMdd'T'HHmmss'Z'.

Example Requests

- Creates a message and delivers it through the default topic of the platform.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/messages

{
  "message_id" : "99b32da9-cd17-4cdf-a286-f6e849cbc364",
  "name" : "messageName",
  "message" : "HelloWorld"
}
```

- Creates a message and delivers it through a custom topic on the platform.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/messages

{
  "message_id" : "99b32da9-cd17-4cdf-a286-f6e849cbc364",
  "name" : "messageName",
  "message" : "HelloWorld",
  "topic" : "testTopic"
}
```

- Creates a message and delivers it through a custom topic.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/messages

{
  "message_id" : "99b32da9-cd17-4cdf-a286-f6e849cbc364",
  "name" : "messageName",
  "message" : "HelloWorld",
  "topic_full_name" : "/test/customTopic/testTopic"
}
```

Example Responses

Status code: 201

Created

```
{
  "message_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "result" : {
    "status" : "PENDING",
    "created_time" : "20151212T121212Z",
    "finished_time" : "20151212T121213Z"
  }
}
```


Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.3.2 Query Device Messages

Function

This API is used by an application to query messages delivered by the platform to a device. By default, the platform stores up to 20 messages for each device. If the number of messages exceeds 20, the earliest messages will be overwritten by subsequent messages.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices/{device_id}/messages

Table 1-110 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: ID of the device to which the message is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-111 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-112 Response body parameters

Parameter	Type	Description
device_id	String	Unique device ID, which is allocated by the platform during device registration.
messages	Array of **DeviceMessage** objects	List of device messages.

Table 1-113 DeviceMessage

Parameter	Type	Description
message_id	String	Device message ID, which is unique and is allocated by the platform during message delivery.
name	String	Message name specified during device message delivery.
message	Object	Message content.
encoding	String	Encoding format for the message content. Possible options are **none** and **base64**. The default value is **none**. **base64** supports only transparent transmission.
payload_format	String	Valid payload format. This parameter is valid when **encoding** is **none**. Possible options are **standard** and **raw**. The default value is **standard**, indicating that the message content is encapsulated in the standard format. **raw** indicates that the message content is directly delivered.
topic	String	Message topic.
properties	**PropertiesDTO** object	Property parameter of the message delivered to the device.
status	String	Message status, including **PENDING**, **DELIVERED**, **FAILED**, and **TIMEOUT**. **PENDING**: The message is cached and delivered after the device goes online. **DELIVERED**: The message is sent successfully. **FAILED**: The message fails to be sent. **TIMEOUT**: If a message is not delivered to the device within the default period (one day), the platform sets the message status to **TIMEOUT**.

Parameter	Type	Description
error_info	**ErrorInfoDTO** object	Message delivery error information, including **error_code** and **error_msg**. **error_code** includes **IOTDA.014016** and **IOTDA.014112**. **IOTDA.014016** indicates that the device is offline. **IOTDA.014112** indicates that the device does not subscribe to any topic.
created_time	String	UTC time when the message was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
finished_time	String	UTC time when the message was delivered to the device or timed out. The value is in the format of yyyyMMdd'T'HHmmss'Z'.

Table 1-114 PropertiesDTO

Parameter	Type	Description
correlation_data	String	**Parameter description**: data in the request and response modes of MQTT 5.0. This parameter is optional. You can use this parameter to configure data in the request and response modes of MQTT. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
response_topic	String	**Parameter description**: response topic in the request and response modes of MQTT 5.0. This parameter is optional. You can use this parameter to configure the response topic in the request and response modes of MQTT. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_-?=$#+/) are allowed. Maximum: **128**
user_properties	Array of **UserPropDTO** objects	**Parameter description**: custom property. This parameter is optional. You can use this parameter to configure custom properties. A maximum of 20 customized attributes can be configured.

Table 1-115 UserPropDTO

Parameter	Type	Description
prop_key	String	**Parameter description**: key of the custom property. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
prop_value	String	**Parameter description**: value of the custom property. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-.

Table 1-116 ErrorInfoDTO

Parameter	Type	Description
error_code	String	**Parameter description**: error code, including **IOTDA.014016** and **IOTDA.014112**. **IOTDA.014016** indicates that the device is offline. **IOTDA.014112** indicates that the device does not subscribe to any topic.
error_msg	String	** Parameter description**: error message. The device is offline or does not subscribe to any topic.

Example Requests

Queries all messages in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/messages
```

Example Responses

Status code: 200

OK

```
{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "messages" : [ {
    "message_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
    "name" : "message_name",
    "message" : "string",
    "topic" : "string",
    "status" : "PENDING",
    "created_time" : "20151212T121212Z",
    "finished_time" : "20151212T121212Z"
  } ]
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.3.3 Query a Message by Message ID

Function

This API is used by an application to query a message with a specified ID delivered by the platform to a device.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices/{device_id}/messages/{message_id}

Table 1-117 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: ID of the device to which the message is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
message_id	Yes	String	**Parameter description**: ID of the delivered message. The ID is unique and is allocated by the platform during message delivery. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-118 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-119 Response body parameters

Parameter	Type	Description
message_id	String	Device message ID, which is unique and is allocated by the platform during message delivery.
name	String	Message name specified during device message delivery.
message	Object	Message content.
encoding	String	Encoding format for the message content. Possible options are **none** and **base64**. The default value is **none**. **base64** supports only transparent transmission.
payload_format	String	Valid payload format. This parameter is valid when **encoding** is **none**. Possible options are **standard** and **raw**. The default value is **standard**, indicating that the message content is encapsulated in the standard format. **raw** indicates that the message content is directly delivered.
topic	String	Message topic.
properties	**PropertiesDTO** object	Property parameter of the message delivered to the device.

Parameter	Type	Description
status	String	Message status, including **PENDING**, **DELIVERED**, **FAILED**, and **TIMEOUT**. **PENDING**: The message is cached and delivered after the device goes online. **DELIVERED**: The message is sent successfully. **FAILED**: The message fails to be sent. **TIMEOUT**: If a message is not delivered to the device within the default period (one day), the platform sets the message status to **TIMEOUT**.
error_info	**ErrorInfoDTO** object	Message delivery error information, including **error_code** and **error_msg**. **error_code** includes **IOTDA.014016** and **IOTDA.014112**. **IOTDA.014016** indicates that the device is offline. **IOTDA.014112** indicates that the device does not subscribe to any topic.
created_time	String	UTC time when the message was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
finished_time	String	UTC time when the message was delivered to the device or timed out. The value is in the format of yyyyMMdd'T'HHmmss'Z'.

Table 1-120 PropertiesDTO

Parameter	Type	Description
correlation_data	String	**Parameter description**: data in the request and response modes of MQTT 5.0. This parameter is optional. You can use this parameter to configure data in the request and response modes of MQTT. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
response_topic	String	**Parameter description**: response topic in the request and response modes of MQTT 5.0. This parameter is optional. You can use this parameter to configure the response topic in the request and response modes of MQTT. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and special characters (_-?=$#+/) are allowed. Maximum: **128**

Parameter	Type	Description
user_properties	Array of **UserPropDTO** objects	**Parameter description**: custom property. This parameter is optional. You can use this parameter to configure custom properties. A maximum of 20 customized attributes can be configured.

Table 1-121 UserPropDTO

Parameter	Type	Description
prop_key	String	**Parameter description**: key of the custom property. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
prop_value	String	**Parameter description**: value of the custom property. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-.

Table 1-122 ErrorInfoDTO

Parameter	Type	Description
error_code	String	**Parameter description**: error code, including **IOTDA.014016** and **IOTDA.014112**. **IOTDA.014016** indicates that the device is offline. **IOTDA.014112** indicates that the device does not subscribe to any topic.
error_msg	String	** Parameter description**: error message. The device is offline or does not subscribe to any topic.

Example Requests

Queries details about a specified message.

```
GET https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/messages/{message_id}
```

Example Responses

Status code: 200

OK


```
{
  "message_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "name" : "message_name",
  "message" : "string",
  "topic" : "string",
  "status" : "PENDING",
  "created_time" : "20151212T121212Z",
  "finished_time" : "20151212T121212Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.4 Device Command APIs

1.4.4.1 Synchronous Device Command

1.4.4.1.1 Deliver a Command to a Device

Function

A product model defines commands that the IoT platform can deliver to devices. An application can call this API to deliver commands to a specific device to control the device synchronously. The platform synchronously sends commands to the device and returns the command execution result to the application. If the device does not respond, the platform returns a timeout message to the application. The timeout interval is 20 seconds. If the command delivery takes more than 20 seconds, use the API **Delivering a Message to a Device**. Notes:

- The API can only be used to deliver commands to MQTT devices but not NB-IoT devices.
- This API supports synchronous command delivery to a single device. For details about how to deliver commands to multiple devices synchronously, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/commands

Table 1-123 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: ID of the device to which the message is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-124 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-125 Request body parameters

Parameter	Mandatory	Type	Description
service_id	No	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. **Value**: The value can contain a maximum of 64 characters. Maximum: **64**
command_name	No	String	**Parameter description**: command name, which is defined in the product model associated with the device. **Value**: The value can contain a maximum of 128 characters. Maximum: **128**

Parameter	Mandatory	Type	Description
paras	Yes	Object	**Parameter description**: command that will be executed. The command is in JSON format (key-value pairs). If **service_id** is specified, each key is a **paraName** in commands in the product model. If **service_id** is left empty, custom command format is used. **{"value":"1"}** is an example device command. The specific format depends on the application and device. This parameter supports only the JSON format and does not support character strings. Maximum: **261952**

Response Parameters

Status code: 200

Table 1-126 Response body parameters

Parameter	Type	Description
command_id	String	Device command ID, which is unique and is allocated by the platform during command delivery.
response	Object	Command execution result reported by the device. The value is in JSON format. The specific format depends on the application and device.
error_code	String	Error code of a command delivery exception.
error_msg	String	Error message of a command delivery exception.

Example Requests

Creates a command. The command name is **ON_OFF**, and the command is **ON**.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/commands

{
  "service_id" : "reboot",
  "command_name" : "ON_OFF",
```



```
  "paras" : {
    "value" : "ON"
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "command_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "response" : {
    "result_code" : 0,
    "response_name" : "COMMAND_RESPONSE",
    "paras" : {
      "result" : "success"
    }
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.4.2 Asynchronous Device Command

1.4.4.2.1 Deliver an Asynchronous Command

Function

A product model defines commands that the IoT platform can deliver to devices. An application can call this API to deliver asynchronous commands to a specific device to control the device. The platform delivers commands to the device and asynchronously returns the command execution result to the application. The command execution result supports data forwarding. After the application calls the API for creating a rule triggering condition (**resource** is set to **device.command.status** and **event** to **update**) and the API for creating a rule

action and activates a rule, the platform pushes the result to the server specified by the rule when the command status changes. The server can be a user-defined HTTP server, AMQP server, or Huawei Cloud storage server. For details, see **Pushing a Device Command Status Change Notification**. Notes:

- This API is used only to deliver commands asynchronously to NB-IoT devices.
- This API supports asynchronous command delivery to a single device. For details about how to deliver commands to multiple devices asynchronously, see **Creating a Batch Task**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/devices/{device_id}/async-commands

Table 1-127 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: ID of the device to which the command is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-128 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-129 Request body parameters

Parameter	Mandatory	Type	Description
service_id	No	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. This parameter is mandatory if the device requires codecs to parse commands. **Value**: The value can contain a maximum of 64 characters. Maximum: **64**

Parameter	Mandatory	Type	Description
command_name	No	String	**Parameter description**: command name, which is defined in the product model associated with the device. This parameter is mandatory if the device requires codecs to parse commands. **Value**: The value can contain a maximum of 128 characters. Maximum: **128**
paras	Yes	Object	**Parameter description**: command executed by the device. The command is in JSON format (key-value pairs). If **service_id** is specified, each key is a **paraName** in commands in the product model. If **service_id** is left empty, custom command format is used. **{"value":"1"}** is an example device command. The specific format depends on the application and device. The maximum size of the request object is 256 KB. Maximum: **261909**
expire_time	No	Integer	**Parameter description**: duration for the platform to cache commands. The unit is second. The platform can cache up to 20 messages (that is, up to 20 commands in the **PENDING** state). This parameter is valid only when **send_strategy** is set to **delay**. By default, commands are cached for 24 hours and can be cached for up to 2 days. Minimum: **0** Maximum: **172800**

Parameter	Mandatory	Type	Description
send_strategy	Yes	String	**Parameter description**: delivery policy. By default, commands are cached and then delivered. **Options**: • **immediately**: The command is delivered immediately. In this case, **expire_time** is invalid. • **delay**: The command is cached and delivered after the device reports data or goes online. If **expire_time** is set to **0** or not specified, the command is cached for 24 hours by default.

Response Parameters

Status code: 200

Table 1-130 Response body parameters

Parameter	Type	Description
device_id	String	Unique device ID, which is allocated by the platform during device registration.
command_id	String	Device command ID, which is unique and is allocated by the platform during command delivery.
service_id	String	ID of the device service that the device command belongs to, which is defined in the product model associated with the device.
command_name	String	Command name, which is defined in the product model associated with the device.
paras	Object	Command executed by the device. The command is in JSON format (key-value pairs). If **service_id** is specified, each key is a **paraName** in commands in the product model. If **service_id** is left empty, custom command format is used. **{"value":"1"}** is an example device command. The specific format depends on the application and device.
expire_time	Integer	Duration for the platform to cache commands, in seconds.

Parameter	Type	Description
status	String	Device command status. If the command is cached, **PENDING** is returned. If the command is delivered to the device, **SENT** is returned.
created_time	String	UTC time when the command was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
send_strategy	String	Delivery policy. **immediately** indicates that the command is delivered immediately. **delay** indicates that the command is cached and delivered when the device reports data or goes online.

Example Requests

Creating an asynchronous command. The command name is **ON_OFF**, and the command is **ON**.

```
POST https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/async-commands

{
  "service_id" : "reboot",
  "command_name" : "ON_OFF",
  "paras" : {
    "value" : "ON"
  },
  "expire_time" : 0,
  "send_strategy" : "immediately"
}
```

Example Responses

Status code: 200

OK

```
{
  "device_id" : "c1224afb-e9f0-4916-8220-b6bab568e888",
  "command_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "service_id" : "Switch",
  "command_name" : "ON_OFF",
  "send_strategy" : "immediately",
  "paras" : {
    "value" : "ON"
  },
  "expire_time" : 0,
  "status" : "SENT",
  "created_time" : "20151212T121212Z"
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.4.2.2 Query a Command with a Specific ID

Function

This API is used to query a command with a specific ID.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices/{device_id}/async-commands/{command_id}

Table 1-131 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: ID of the device to which the command is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
command_id	Yes	String	**Parameter description**: command ID, which uniquely identifies a message. The value of this parameter is allocated by the platform during command delivery. **Value**: The value can contain a maximum of 100 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-132 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-133 Response body parameters

Parameter	Type	Description
device_id	String	Unique device ID, which is allocated by the platform during device registration.
command_id	String	Device command ID, which is unique and is allocated by the platform during command delivery.
service_id	String	ID of the device service that the device command belongs to, which is defined in the product model associated with the device.
command_name	String	Command name, which is defined in the product model associated with the device.
paras	Object	Command executed by the device. The command is in JSON format (key-value pairs). If **service_id** is specified, each key is a **paraName** in commands in the product model. If **service_id** is left empty, custom command format is used. **{"value":"1"}** is an example device command. The specific format depends on the application and device.
expire_time	Integer	Duration for the platform to cache commands, in seconds.

Parameter	Type	Description
status	String	Command status. **PENDING**: The command is not delivered and is cached on the platform. **EXPIRED**: The command has expired, that is, the cache time exceeds the value of **expire_time**. **SENT**: The command is being delivered. **DELIVERED**: The command has been delivered. **SUCCESSFUL**: The command has been executed. **FAILED**: The command fails to be executed. **TIMEOUT**: After a command is delivered, no response is received from the device or the response times out.
result	Object	Detailed command execution result, which is returned by the device in JSON format.
created_time	String	UTC time when the command was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
sent_time	String	UTC time when the platform sent the command. If the command was sent immediately, the value was the same as the command creation time. If the command had been cached, the value was the time when the command was actually sent. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
delivered_time	String	UTC time when the device received the command. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
send_strategy	String	Delivery policy. **immediately** indicates that the command is delivered immediately. **delay** indicates that the command is cached and delivered when the device reports data or goes online.
response_time	String	UTC time when the device responded to the command. The value is in the format of yyyyMMdd'T'HHmmss'Z'.

Example Requests

Querying a command with a specific ID.

```
GET https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/async-commands/{command_id}
```

Example Responses

Status code: 200

OK


```
{
  "device_id" : "c1224afb-e9f0-4916-8220-b6bab568e888",
  "command_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "service_id" : "Switch",
  "command_name" : "ON_OFF",
  "paras" : {
    "value" : "ON"
  },
  "expire_time" : 0,
  "send_strategy" : "immediately",
  "created_time" : "20151212T121212Z",
  "status" : "DELIVERED",
  "result" : {
    "code" : 200
  },
  "sent_time" : "20151212T121212Z",
  "delivered_time" : "20151212T121212Z",
  "response_time" : "20151212T131312Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.5 Device Property

1.4.5.1 Modify Device Properties

Function

A product model defines properties that the platform can deliver to devices. This API is used by an application to deliver properties to a specific device. The platform sends the properties to the device in synchronous mode and returns the execution result synchronously to the application. This API can only be used for MQTT devices. NB-IoT devices are not supported.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/devices/{device_id}/properties

Table 1-134 Path Parameters

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: ID of the device to which the property is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-135 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-136 Request body parameters

Parameter	Mandatory	Type	Description
services	No	Object	**Parameter description**: properties that will be modified. The value is in JSON format (key-value pairs). If **service_id** is specified, each key is a **paraName** of a property in the product model. If **service_id** is left empty, custom property format is used. **[{"service_id": "Temperature","properties": {"value": 57}},{"service_id": "Battery","properties": {"level": 80}}]** is an example. The specific format is determined by the application and device. The maximum length is 256 KB. Maximum: **262144**

Response Parameters

Status code: 200

Table 1-137 Response body parameters

Parameter	Type	Description
request_id	String	An ID that uniquely identifies a device property update request. It is unique and is allocated by the platform when delivering a command for device property update.
response	Object	Property execution result reported by the device. The value is in JSON format. The specific format depends on the application and device.
error_code	String	Error code of a property update exception.
error_msg	String	Error message of a property update exception.

Example Requests

Delivers device properties. The property of the **Temperature** service is **value** and the value is **57**. The property of the **Batter** service is **level** and the value is **80**.

```
PUT https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/properties

{
  "services" : [ {
    "service_id" : "Temperature",
    "properties" : {
      "value" : 57
    }
  }, {
    "service_id" : "Battery",
    "properties" : {
      "level" : 80
    }
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "response" : {
    "result_code" : 0,
    "result_desc" : "success"
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.5.2 Query Device Properties

Function

A product model defines properties that the platform can deliver to devices. This API is used by an application to query real-time properties of a device. The device returns the query result to the application. This API can only be used for MQTT devices. NB-IoT devices are not supported.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices/{device_id}/properties

Table 1-138 Path Parameters

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: ID of the device to which the property is delivered. The ID is unique and is allocated by the platform during device registration. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-139 Query Parameters

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: device service ID, which is defined in the product model associated with the device.

Request Parameters

Table 1-140 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-141 Response body parameters

Parameter	Type	Description
request_id	String	An ID that uniquely identifies a device property query request. It is unique and is allocated by the platform when delivering a command for device property query.
response	Object	Property execution result reported by the device. The value is in JSON format. The specific format depends on the application and device.
error_code	String	Error code of a property query exception.
error_msg	String	Error message of a property query exception.

Example Requests

Queries all properties of the **Temperature** service of a specified device.

```
GET https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/properties?service_id=Temperature
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "response" : {
    "services" : {
```



```
    "serviceId" : "Temperature",
    "properties" : {
      "PhV_phsA" : "1",
      "PhV_phsB" : "2"
    },
    "eventTime" : "20190606T121212Z"
  }
 }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
503	Service Unavailable

Error Codes

See **Error Codes**.

1.4.6 AMQP Queue Management

1.4.6.1 Create an AMQP Queue

Function

This API is used by an application to create an AMQP queue on the IoT platform. Each tenant can create up to 100 queues. If the number of queues exceeds 100 or the name of a queue to create is the same as an existing queue name, the creation fails.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/amqp-queues

Table 1-142 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-143 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-144 Request body parameters

Parameter	Mandatory	Type	Description
queue_name	Yes	String	**Parameter description**: queue name, which must be unique for a tenant. **Value**: The value can contain 8 to 128 characters. Only letters, digits, underscores (_), hyphens (-), periods (.), and colons (:) are allowed. Minimum: **8** Maximum: **128**

Response Parameters

Status code: 201

Table 1-145 Response body parameters

Parameter	Type	Description
queue_id	String	Unique queue ID.
queue_name	String	Queue name, which must be unique for the same tenant. Minimum: **8** Maximum: **128**
create_time	String	Time when the queue was created on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
last_modify_time	String	Time when the queue was last modified on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Creates an AMQP queue named **myQueue**.

```
POST https://{endpoint}/v5/iot/{project_id}/amqp-queues

{
  "queue_name" : "myQueue"
}
```


Example Responses

Status code: 201

Created

```
{
  "queue_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "queue_name" : "myQueue0",
  "create_time" : "20190303T081011Z",
  "last_modify_time" : "20190303T081011Z"
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.6.2 Query the AMQP List

Function

This API is used by an application to query the Advanced Message Queuing Protocol (AMQP) list on the IoT platform. Fuzzy query can be performed based on the queue name. Results can be displayed in separate pages.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/amqp-queues

Table 1-146 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-147 Query Parameters

Parameter	Mandatory	Type	Description
queue_name	No	String	**Parameter description**: AMQP queue name. Fuzzy query is supported. If this parameter is left blank, all queues are queried. Currently, a maximum of 100 queues can be queried by a single user. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), hyphens (-), periods (.), and colons (:) are allowed. Maximum: **128**
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-148 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-149 Response body parameters

Parameter	Type	Description
queues	Array of **QueryQueueBase** objects	Queue list.
page	**Page** object	Pagination information of the query results.

Table 1-150 QueryQueueBase

Parameter	Type	Description
queue_id	String	Unique queue ID.
queue_name	String	Queue name, which must be unique for the same tenant. Minimum: **8** Maximum: **128**
create_time	String	Time when the queue was created on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
last_modify_time	String	Time when the queue was last modified on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-151 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries all AMQP queues.

```
GET https://{endpoint}/v5/iot/{project_id}/amqp-queues
```

Example Responses

Status code: 200

OK

```
{
  "queues" : [ {
    "queue_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "queue_name" : "myQueue0",
    "create_time" : "20190303T081011Z",
    "last_modify_time" : "20190303T081011Z"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.6.3 Query an AMQP Queue

Function

This API is used by an application to query the details of a specific queue on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/amqp-queues/{queue_id}

Table 1-152 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
queue_id	Yes	String	**Parameter description**: queue ID, which uniquely identifies a queue. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-153 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-154 Response body parameters

Parameter	Type	Description
queue_id	String	Unique queue ID.
queue_name	String	Queue name, which must be unique for the same tenant. Minimum: **8** Maximum: **128**
create_time	String	Time when the queue was created on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Type	Description
last_modify_time	String	Time when the queue was last modified on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Queries details about a specified queue.

```
GET https://{endpoint}/v5/iot/{project_id}/amqp-queues/{queue_id}
```

Example Responses

Status code: 200

OK

```
{
  "queue_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "queue_name" : "myQueue0",
  "create_time" : "20190303T081011Z",
  "last_modify_time" : "20190303T081011Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.6.4 Delete an AMQP Queue

Function

This API is used by an application to delete a specific AMQP queue from the IoT platform. If the queue is in use, the deletion fails.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/amqp-queues/{queue_id}

Table 1-155 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
queue_id	Yes	String	**Parameter description**: queue ID, which uniquely identifies a queue. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-156 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a specified AMQP queue.

```
DELETE https://{endpoint}/v5/iot/{project_id}/amqp-queues/{queue_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.7 Access Credential Management

Access credential APIs are used for authentication when long connections are established using protocols such as AMQP and MQTTS.

1.4.7.1 Generate an Access Credential

Function

This API is used by a client to generate an access credential, so the client can use the credential to connect to the platform through a protocol such as AMQP. Only one record is retained. If the API is called again, the access credential is reset and the previous one becomes invalid.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/auth/accesscode

Table 1-157 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-158 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Call the IAM API **Obtaining a User Token Through Password Authentication**. For details, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **0** Maximum: **36**

Table 1-159 Request body parameters

Parameter	Mandatory	Type	Description
type	No	String	**Parameter description**: access credential type. The default value is **AMQP**. **Options**: • AMQP Minimum: **0** Maximum: **16**
force_disconnect	No	Boolean	**Parameter description**: whether to disconnect the AMQP/MQTT connection. Default: **false**

Response Parameters

Status code: 201

Table 1-160 Response body parameters

Parameter	Type	Description
access_key	String	Access credential name, which consists of 8 random characters. Minimum: **0** Maximum: **8**

Parameter	Type	Description
access_code	String	Access credential. Minimum: **0** Maximum: **1024**

Example Requests

Generates an AMQP access credential.

```
POST https://{endpoint}/v5/iot/{project_id}/auth/accesscode

{
  "type" : "AMQP"
}
```

Example Responses

Status code: 201

Created

```
{
  "access_key" : "examples",
  "access_code" : "examples"
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8 Data Forwarding Rule Management

1.4.8.1 Create a Rule Triggering Condition

Function

This API is used by an application to create a rule triggering condition on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/routing-rule/rules

Table 1-161 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-162 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-163 Request body parameters

Parameter	Mandatory	Type	Description
rule_name	No	String	**Parameter description**: rule name. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	No	String	**Parameter description**: custom rule description. Minimum: **0** Maximum: **256**
subject	Yes	**RoutingRuleSubject** object	**Parameter description**: resource event details.

Parameter	Mandatory	Type	Description
app_type	No	String	**Parameter description**: application scope of the tenant rule. The default value is **GLOBAL**. **Options**: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space. If this parameter is set to **APP**, the rule to create takes effect in the resource space specified by **app_id** carried in the request. If **app_id** is not carried, the rule to create takes effect in the default **resource space**.
app_id	No	String	**Parameter description**: resource space ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
select	No	String	**Parameter description**: custom SQL **select** statement. The value can contain up to 2,500 characters. This parameter is used only by users of the Standard and Enterprise editions. For details about the available parameters, see the request parameters of data forwarding APIs in the help document, for example, **notify_data.body**. Minimum: **0** Maximum: **2500**

Parameter	Mandatory	Type	Description
where	No	String	**Parameter description**: user-defined SQL **where** statement. The maximum length is 2,500 characters. This parameter is used only by users of the standard and enterprise editions. For details about the available parameters, see the request parameters of data forwarding APIs in the help document, for example, **notify_data.body**. Minimum: **0** Maximum: **2500**

Table 1-164 RoutingRuleSubject

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: resource name. **Options**: • **device**: device. • **device.property**: device property. • **device.message**: device message. • **device.message.status**: device message status. • **device.status**: device status. • **batchtask**: batch task. • **product**: product. • **device.command.status**: asynchronous command status of the device. Minimum: **1** Maximum: **50**

Parameter	Mandatory	Type	Description
event	Yes	String	**Parameter description**: resource event. **Value**: The value range varies by resource type. **event** must be associated with **resource**. The mapping between **event** and **resource** is as follows: • **device** maps to **create** (device addition). • **device** maps to **delete** (device deletion). • **device** maps to **update** (device data change). • **device.status** maps to **update** (device status change). • **device.property** maps to **report** (device property reporting). • **device.message** maps to **report** (device message reporting). • **device.message.status** maps to **update** (device message status change). • **batchtask** maps to **update** (batch task status change). • **product** maps to **create** (product addition). • **product** maps to **delete** (product deletion). • **product** maps to **update** (product update). • **device.command.status** maps to **update** (update of the device asynchronous command status). Minimum: **1** Maximum: **50**

Response Parameters

Status code: 201

Table 1-165 Response body parameters

Parameter	Type	Description
rule_id	String	Unique ID of a rule triggering condition. The value is allocated by the platform during rule triggering condition creation.
rule_name	String	Custom rule name. Minimum: **1** Maximum: **256**
description	String	Custom rule description. Minimum: **1** Maximum: **256**
subject	**RoutingRuleSubject** object	Resource change event.
app_type	String	Application scope of the tenant rule. Options: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space.
app_id	String	Resource space ID.
select	String	Custom SQL **select** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
where	String	Custom SQL **where** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
active	Boolean	Whether the rule triggering condition is activated.

Table 1-166 RoutingRuleSubject

Parameter	Type	Description
resource	String	**Parameter description**: resource name. **Options**: • **device**: device. • **device.property**: device property. • **device.message**: device message. • **device.message.status**: device message status. • **device.status**: device status. • **batchtask**: batch task. • **product**: product. • **device.command.status**: asynchronous command status of the device. Minimum: **1** Maximum: **50**
event	String	**Parameter description**: resource event. **Value**: The value range varies by resource type. **event** must be associated with **resource**. The mapping between **event** and **resource** is as follows: • **device** maps to **create** (device addition). • **device** maps to **delete** (device deletion). • **device** maps to **update** (device data change). • **device.status** maps to **update** (device status change). • **device.property** maps to **report** (device property reporting). • **device.message** maps to **report** (device message reporting). • **device.message.status** maps to **update** (device message status change). • **batchtask** maps to **update** (batch task status change). • **product** maps to **create** (product addition). • **product** maps to **delete** (product deletion). • **product** maps to **update** (product update). • **device.command.status** maps to **update** (update of the device asynchronous command status). Minimum: **1** Maximum: **50**

Example Requests

- Creates a trigger condition (device creation notification) for the rule.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/rules

{
  "rule_name" : "rulename",
  "subject" : {
    "resource" : "device",
    "event" : "create"
  },
  "app_type" : "GLOBAL",
  "description" : "description"
}
```

- Creates a trigger condition (property reporting) for the rule.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/rules

{
  "rule_name" : "rulename",
  "subject" : {
    "resource" : "device.property",
    "event" : "report"
  },
  "app_type" : "GLOBAL",
  "description" : "description"
}
```

- Creates a trigger condition (message reporting) for the rule. (Topics are filtered based on SQL statements. The Basic edition does not support the SQL filtering capability.)

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/rules

{
  "rule_name" : "rulename",
  "subject" : {
    "resource" : "device.message",
    "event" : "report"
  },
  "app_type" : "GLOBAL",
  "description" : "description",
  "select" : "notify_data.header as header,notify_data.body as body,'12345678901234abcd' as id",
  "where" : "notify_data.body.topic = '$oc/devices/646c7579a5adc915f8966e8b_8514932826827763/user/testmsg'"
}
```

Example Responses

Status code: 201

Created

```
{
  "rule_id" : "5bcaddda-75bf-4623-8c8d-26175c41fcca",
  "rule_name" : "rulename",
  "description" : "description",
  "subject" : {
    "resource" : "device",
    "event" : "create"
  },
  "app_type" : "GLOBAL",
  "app_id" : "1a7ffc5cd89c44dd8265b1653d951ce0",
  "select" : "*",
  "where" : "product_id='d89c-44dd-8265-b1653d951ce0'",
```



```
  "active" : false
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.2 Query the Rule Triggering Condition List

Function

This API is used by an application to query the rule triggering condition list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/rules

Table 1-167 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-168 Query Parameters

Parameter	Mandatory	Type	Description
resource	No	String	**Parameter description**: subscribed resource name. **Options**: • **device**: device. • **device.property**: device property. • **device.message**: device message. • **device.message.status**: device message status. • **device.status**: device status. • **batchtask**: batch task. • **product**: product. • **device.command.status**: asynchronous command status of the device. Minimum: **1** Maximum: **50**

Parameter	Mandatory	Type	Description
event	No	String	**Parameter description**: subscribed event. **Value**: The value range varies by resource type. **event** must be associated with **resource**. The mapping between **event** and **resource** is as follows: • **device** maps to **create** (device addition). • **device** maps to **delete** (device deletion). • **device** maps to **update** (device data change). • **device.status** maps to **update** (device status change). • **device.property** maps to **report** (device property reporting). • **device.message** maps to **report** (device message reporting). • **device.message.status** maps to **update** (device message status change). • **batchtask** maps to **update** (batch task status change). • **product** maps to **create** (product addition). • **product** maps to **delete** (product deletion). • **product** maps to **update** (product update). • **device.command.status** maps to **update** (update of the device asynchronous command status). Minimum: **1** Maximum: **50**

Parameter	Mandatory	Type	Description
app_type	No	String	**Parameter description**: application scope of the tenant rule. **Options**: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space. If **app_id** is specified, rule actions in the specified resource space are queried. If **app_id** is not specified, rule actions in the **default resource space** are queried.
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If **app_id** is specified, the rule actions in the specified resource space are queried. If **app_id** is not specified, the rule actions in the **default resource space** are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
rule_name	No	String	**Parameter description**: custom rule name. Minimum: **1** Maximum: **256**
active	No	Boolean	**Parameter description**: whether the rule triggering condition is activated.

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. By default, 10 records are displayed on each page. A maximum of 50 records can be displayed on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. - To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-169 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-170 Response body parameters

Parameter	Type	Description
rules	Array of **RoutingRule** objects	List of rule triggering conditions.
count	Integer	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Table 1-171 RoutingRule

Parameter	Type	Description
rule_id	String	Unique ID of a rule triggering condition. The value is allocated by the platform during rule triggering condition creation.
rule_name	String	Custom rule name. Minimum: **1** Maximum: **256**
description	String	Custom rule description. Minimum: **1** Maximum: **256**
subject	**RoutingRuleSubject** object	Resource change event.
app_type	String	Application scope of the tenant rule. Options: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space.
app_id	String	Resource space ID.
select	String	Custom SQL **select** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
where	String	Custom SQL **where** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
active	Boolean	Whether the rule triggering condition is activated.

Table 1-172 RoutingRuleSubject

Parameter	Type	Description
resource	String	**Parameter description**: resource name. **Options**: • **device**: device. • **device.property**: device property. • **device.message**: device message. • **device.message.status**: device message status. • **device.status**: device status. • **batchtask**: batch task. • **product**: product. • **device.command.status**: asynchronous command status of the device. Minimum: **1** Maximum: **50**
event	String	**Parameter description**: resource event. **Value**: The value range varies by resource type. **event** must be associated with **resource**. The mapping between **event** and **resource** is as follows: • **device** maps to **create** (device addition). • **device** maps to **delete** (device deletion). • **device** maps to **update** (device data change). • **device.status** maps to **update** (device status change). • **device.property** maps to **report** (device property reporting). • **device.message** maps to **report** (device message reporting). • **device.message.status** maps to **update** (device message status change). • **batchtask** maps to **update** (batch task status change). • **product** maps to **create** (product addition). • **product** maps to **delete** (product deletion). • **product** maps to **update** (product update). • **device.command.status** maps to **update** (update of the device asynchronous command status). Minimum: **1** Maximum: **50**

Example Requests

Queries data forwarding rules in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/rules
```

Example Responses

Status code: 200

Successful response

```
{
  "rules" : [ {
    "rule_id" : "5bcaddda-75bf-4623-8c8d-26175c41fcca",
    "app_type" : "GLOBAL",
    "select" : "*",
    "rule_name" : "rulename",
    "subject" : {
      "resource" : "device",
      "event" : "create"
    },
    "description" : "description",
    "active" : true,
    "where" : "product_id='d89c-44dd-8265-b1653d951ce0'",
    "app_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0"
  } ],
  "count" : 10,
  "marker" : "5c90fa7d3c4e4405e8525079"
}
```

Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.3 Query a Rule Triggering Condition

Function

This API is used by an application to query the configuration of a specific rule triggering condition on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/rules/{rule_id}

Table 1-173 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
rule_id	Yes	String	**Parameter description**: ID of the rule triggering condition. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-174 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-175 Response body parameters

Parameter	Type	Description
rule_id	String	Unique ID of a rule triggering condition. The value is allocated by the platform during rule triggering condition creation.
rule_name	String	Custom rule name. Minimum: **1** Maximum: **256**
description	String	Custom rule description. Minimum: **1** Maximum: **256**
subject	**RoutingRuleSubject** object	Resource change event.
app_type	String	Application scope of the tenant rule. Options: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space.
app_id	String	Resource space ID.
select	String	Custom SQL **select** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**

Parameter	Type	Description
where	String	Custom SQL **where** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
active	Boolean	Whether the rule triggering condition is activated.

Table 1-176 RoutingRuleSubject

Parameter	Type	Description
resource	String	**Parameter description**: resource name. **Options**: • **device**: device. • **device.property**: device property. • **device.message**: device message. • **device.message.status**: device message status. • **device.status**: device status. • **batchtask**: batch task. • **product**: product. • **device.command.status**: asynchronous command status of the device. Minimum: **1** Maximum: **50**

Parameter	Type	Description
event	String	**Parameter description**: resource event. **Value**: The value range varies by resource type. **event** must be associated with **resource**. The mapping between **event** and **resource** is as follows: • **device** maps to **create** (device addition). • **device** maps to **delete** (device deletion). • **device** maps to **update** (device data change). • **device.status** maps to **update** (device status change). • **device.property** maps to **report** (device property reporting). • **device.message** maps to **report** (device message reporting). • **device.message.status** maps to **update** (device message status change). • **batchtask** maps to **update** (batch task status change). • **product** maps to **create** (product addition). • **product** maps to **delete** (product deletion). • **product** maps to **update** (product update). • **device.command.status** maps to **update** (update of the device asynchronous command status). Minimum: **1** Maximum: **50**

Example Requests

Queries details about a specified rule.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/rules/{rule_id}
```

Example Responses

Status code: 200

OK

```
{
  "rule_id" : "5bcaddda-75bf-4623-8c8d-26175c41fcca",
  "rule_name" : "rulename",
  "description" : "description",
  "subject" : {
    "resource" : "device",
    "event" : "create"
  },
```



```
  "app_type" : "GLOBAL",
  "app_id" : "1a7ffc5cd89c44dd8265b1653d951ce0",
  "select" : "*",
  "where" : "product_id='d89c-44dd-8265-b1653d951ce0'",
  "active" : false
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.4 Modify a Rule Triggering Condition

Function

This API is used by an application to modify the configuration of a specific rule triggering condition on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/routing-rule/rules/{rule_id}

Table 1-177 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
rule_id	Yes	String	**Parameter description**: ID of the rule triggering condition. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-178 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-179 Request body parameters

Parameter	Mandatory	Type	Description
rule_name	No	String	**Parameter description**: rule name. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	No	String	**Parameter description**: custom rule description. Minimum: **0** Maximum: **256**
select	No	String	**Parameter description**: custom SQL **select** statement. The value can contain up to 2500 characters. When updating a SQL statement, you must transfer the **select** and **where** parameters. If you need to clear the value of this parameter, set the parameter value to an empty string. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
where	No	String	**Parameter description**: custom SQL **where** statement. The value can contain up to 2500 characters. When updating a SQL statement, you must transfer the **select** and **where** parameters. If you need to clear the value of this parameter, set the parameter value to an empty string. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**

Parameter	Mandatory	Type	Description
active	No	Boolean	**Parameter description**: whether the rule triggering condition is activated.

Response Parameters

Status code: 200

Table 1-180 Response body parameters

Parameter	Type	Description
rule_id	String	Unique ID of a rule triggering condition. The value is allocated by the platform during rule triggering condition creation.
rule_name	String	Custom rule name. Minimum: **1** Maximum: **256**
description	String	Custom rule description. Minimum: **1** Maximum: **256**
subject	**RoutingRuleSubject** object	Resource change event.
app_type	String	Application scope of the tenant rule. Options: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space.
app_id	String	Resource space ID.
select	String	Custom SQL **select** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**
where	String	Custom SQL **where** statement. The value can contain up to 2500 characters. This parameter is used only by users of the Standard and Enterprise editions. Minimum: **0** Maximum: **2500**

Parameter	Type	Description
active	Boolean	Whether the rule triggering condition is activated.

Table 1-181 RoutingRuleSubject

Parameter	Type	Description
resource	String	**Parameter description**: resource name. **Options**: • **device**: device. • **device.property**: device property. • **device.message**: device message. • **device.message.status**: device message status. • **device.status**: device status. • **batchtask**: batch task. • **product**: product. • **device.command.status**: asynchronous command status of the device. Minimum: **1** Maximum: **50**

Parameter	Type	Description
event	String	**Parameter description**: resource event. **Value**: The value range varies by resource type. **event** must be associated with **resource**. The mapping between **event** and **resource** is as follows: • **device** maps to **create** (device addition). • **device** maps to **delete** (device deletion). • **device** maps to **update** (device data change). • **device.status** maps to **update** (device status change). • **device.property** maps to **report** (device property reporting). • **device.message** maps to **report** (device message reporting). • **device.message.status** maps to **update** (device message status change). • **batchtask** maps to **update** (batch task status change). • **product** maps to **create** (product addition). • **product** maps to **delete** (product deletion). • **product** maps to **update** (product update). • **device.command.status** maps to **update** (update of the device asynchronous command status). Minimum: **1** Maximum: **50**

Example Requests

Modifies the trigger condition for the rule of the **d89c-44dd-8265-b1653d951ce0** product.

```
PUT https://{endpoint}/v5/iot/{project_id}/routing-rule/rules/{rule_id}

{
  "rule_name" : "rulename",
  "description" : "description",
  "select" : "*",
  "where" : "product_id='d89c-44dd-8265-b1653d951ce0'",
  "active" : true
}
```

Example Responses

Status code: 200

OK


```
{
  "rule_id" : "5bcaddda-75bf-4623-8c8d-26175c41fcca",
  "rule_name" : "rulename",
  "description" : "description",
  "subject" : {
    "resource" : "device",
    "event" : "create"
  },
  "app_type" : "GLOBAL",
  "app_id" : "1a7ffc5cd89c44dd8265b1653d951ce0",
  "select" : "*",
  "where" : "product_id='d89c-44dd-8265-b1653d951ce0'",
  "active" : false
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.5 Delete a Rule Triggering Condition

Function

This API is used by an application to delete a specific rule triggering condition from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/routing-rule/rules/{rule_id}

Table 1-182 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
rule_id	Yes	String	**Parameter description**: ID of the rule triggering condition. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-183 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

None

Example Requests

Deletes the trigger condition of a specified rule.

```
DELETE https://{endpoint}/v5/iot/{project_id}/routing-rule/rules/{rule_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.6 Create a Rule Action

Function

This API is used by an application to create a rule action on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/routing-rule/actions

Table 1-184 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-185 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-186 Request body parameters

Parameter	Mandatory	Type	Description
rule_id	Yes	String	**Parameter description**: unique ID of a rule triggering condition. The value is allocated by the platform during rule creation. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
channel	Yes	String	**Parameter description**: type of the rule action. **Options**: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
channel_detail	Yes	**ChannelDetail** object	**Parameter description**: channel parameters.

Table 1-187 ChannelDetail

Parameter	Mandatory	Type	Description
http_forwarding	No	**HttpForwarding** object	**Parameter description**: HTTP server message forwarding. This parameter is mandatory when **channel** is set to **HTTP_FORWARDING**.
dis_forwarding	No	**DisForwarding** object	** Parameter description**: DIS service message forwarding. This parameter is mandatory when **channel** is set to **DIS_FORWARDING**.
obs_forwarding	No	**ObsForwarding** object	** Parameter description**: OBS service message forwarding. This parameter is mandatory when **channel** is set to **OBS_FORWARDING**.
amqp_forwarding	No	**AmqpForwarding** object	**Parameter description**: AMQP service message forwarding. This parameter is mandatory when **channel** is set to **AMQP_FORWARDING**.
dms_kafka_forwarding	No	**DmsKafkaForwarding** object	**Parameter description**: Kafka message forwarding. This parameter is mandatory when **channel** is set to **DMS_KAFKA_FORWARDING**.
mysql_forwarding	No	**MysqlForwarding** object	**Parameter description**: MySQL message forwarding. This parameter is mandatory when **channel** is set to **MYSQL_FORWARDING**.
influxdb_forwarding	No	**InfluxDBForwarding** object	**Parameter description**: InfluxDB forwarding configuration parameters. This parameter is mandatory when **channel** is set to **INFLUXDB_FORWARDING**.
functiongraph_forwarding	No	**FunctionGraphForwarding** object	**Parameter description**: FunctionGraph service message forwarding. This parameter is mandatory when **channel** is set to **FUNCTIONGRAPH_FORWARDING**.

Table 1-188 HttpForwarding

Parameter	Mandatory	Type	Description
url	Yes	String	**Parameter description**: IP address of the HTTP server for receiving data that meets the rule triggering condition. HTTP is an insecure non-encrypted data transmission mode. You are advised to use HTTPS, which is more secure. Minimum: **1** Maximum: **256**
cert_id	No	String	**Parameter description**: certificate ID. For details about how to obtain a certificate ID, see **step 3 in Loading the CA Certificate**. Minimum: **1** Maximum: **64**
cn_name	No	String	**Parameter description**: If **sni_enable** is set to **true**, this parameter is mandatory and the value is the domain name of the server certificate, for example, **domain:8443**. If **sni_enable** is set to **false**, this parameter is left blank by default. Minimum: **1** Maximum: **64**
sni_enable	No	Boolean	**Parameter description**: The HTTPS server and client must support this function. The default value is **false**. If this parameter is set to **true**, the HTTPS client needs to carry **cn_name** when initiating a request. The HTTPS server returns the corresponding certificate based on **cn_name**. If this parameter is set to **false**, this function is disabled.

Parameter	Mandatory	Type	Description
signature_enable	No	Boolean	**Parameter description**: whether to enable a signature. The token takes effect only when this parameter is set to **true**. You are advised to set this parameter to **true** to use the token signature to check whether messages are from the platform.
token	No	String	**Parameter description**: Token used to generate a signature. The client can use the token to generate a signature based on rules and compare the signature with that carried in the push message for security verification. **Value**: The value can contain 3 to 32 characters. Only letters and digits are allowed. For details, see [**Token-based IoT Platform Authentication for HTTP/HTTPS Subscription/Push**]. Minimum: **3** Maximum: **32**

Table 1-189 DisForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the DIS service is deployed. Minimum: **1** Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the DIS service belongs. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
stream_name	No	String	**Parameter description**: DIS stream name. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**
stream_id	No	String	**Parameter description**: DIS stream ID. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**

Table 1-190 ObsForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the OBS service is deployed. Minimum: **1** Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the OBS service belongs. Minimum: **1** Maximum: **256**
bucket_name	Yes	String	**Parameter description**: OBS bucket name. Minimum: **1** Maximum: **256**
location	No	String	**Parameter description**: region where the OBS bucket is deployed. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
file_path	No	String	**Parameter description**: custom directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/). The directory cannot start or end with a slash (/) or contain two or more consecutive slashes (/). **Value**: The value can contain a maximum of 256 characters. Only letters (a–z and A–Z), digits (0–9), underscores (_), hyphens (-), slashes (/), and braces ({}) are allowed. Braces can be used only for template parameters. **Template parameters**: • *{YYYY}*: year • *{MM}*: month • *{DD}*: day • *{HH}*: hour • *{appId}*: application ID • *{deviceId}*: device ID For example, if the custom directory structure is *{YYYY}/{MM}/{DD}/{HH}*, data is generated in the corresponding directory structure **2021>08>11>09** based on the current time during data forwarding.

Table 1-191 AmqpForwarding

Parameter	Mandatory	Type	Description
queue_name	Yes	String	**Parameter description**: AMQP queue for receiving data that meets the rule triggering condition. Minimum: **8** Maximum: **256**

Table 1-192 DmsKafkaForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the Kafka service is deployed. Minimum: **1** Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the Kafka service belongs. Minimum: **1** Maximum: **256**
addresses	Yes	Array of **SupportPrivateLinkNetAddress** objects	**Parameter description**: list of addresses for Kafka message forwarding.
topic	Yes	String	**Parameter description**: topic associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
username	No	String	**Parameter description**: username associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
password	No	String	**Parameter description**: password associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
mechanism	No	String	**Parameter description**: SASL authentication mechanism associated with the message to be forwarded to Kafka. **Options**: • **PAAS**: In this mode, data is not encrypted and transmitted in plaintext, which is insecure. You are advised to use a more secure data encryption mode. • **PLAIN**: SASL/PLAIN authentication. You must enter the username and password. It is a simple username and password verification mechanism. It is not recommended in the SASL_PLAINTEXT scenario. • **SCRAM-SHA-512**: SASL/SCRAM-SHA-512 authentication. You must enter the username and password. This mechanism uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.

Parameter	Mandatory	Type	Description
security_protocol	No	String	**Parameter description**: Kafka transmission security protocol. If this parameter is left blank, the default value **SASL_SSL** is used. This parameter does not take effect when **mechanism** is set to **PAAS** or left empty. **Options**: • **SASL_SSL**: Data is encrypted with SSL certificates for high-security transmission. • **SASL_PLAINTEXT**: Data is transmitted in plaintext with username and password authentication. It is recommended that **mechanism** is set to **SCRAM-SHA-512**.

Table 1-193 SupportPrivateLinkNetAddress

Parameter	Mandatory	Type	Description
ip	No	String	**Parameter description**: IP address of the service.
port	No	Integer	**Parameter description**: Port number of the service. Minimum: **0** Maximum: **65535**
domain	No	String	**Parameter description**: Domain name of the service. Minimum: **4** Maximum: **255**

Table 1-194 MysqlForwarding

Parameter	Mandatory	Type	Description
address	Yes	**NetAddress** object	List of addresses to which the message is forwarded to ROMA.

Parameter	Mandatory	Type	Description
db_name	Yes	String	**Parameter description**: name of the MySQL database to be connected. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
username	Yes	String	**Parameter description**: username of the MySQL database to be connected. Minimum: **1** Maximum: **256**
password	Yes	String	**Parameter description**: password of the MySQL database to be connected. Minimum: **1** Maximum: **256**
enable_ssl	No	Boolean	**Parameter description**: Whether the client uses SSL to connect to the server. The default value is **true**. If the value is **false**, the insecure non-encrypted data transmission mode is used. You are advised to use the SSL mode. Default: **true**
table_name	Yes	String	**Parameter description**: table name of the MySQL database. Minimum: **1** Maximum: **64**
column_mappings	Yes	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between MySQL databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-195 InfluxDBForwarding

Parameter	Mandatory	Type	Description
address	Yes	**NetAddress** object	**Parameter description**: addresses for InfluxDB message forwarding.
db_name	Yes	String	**Parameter description**: name of the InfluxDB database to be connected. The database will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
username	Yes	String	**Parameter description**: username of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
password	Yes	String	**Parameter description**: password of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
measurement	Yes	String	**Parameter description**: measurement of the InfluxDB database. The measurement will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
column_mappings	Yes	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between InfluxDB databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-196 NetAddress

Parameter	Mandatory	Type	Description
ip	No	String	**Parameter description**: IP address of the service.

Parameter	Mandatory	Type	Description
port	No	Integer	**Parameter description**: port number of the service. Minimum: **0** Maximum: **65535**
domain	No	String	**Parameter description**: domain name of the service. Minimum: **4** Maximum: **255**

Table 1-197 ColumnMapping

Parameter	Mandatory	Type	Description
column_name	Yes	String	**Parameter description**: database column name. Minimum: **1** Maximum: **256**
json_key	Yes	String	**Parameter description**: attribute name of the data to be forwarded. Minimum: **1** Maximum: **256**

Table 1-198 FunctionGraphForwarding

Parameter	Mandatory	Type	Description
func_urn	Yes	String	**Parameter description**: Function Uniform Resource Name (URN), which uniquely identifies a function. Minimum: **0** Maximum: **65535**
func_name	Yes	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**

Response Parameters

Status code: 201

Table 1-199 Response body parameters

Parameter	Type	Description
action_id	String	Unique rule ID, which is allocated by the platform during rule creation. You do not need to carry this parameter when creating a rule.
rule_id	String	ID of the rule triggering condition corresponding to the rule action.
app_id	String	Resource space ID.
channel	String	Rule action type. Options: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
channel_detail	**ChannelDetail** object	Channel configuration details.

Table 1-200 ChannelDetail

Parameter	Type	Description
http_forwarding	**HttpForwarding** object	**Parameter description**: HTTP server message forwarding. This parameter is mandatory when **channel** is set to **HTTP_FORWARDING**.
dis_forwarding	**DisForwarding** object	** Parameter description**: DIS service message forwarding. This parameter is mandatory when **channel** is set to **DIS_FORWARDING**.
obs_forwarding	**ObsForwarding** object	** Parameter description**: OBS service message forwarding. This parameter is mandatory when **channel** is set to **OBS_FORWARDING**.

Parameter	Type	Description
amqp_forwarding	**AmqpForwarding** object	**Parameter description**: AMQP service message forwarding. This parameter is mandatory when **channel** is set to **AMQP_FORWARDING**.
dms_kafka_forwarding	**DmsKafkaForwarding** object	**Parameter description**: Kafka message forwarding. This parameter is mandatory when **channel** is set to **DMS_KAFKA_FORWARDING**.
mysql_forwarding	**MysqlForwarding** object	**Parameter description**: MySQL message forwarding. This parameter is mandatory when **channel** is set to **MYSQL_FORWARDING**.
influxdb_forwarding	**InfluxDBForwarding** object	**Parameter description**: InfluxDB forwarding configuration parameters. This parameter is mandatory when **channel** is set to **INFLUXDB_FORWARDING**.
functiongraph_forwarding	**FunctionGraphForwarding** object	**Parameter description**: FunctionGraph service message forwarding. This parameter is mandatory when **channel** is set to **FUNCTIONGRAPH_FORWARDING**.

Table 1-201 HttpForwarding

Parameter	Type	Description
url	String	**Parameter description**: IP address of the HTTP server for receiving data that meets the rule triggering condition. HTTP is an insecure non-encrypted data transmission mode. You are advised to use HTTPS, which is more secure. Minimum: **1** Maximum: **256**
cert_id	String	**Parameter description**: certificate ID. For details about how to obtain a certificate ID, see **step 3 in Loading the CA Certificate**. Minimum: **1** Maximum: **64**
cn_name	String	**Parameter description**: If **sni_enable** is set to **true**, this parameter is mandatory and the value is the domain name of the server certificate, for example, **domain:8443**. If **sni_enable** is set to **false**, this parameter is left blank by default. Minimum: **1** Maximum: **64**

Parameter	Type	Description
sni_enable	Boolean	**Parameter description**: The HTTPS server and client must support this function. The default value is **false**. If this parameter is set to **true**, the HTTPS client needs to carry **cn_name** when initiating a request. The HTTPS server returns the corresponding certificate based on **cn_name**. If this parameter is set to **false**, this function is disabled.
signature_enable	Boolean	**Parameter description**: whether to enable a signature. The token takes effect only when this parameter is set to **true**. You are advised to set this parameter to **true** to use the token signature to check whether messages are from the platform.
token	String	**Parameter description**: Token used to generate a signature. The client can use the token to generate a signature based on rules and compare the signature with that carried in the push message for security verification. **Value**: The value can contain 3 to 32 characters. Only letters and digits are allowed. For details, see [**Token-based IoT Platform Authentication for HTTP/HTTPS Subscription/Push**]. Minimum: **3** Maximum: **32**

Table 1-202 DisForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the DIS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the DIS service belongs. Minimum: **1** Maximum: **256**

Parameter	Type	Description
stream_name	String	**Parameter description**: DIS stream name. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**
stream_id	String	**Parameter description**: DIS stream ID. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**

Table 1-203 ObsForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the OBS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the OBS service belongs. Minimum: **1** Maximum: **256**
bucket_name	String	**Parameter description**: OBS bucket name. Minimum: **1** Maximum: **256**
location	String	**Parameter description**: region where the OBS bucket is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
file_path	String	**Parameter description**: custom directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/). The directory cannot start or end with a slash (/) or contain two or more consecutive slashes (/). **Value**: The value can contain a maximum of 256 characters. Only letters (a–z and A–Z), digits (0–9), underscores (_), hyphens (-), slashes (/), and braces ({}) are allowed. Braces can be used only for template parameters. **Template parameters**: • *{YYYY}*: year • *{MM}*: month • *{DD}*: day • *{HH}*: hour • *{appId}*: application ID • *{deviceId}*: device ID For example, if the custom directory structure is *{YYYY}/{MM}/{DD}/{HH}*, data is generated in the corresponding directory structure **2021>08>11>09** based on the current time during data forwarding.

Table 1-204 AmqpForwarding

Parameter	Type	Description
queue_name	String	**Parameter description**: AMQP queue for receiving data that meets the rule triggering condition. Minimum: **8** Maximum: **256**

Table 1-205 DmsKafkaForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the Kafka service is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
project_id	String	**Parameter description**: ID of the project to which the Kafka service belongs. Minimum: **1** Maximum: **256**
addresses	Array of **SupportPrivateLinkNetAddress** objects	**Parameter description**: list of addresses for Kafka message forwarding.
topic	String	**Parameter description**: topic associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
username	String	**Parameter description**: username associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
mechanism	String	**Parameter description**: SASL authentication mechanism associated with the message to be forwarded to Kafka. **Options**: • **PAAS**: In this mode, data is not encrypted and transmitted in plaintext, which is insecure. You are advised to use a more secure data encryption mode. • **PLAIN**: SASL/PLAIN authentication. You must enter the username and password. It is a simple username and password verification mechanism. It is not recommended in the SASL_PLAINTEXT scenario. • **SCRAM-SHA-512**: SASL/SCRAM-SHA-512 authentication. You must enter the username and password. This mechanism uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.

Parameter	Type	Description
security_protocol	String	**Parameter description**: Kafka transmission security protocol. If this parameter is left blank, the default value **SASL_SSL** is used. This parameter does not take effect when **mechanism** is set to **PAAS** or left empty. **Options**: • **SASL_SSL**: Data is encrypted with SSL certificates for high-security transmission. • **SASL_PLAINTEXT**: Data is transmitted in plaintext with username and password authentication. It is recommended that **mechanism** is set to **SCRAM-SHA-512**.

Table 1-206 SupportPrivateLinkNetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: Port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: Domain name of the service. Minimum: **4** Maximum: **255**

Table 1-207 MysqlForwarding

Parameter	Type	Description
address	**NetAddress** object	List of addresses to which the message is forwarded to ROMA.
db_name	String	**Parameter description**: name of the MySQL database to be connected. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
username	String	**Parameter description**: username of the MySQL database to be connected. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password of the MySQL database to be connected. Minimum: **1** Maximum: **256**
enable_ssl	Boolean	**Parameter description**: Whether the client uses SSL to connect to the server. The default value is **true**. If the value is **false**, the insecure non-encrypted data transmission mode is used. You are advised to use the SSL mode. Default: **true**
table_name	String	**Parameter description**: table name of the MySQL database. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between MySQL databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-208 InfluxDBForwarding

Parameter	Type	Description
address	**NetAddress** object	**Parameter description**: addresses for InfluxDB message forwarding.
db_name	String	**Parameter description**: name of the InfluxDB database to be connected. The database will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
username	String	**Parameter description**: username of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**

Parameter	Type	Description
password	String	**Parameter description**: password of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
measurement	String	**Parameter description**: measurement of the InfluxDB database. The measurement will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between InfluxDB databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-209 NetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: domain name of the service. Minimum: **4** Maximum: **255**

Table 1-210 ColumnMapping

Parameter	Type	Description
column_name	String	**Parameter description**: database column name. Minimum: **1** Maximum: **256**

Parameter	Type	Description
json_key	String	**Parameter description**: attribute name of the data to be forwarded. Minimum: **1** Maximum: **256**

Table 1-211 FunctionGraphForwarding

Parameter	Type	Description
func_urn	String	**Parameter description**: Function Uniform Resource Name (URN), which uniquely identifies a function. Minimum: **0** Maximum: **65535**
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**

Example Requests

- Creates a rule action and pushes it to the HTTP server.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/actions

{
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "channel" : "HTTP_FORWARDING",
  "channel_detail" : {
    "http_forwarding" : {
      "url" : "http://host:port/callbackurltest"
    }
  }
}
```

- Creates a rule action and pushes it to OBS.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/actions

{
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "channel" : "OBS_FORWARDING",
  "channel_detail" : {
    "obs_forwarding" : {
      "file_path" : "yourPath",
      "project_id" : "yourProjectId",
      "bucket_name" : "yourBucket_name",
      "region_name" : "yourRegion"
    }
  }
}
```

- Creates a rule action and pushes it to an AMQP queue.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/actions
```



```
{
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "channel" : "AMQP_FORWARDING",
  "channel_detail" : {
    "amqp_forwarding" : {
      "queue_name" : "yourQueueName"
    }
  }
}
```

- Creates a rule action and pushes it to a MySQL database.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/actions

{
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "channel" : "MYSQL_FORWARDING",
  "channel_detail" : {
    "mysql_forwarding" : {
      "address" : {
        "ip" : "yourIp",
        "port" : 3306
      },
      "username" : "userName",
      "password" : "passworld",
      "db_name" : "yourDBName",
      "table_name" : "yourTableName",
      "enable_ssl" : true,
      "column_mappings" : [ {
        "column_name" : "serviceId",
        "json_key" : "notify_data.body.services[0].service_id"
      } ]
    }
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "action_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce1",
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "app_id" : "1a7ffc5cd89c44dd8265b1653d951ce0",
  "channel" : "HTTP_FORWARDING",
  "channel_detail" : {
    "http_forwarding" : {
      "url" : "http://host:port/callbackurltest"
    }
  }
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden

Status Code	Description
404	not found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.7 Query the Rule Action List

Function

This API is used by an application to query the rule action list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/actions

Table 1-212 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-213 Query Parameters

Parameter	Mandatory	Type	Description
rule_id	No	String	**Parameter description**: ID of the rule triggering condition. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
channel	No	String	**Parameter description**: type of the rule action. **Options**: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
app_type	No	String	**Parameter description**: application scope of the tenant rule. **Options**: • **GLOBAL**: The rule takes effect for all resources under the tenant. • **APP**: The rule takes effect for resources in a resource space. If **app_id** is specified, rule actions in the specified resource space are queried. If **app_id** is not specified, rule actions in the **default resource space** are queried.

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. This parameter is valid only when **rule_id** is not carried and **app_type** is set to **APP**. If **app_id** is specified, rule actions in the specified resource space are queried. If **app_id** is not specified, rule actions in the **default resource space** are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
limit	No	Integer	**Parameter description**: number of records to display on each page. By default, 10 records are displayed on each page. A maximum of 50 records can be displayed on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. - To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-214 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-215 Response body parameters

Parameter	Type	Description
actions	Array of **RoutingRuleAction** objects	Rule action list.
count	Integer	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Table 1-216 RoutingRuleAction

Parameter	Type	Description
action_id	String	Unique rule ID, which is allocated by the platform during rule creation. You do not need to carry this parameter when creating a rule.
rule_id	String	ID of the rule triggering condition corresponding to the rule action.
app_id	String	Resource space ID.
channel	String	Rule action type. Options: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
channel_detail	**ChannelDetail** object	Channel configuration details.

Table 1-217 ChannelDetail

Parameter	Type	Description
http_forwarding	**HttpForwarding** object	**Parameter description**: HTTP server message forwarding. This parameter is mandatory when **channel** is set to **HTTP_FORWARDING**.
dis_forwarding	**DisForwarding** object	** Parameter description**: DIS service message forwarding. This parameter is mandatory when **channel** is set to **DIS_FORWARDING**.
obs_forwarding	**ObsForwarding** object	** Parameter description**: OBS service message forwarding. This parameter is mandatory when **channel** is set to **OBS_FORWARDING**.

Parameter	Type	Description
amqp_forwarding	**AmqpForwarding** object	**Parameter description**: AMQP service message forwarding. This parameter is mandatory when **channel** is set to **AMQP_FORWARDING**.
dms_kafka_forwarding	**DmsKafkaForwarding** object	**Parameter description**: Kafka message forwarding. This parameter is mandatory when **channel** is set to **DMS_KAFKA_FORWARDING**.
mysql_forwarding	**MysqlForwarding** object	**Parameter description**: MySQL message forwarding. This parameter is mandatory when **channel** is set to **MYSQL_FORWARDING**.
influxdb_forwarding	**InfluxDBForwarding** object	**Parameter description**: InfluxDB forwarding configuration parameters. This parameter is mandatory when **channel** is set to **INFLUXDB_FORWARDING**.
functiongraph_forwarding	**FunctionGraphForwarding** object	**Parameter description**: FunctionGraph service message forwarding. This parameter is mandatory when **channel** is set to **FUNCTIONGRAPH_FORWARDING**.

Table 1-218 HttpForwarding

Parameter	Type	Description
url	String	**Parameter description**: IP address of the HTTP server for receiving data that meets the rule triggering condition. HTTP is an insecure non-encrypted data transmission mode. You are advised to use HTTPS, which is more secure. Minimum: **1** Maximum: **256**
cert_id	String	**Parameter description**: certificate ID. For details about how to obtain a certificate ID, see **step 3 in Loading the CA Certificate**. Minimum: **1** Maximum: **64**
cn_name	String	**Parameter description**: If **sni_enable** is set to **true**, this parameter is mandatory and the value is the domain name of the server certificate, for example, **domain:8443**. If **sni_enable** is set to **false**, this parameter is left blank by default. Minimum: **1** Maximum: **64**

Parameter	Type	Description
sni_enable	Boolean	**Parameter description**: The HTTPS server and client must support this function. The default value is **false**. If this parameter is set to **true**, the HTTPS client needs to carry **cn_name** when initiating a request. The HTTPS server returns the corresponding certificate based on **cn_name**. If this parameter is set to **false**, this function is disabled.
signature_enable	Boolean	**Parameter description**: whether to enable a signature. The token takes effect only when this parameter is set to **true**. You are advised to set this parameter to **true** to use the token signature to check whether messages are from the platform.
token	String	**Parameter description**: Token used to generate a signature. The client can use the token to generate a signature based on rules and compare the signature with that carried in the push message for security verification. **Value**: The value can contain 3 to 32 characters. Only letters and digits are allowed. For details, see [**Token-based IoT Platform Authentication for HTTP/HTTPS Subscription/Push**]. Minimum: **3** Maximum: **32**

Table 1-219 DisForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the DIS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the DIS service belongs. Minimum: **1** Maximum: **256**

Parameter	Type	Description
stream_name	String	**Parameter description**: DIS stream name. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**
stream_id	String	**Parameter description**: DIS stream ID. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**

Table 1-220 ObsForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the OBS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the OBS service belongs. Minimum: **1** Maximum: **256**
bucket_name	String	**Parameter description**: OBS bucket name. Minimum: **1** Maximum: **256**
location	String	**Parameter description**: region where the OBS bucket is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
file_path	String	**Parameter description**: custom directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/). The directory cannot start or end with a slash (/) or contain two or more consecutive slashes (/). **Value**: The value can contain a maximum of 256 characters. Only letters (a–z and A–Z), digits (0–9), underscores (_), hyphens (-), slashes (/), and braces ({}) are allowed. Braces can be used only for template parameters. **Template parameters**: • *{YYYY}*: year • *{MM}*: month • *{DD}*: day • *{HH}*: hour • *{appId}*: application ID • *{deviceId}*: device ID For example, if the custom directory structure is *{YYYY}/{MM}/{DD}/{HH}*, data is generated in the corresponding directory structure **2021>08>11>09** based on the current time during data forwarding.

Table 1-221 AmqpForwarding

Parameter	Type	Description
queue_name	String	**Parameter description**: AMQP queue for receiving data that meets the rule triggering condition. Minimum: **8** Maximum: **256**

Table 1-222 DmsKafkaForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the Kafka service is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
project_id	String	**Parameter description**: ID of the project to which the Kafka service belongs. Minimum: **1** Maximum: **256**
addresses	Array of **SupportPrivateLinkNetAddress** objects	**Parameter description**: list of addresses for Kafka message forwarding.
topic	String	**Parameter description**: topic associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
username	String	**Parameter description**: username associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
mechanism	String	**Parameter description**: SASL authentication mechanism associated with the message to be forwarded to Kafka. **Options**: • **PAAS**: In this mode, data is not encrypted and transmitted in plaintext, which is insecure. You are advised to use a more secure data encryption mode. • **PLAIN**: SASL/PLAIN authentication. You must enter the username and password. It is a simple username and password verification mechanism. It is not recommended in the SASL_PLAINTEXT scenario. • **SCRAM-SHA-512**: SASL/SCRAM-SHA-512 authentication. You must enter the username and password. This mechanism uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.

Parameter	Type	Description
security_protocol	String	**Parameter description**: Kafka transmission security protocol. If this parameter is left blank, the default value **SASL_SSL** is used. This parameter does not take effect when **mechanism** is set to **PAAS** or left empty. **Options**: • **SASL_SSL**: Data is encrypted with SSL certificates for high-security transmission. • **SASL_PLAINTEXT**: Data is transmitted in plaintext with username and password authentication. It is recommended that **mechanism** is set to **SCRAM-SHA-512**.

Table 1-223 SupportPrivateLinkNetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: Port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: Domain name of the service. Minimum: **4** Maximum: **255**

Table 1-224 MysqlForwarding

Parameter	Type	Description
address	**NetAddress** object	List of addresses to which the message is forwarded to ROMA.
db_name	String	**Parameter description**: name of the MySQL database to be connected. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
username	String	**Parameter description**: username of the MySQL database to be connected. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password of the MySQL database to be connected. Minimum: **1** Maximum: **256**
enable_ssl	Boolean	**Parameter description**: Whether the client uses SSL to connect to the server. The default value is **true**. If the value is **false**, the insecure non-encrypted data transmission mode is used. You are advised to use the SSL mode. Default: **true**
table_name	String	**Parameter description**: table name of the MySQL database. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between MySQL databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-225 InfluxDBForwarding

Parameter	Type	Description
address	**NetAddress** object	**Parameter description**: addresses for InfluxDB message forwarding.
db_name	String	**Parameter description**: name of the InfluxDB database to be connected. The database will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
username	String	**Parameter description**: username of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**

Parameter	Type	Description
password	String	**Parameter description**: password of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
measurement	String	**Parameter description**: measurement of the InfluxDB database. The measurement will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between InfluxDB databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-226 NetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: domain name of the service. Minimum: **4** Maximum: **255**

Table 1-227 ColumnMapping

Parameter	Type	Description
column_name	String	**Parameter description**: database column name. Minimum: **1** Maximum: **256**

Parameter	Type	Description
json_key	String	**Parameter description**: attribute name of the data to be forwarded. Minimum: **1** Maximum: **256**

Table 1-228 FunctionGraphForwarding

Parameter	Type	Description
func_urn	String	**Parameter description**: Function Uniform Resource Name (URN), which uniquely identifies a function. Minimum: **0** Maximum: **65535**
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**

Example Requests

Queries rule actions in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/actions
```

Example Responses

Status code: 200

Successful response

```
{
  "actions" : [ {
    "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce1",
    "action_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
    "channel_detail" : {
      "amqp_forwarding" : {
        "queue_name" : "test"
      },
      "obs_forwarding" : {
        "file_path" : "device_property_report/{YYYY}/{MM}/{DD}/{HH}",
        "project_id" : "project_id",
        "bucket_name" : "bucket_name",
        "region_name" : "region_name",
        "location" : "location"
      },
      "http_forwarding" : {
        "sni_enable" : false,
        "cn_name" : "domain:8443",
        "cert_id" : "0ae892cfeff641158920300b2292d2ca",
        "url" : "http://host:port/callbackurltest"
      },
      "dis_forwarding" : {
```



```
      "stream_name" : "stream_name",
      "project_id" : "project_id",
      "stream_id" : "stream_id",
      "region_name" : "region_name"
     },
     "dms_kafka_forwarding" : {
      "addresses" : [ {
       "port" : 443,
       "ip" : "host",
       "domain" : "huawei.com"
      } ],
      "password" : "password",
      "project_id" : "project_id",
      "topic" : "topic",
      "region_name" : "region_name",
      "mechanism" : "PLAIN",
      "security_protocol" : "SASL_SSL",
      "username" : "username"
     }
    },
    "channel" : "HTTP_FORWARDING",
    "app_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce2"
  } ],
  "count" : 10,
  "marker" : "5c90fa7d3c4e4405e8525079"
}
```


Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.8 Query a Rule Action

Function

This API is used by an application to query the configuration of a specific rule action on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/actions/{action_id}

Table 1-229 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
action_id	Yes	String	**Parameter description**: rule action ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-230 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-231 Response body parameters

Parameter	Type	Description
action_id	String	Unique rule ID, which is allocated by the platform during rule creation. You do not need to carry this parameter when creating a rule.
rule_id	String	ID of the rule triggering condition corresponding to the rule action.
app_id	String	Resource space ID.
channel	String	Rule action type. Options: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
channel_detail	**ChannelDetail** object	Channel configuration details.

Table 1-232 ChannelDetail

Parameter	Type	Description
http_forwarding	**HttpForwarding** object	**Parameter description**: HTTP server message forwarding. This parameter is mandatory when **channel** is set to **HTTP_FORWARDING**.
dis_forwarding	**DisForwarding** object	** Parameter description**: DIS service message forwarding. This parameter is mandatory when **channel** is set to **DIS_FORWARDING**.

Parameter	Type	Description
obs_forwarding	**ObsForwarding** object	** Parameter description**: OBS service message forwarding. This parameter is mandatory when **channel** is set to **OBS_FORWARDING**.
amqp_forwarding	**AmqpForwarding** object	**Parameter description**: AMQP service message forwarding. This parameter is mandatory when **channel** is set to **AMQP_FORWARDING**.
dms_kafka_forwarding	**DmsKafkaForwarding** object	**Parameter description**: Kafka message forwarding. This parameter is mandatory when **channel** is set to **DMS_KAFKA_FORWARDING**.
mysql_forwarding	**MysqlForwarding** object	**Parameter description**: MySQL message forwarding. This parameter is mandatory when **channel** is set to **MYSQL_FORWARDING**.
influxdb_forwarding	**InfluxDBForwarding** object	**Parameter description**: InfluxDB forwarding configuration parameters. This parameter is mandatory when **channel** is set to **INFLUXDB_FORWARDING**.
functiongraph_forwarding	**FunctionGraphForwarding** object	**Parameter description**: FunctionGraph service message forwarding. This parameter is mandatory when **channel** is set to **FUNCTIONGRAPH_FORWARDING**.

Table 1-233 HttpForwarding

Parameter	Type	Description
url	String	**Parameter description**: IP address of the HTTP server for receiving data that meets the rule triggering condition. HTTP is an insecure non-encrypted data transmission mode. You are advised to use HTTPS, which is more secure. Minimum: **1** Maximum: **256**
cert_id	String	**Parameter description**: certificate ID. For details about how to obtain a certificate ID, see **step 3 in Loading the CA Certificate**. Minimum: **1** Maximum: **64**

Parameter	Type	Description
cn_name	String	**Parameter description**: If **sni_enable** is set to **true**, this parameter is mandatory and the value is the domain name of the server certificate, for example, **domain:8443**. If **sni_enable** is set to **false**, this parameter is left blank by default. Minimum: **1** Maximum: **64**
sni_enable	Boolean	**Parameter description**: The HTTPS server and client must support this function. The default value is **false**. If this parameter is set to **true**, the HTTPS client needs to carry **cn_name** when initiating a request. The HTTPS server returns the corresponding certificate based on **cn_name**. If this parameter is set to **false**, this function is disabled.
signature_enable	Boolean	**Parameter description**: whether to enable a signature. The token takes effect only when this parameter is set to **true**. You are advised to set this parameter to **true** to use the token signature to check whether messages are from the platform.
token	String	**Parameter description**: Token used to generate a signature. The client can use the token to generate a signature based on rules and compare the signature with that carried in the push message for security verification. **Value**: The value can contain 3 to 32 characters. Only letters and digits are allowed. For details, see [**Token-based IoT Platform Authentication for HTTP/HTTPS Subscription/Push**]. Minimum: **3** Maximum: **32**

Table 1-234 DisForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the DIS service is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
project_id	String	**Parameter description**: ID of the project to which the DIS service belongs. Minimum: **1** Maximum: **256**
stream_name	String	**Parameter description**: DIS stream name. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**
stream_id	String	**Parameter description**: DIS stream ID. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**

Table 1-235 ObsForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the OBS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the OBS service belongs. Minimum: **1** Maximum: **256**
bucket_name	String	**Parameter description**: OBS bucket name. Minimum: **1** Maximum: **256**
location	String	**Parameter description**: region where the OBS bucket is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
file_path	String	**Parameter description**: custom directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/). The directory cannot start or end with a slash (/) or contain two or more consecutive slashes (/). **Value**: The value can contain a maximum of 256 characters. Only letters (a–z and A–Z), digits (0–9), underscores (_), hyphens (-), slashes (/), and braces ({}) are allowed. Braces can be used only for template parameters. **Template parameters**: • *{YYYY}*: year • *{MM}*: month • *{DD}*: day • *{HH}*: hour • *{appId}*: application ID • *{deviceId}*: device ID For example, if the custom directory structure is *{YYYY}/{MM}/{DD}/{HH}*, data is generated in the corresponding directory structure **2021>08>11>09** based on the current time during data forwarding.

Table 1-236 AmqpForwarding

Parameter	Type	Description
queue_name	String	**Parameter description**: AMQP queue for receiving data that meets the rule triggering condition. Minimum: **8** Maximum: **256**

Table 1-237 DmsKafkaForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the Kafka service is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
project_id	String	**Parameter description**: ID of the project to which the Kafka service belongs. Minimum: **1** Maximum: **256**
addresses	Array of **SupportPrivateLinkNetAddress** objects	**Parameter description**: list of addresses for Kafka message forwarding.
topic	String	**Parameter description**: topic associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
username	String	**Parameter description**: username associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
mechanism	String	**Parameter description**: SASL authentication mechanism associated with the message to be forwarded to Kafka. **Options**: • **PAAS**: In this mode, data is not encrypted and transmitted in plaintext, which is insecure. You are advised to use a more secure data encryption mode. • **PLAIN**: SASL/PLAIN authentication. You must enter the username and password. It is a simple username and password verification mechanism. It is not recommended in the SASL_PLAINTEXT scenario. • **SCRAM-SHA-512**: SASL/SCRAM-SHA-512 authentication. You must enter the username and password. This mechanism uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.

Parameter	Type	Description
security_protocol	String	**Parameter description**: Kafka transmission security protocol. If this parameter is left blank, the default value **SASL_SSL** is used. This parameter does not take effect when **mechanism** is set to **PAAS** or left empty. **Options**: • **SASL_SSL**: Data is encrypted with SSL certificates for high-security transmission. • **SASL_PLAINTEXT**: Data is transmitted in plaintext with username and password authentication. It is recommended that **mechanism** is set to **SCRAM-SHA-512**.

Table 1-238 SupportPrivateLinkNetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: Port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: Domain name of the service. Minimum: **4** Maximum: **255**

Table 1-239 MysqlForwarding

Parameter	Type	Description
address	**NetAddress** object	List of addresses to which the message is forwarded to ROMA.
db_name	String	**Parameter description**: name of the MySQL database to be connected. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
username	String	**Parameter description**: username of the MySQL database to be connected. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password of the MySQL database to be connected. Minimum: **1** Maximum: **256**
enable_ssl	Boolean	**Parameter description**: Whether the client uses SSL to connect to the server. The default value is **true**. If the value is **false**, the insecure non-encrypted data transmission mode is used. You are advised to use the SSL mode. Default: **true**
table_name	String	**Parameter description**: table name of the MySQL database. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between MySQL databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-240 InfluxDBForwarding

Parameter	Type	Description
address	**NetAddress** object	**Parameter description**: addresses for InfluxDB message forwarding.
db_name	String	**Parameter description**: name of the InfluxDB database to be connected. The database will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
username	String	**Parameter description**: username of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**

Parameter	Type	Description
password	String	**Parameter description**: password of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
measurement	String	**Parameter description**: measurement of the InfluxDB database. The measurement will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between InfluxDB databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-241 NetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: domain name of the service. Minimum: **4** Maximum: **255**

Table 1-242 ColumnMapping

Parameter	Type	Description
column_name	String	**Parameter description**: database column name. Minimum: **1** Maximum: **256**

Parameter	Type	Description
json_key	String	**Parameter description**: attribute name of the data to be forwarded. Minimum: **1** Maximum: **256**

Table 1-243 FunctionGraphForwarding

Parameter	Type	Description
func_urn	String	**Parameter description**: Function Uniform Resource Name (URN), which uniquely identifies a function. Minimum: **0** Maximum: **65535**
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**

Example Requests

Queries details about a specified rule action.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/actions/{action_id}
```

Example Responses

Status code: 200

OK

```
{
  "action_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce1",
  "app_id" : "1a7ffc5cd89c44dd8265b1653d951ce0",
  "channel" : "HTTP_FORWARDING",
  "channel_detail" : {
    "http_forwarding" : {
      "url" : "http://host:port/callbackurltest"
    }
  }
}
```

Status Codes

Status Code	Description
200	OK

Status Code	Description
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.9 Modify a Rule Action

Function

This API is used by an application to modify the configuration of a specific rule action on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/routing-rule/actions/{action_id}

Table 1-244 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
action_id	Yes	String	**Parameter description**: rule action ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-245 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-246 Request body parameters

Parameter	Mandatory	Type	Description
channel	No	String	**Parameter description**: type of the rule action. **Options**: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
channel_detail	No	**ChannelDetail** object	**Parameter description**: channel configuration details.

Table 1-247 ChannelDetail

Parameter	Mandatory	Type	Description
http_forwarding	No	**HttpForwarding** object	**Parameter description**: HTTP server message forwarding. This parameter is mandatory when **channel** is set to **HTTP_FORWARDING**.
dis_forwarding	No	**DisForwarding** object	** Parameter description**: DIS service message forwarding. This parameter is mandatory when **channel** is set to **DIS_FORWARDING**.

Parameter	Mandatory	Type	Description
obs_forwarding	No	**ObsForwarding** object	** Parameter description**: OBS service message forwarding. This parameter is mandatory when **channel** is set to **OBS_FORWARDING**.
amqp_forwarding	No	**AmqpForwarding** object	**Parameter description**: AMQP service message forwarding. This parameter is mandatory when **channel** is set to **AMQP_FORWARDING**.
dms_kafka_forwarding	No	**DmsKafkaForwarding** object	**Parameter description**: Kafka message forwarding. This parameter is mandatory when **channel** is set to **DMS_KAFKA_FORWARDING**.
mysql_forwarding	No	**MysqlForwarding** object	**Parameter description**: MySQL message forwarding. This parameter is mandatory when **channel** is set to **MYSQL_FORWARDING**.
influxdb_forwarding	No	**InfluxDBForwarding** object	**Parameter description**: InfluxDB forwarding configuration parameters. This parameter is mandatory when **channel** is set to **INFLUXDB_FORWARDING**.
functiongraph_forwarding	No	**FunctionGraphForwarding** object	**Parameter description**: FunctionGraph service message forwarding. This parameter is mandatory when **channel** is set to **FUNCTIONGRAPH_FORWARDING**.

Table 1-248 HttpForwarding

Parameter	Mandatory	Type	Description
url	Yes	String	**Parameter description**: IP address of the HTTP server for receiving data that meets the rule triggering condition. HTTP is an insecure non-encrypted data transmission mode. You are advised to use HTTPS, which is more secure. Minimum: **1** Maximum: **256**
cert_id	No	String	**Parameter description**: certificate ID. For details about how to obtain a certificate ID, see **step 3 in Loading the CA Certificate**. Minimum: **1** Maximum: **64**
cn_name	No	String	**Parameter description**: If **sni_enable** is set to **true**, this parameter is mandatory and the value is the domain name of the server certificate, for example, **domain:8443**. If **sni_enable** is set to **false**, this parameter is left blank by default. Minimum: **1** Maximum: **64**
sni_enable	No	Boolean	**Parameter description**: The HTTPS server and client must support this function. The default value is **false**. If this parameter is set to **true**, the HTTPS client needs to carry **cn_name** when initiating a request. The HTTPS server returns the corresponding certificate based on **cn_name**. If this parameter is set to **false**, this function is disabled.

Parameter	Mandatory	Type	Description
signature_enable	No	Boolean	**Parameter description**: whether to enable a signature. The token takes effect only when this parameter is set to **true**. You are advised to set this parameter to **true** to use the token signature to check whether messages are from the platform.
token	No	String	**Parameter description**: Token used to generate a signature. The client can use the token to generate a signature based on rules and compare the signature with that carried in the push message for security verification. **Value**: The value can contain 3 to 32 characters. Only letters and digits are allowed. For details, see [**Token-based IoT Platform Authentication for HTTP/HTTPS Subscription/Push**]. Minimum: **3** Maximum: **32**

Table 1-249 DisForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the DIS service is deployed. Minimum: **1** Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the DIS service belongs. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
stream_name	No	String	**Parameter description**: DIS stream name. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**
stream_id	No	String	**Parameter description**: DIS stream ID. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**

Table 1-250 ObsForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the OBS service is deployed. Minimum: **1** Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the OBS service belongs. Minimum: **1** Maximum: **256**
bucket_name	Yes	String	**Parameter description**: OBS bucket name. Minimum: **1** Maximum: **256**
location	No	String	**Parameter description**: region where the OBS bucket is deployed. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
file_path	No	String	**Parameter description**: custom directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/). The directory cannot start or end with a slash (/) or contain two or more consecutive slashes (/). **Value**: The value can contain a maximum of 256 characters. Only letters (a–z and A–Z), digits (0–9), underscores (_), hyphens (-), slashes (/), and braces ({}) are allowed. Braces can be used only for template parameters. **Template parameters**: • *{YYYY}*: year • *{MM}*: month • *{DD}*: day • *{HH}*: hour • *{appId}*: application ID • *{deviceId}*: device ID For example, if the custom directory structure is *{YYYY}/{MM}/{DD}/{HH}*, data is generated in the corresponding directory structure **2021>08>11>09** based on the current time during data forwarding.

Table 1-251 AmqpForwarding

Parameter	Mandatory	Type	Description
queue_name	Yes	String	**Parameter description**: AMQP queue for receiving data that meets the rule triggering condition. Minimum: **8** Maximum: **256**

Table 1-252 DmsKafkaForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the Kafka service is deployed. Minimum: **1** Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the Kafka service belongs. Minimum: **1** Maximum: **256**
addresses	Yes	Array of **SupportPrivateLinkNetAddress** objects	**Parameter description**: list of addresses for Kafka message forwarding.
topic	Yes	String	**Parameter description**: topic associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
username	No	String	**Parameter description**: username associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
password	No	String	**Parameter description**: password associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
mechanism	No	String	**Parameter description**: SASL authentication mechanism associated with the message to be forwarded to Kafka. **Options**: • **PAAS**: In this mode, data is not encrypted and transmitted in plaintext, which is insecure. You are advised to use a more secure data encryption mode. • **PLAIN**: SASL/PLAIN authentication. You must enter the username and password. It is a simple username and password verification mechanism. It is not recommended in the SASL_PLAINTEXT scenario. • **SCRAM-SHA-512**: SASL/SCRAM-SHA-512 authentication. You must enter the username and password. This mechanism uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.

Parameter	Mandatory	Type	Description
security_protocol	No	String	**Parameter description**: Kafka transmission security protocol. If this parameter is left blank, the default value **SASL_SSL** is used. This parameter does not take effect when **mechanism** is set to **PAAS** or left empty. **Options**: • **SASL_SSL**: Data is encrypted with SSL certificates for high-security transmission. • **SASL_PLAINTEXT**: Data is transmitted in plaintext with username and password authentication. It is recommended that **mechanism** is set to **SCRAM-SHA-512**.

Table 1-253 SupportPrivateLinkNetAddress

Parameter	Mandatory	Type	Description
ip	No	String	**Parameter description**: IP address of the service.
port	No	Integer	**Parameter description**: Port number of the service. Minimum: **0** Maximum: **65535**
domain	No	String	**Parameter description**: Domain name of the service. Minimum: **4** Maximum: **255**

Table 1-254 MysqlForwarding

Parameter	Mandatory	Type	Description
address	Yes	**NetAddress** object	List of addresses to which the message is forwarded to ROMA.

Parameter	Mandatory	Type	Description
db_name	Yes	String	**Parameter description**: name of the MySQL database to be connected. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
username	Yes	String	**Parameter description**: username of the MySQL database to be connected. Minimum: **1** Maximum: **256**
password	Yes	String	**Parameter description**: password of the MySQL database to be connected. Minimum: **1** Maximum: **256**
enable_ssl	No	Boolean	**Parameter description**: Whether the client uses SSL to connect to the server. The default value is **true**. If the value is **false**, the insecure non-encrypted data transmission mode is used. You are advised to use the SSL mode. Default: **true**
table_name	Yes	String	**Parameter description**: table name of the MySQL database. Minimum: **1** Maximum: **64**
column_mappings	Yes	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between MySQL databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-255 InfluxDBForwarding

Parameter	Mandatory	Type	Description
address	Yes	**NetAddress** object	**Parameter description**: addresses for InfluxDB message forwarding.
db_name	Yes	String	**Parameter description**: name of the InfluxDB database to be connected. The database will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
username	Yes	String	**Parameter description**: username of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
password	Yes	String	**Parameter description**: password of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
measurement	Yes	String	**Parameter description**: measurement of the InfluxDB database. The measurement will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
column_mappings	Yes	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between InfluxDB databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-256 NetAddress

Parameter	Mandatory	Type	Description
ip	No	String	**Parameter description**: IP address of the service.

Parameter	Mandatory	Type	Description
port	No	Integer	**Parameter description**: port number of the service. Minimum: **0** Maximum: **65535**
domain	No	String	**Parameter description**: domain name of the service. Minimum: **4** Maximum: **255**

Table 1-257 ColumnMapping

Parameter	Mandatory	Type	Description
column_name	Yes	String	**Parameter description**: database column name. Minimum: **1** Maximum: **256**
json_key	Yes	String	**Parameter description**: attribute name of the data to be forwarded. Minimum: **1** Maximum: **256**

Table 1-258 FunctionGraphForwarding

Parameter	Mandatory	Type	Description
func_urn	Yes	String	**Parameter description**: Function Uniform Resource Name (URN), which uniquely identifies a function. Minimum: **0** Maximum: **65535**
func_name	Yes	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**

Response Parameters

Status code: 200

Table 1-259 Response body parameters

Parameter	Type	Description
action_id	String	Unique rule ID, which is allocated by the platform during rule creation. You do not need to carry this parameter when creating a rule.
rule_id	String	ID of the rule triggering condition corresponding to the rule action.
app_id	String	Resource space ID.
channel	String	Rule action type. Options: • **HTTP_FORWARDING**: HTTP service message forwarding. • **DIS_FORWARDING**: DIS service message forwarding. • **OBS_FORWARDING**: OBS service message forwarding. • **AMQP_FORWARDING**: AMQP service message forwarding. • **DMS_KAFKA_FORWARDING**: Kafka message forwarding. • **INFLUXDB_FORWARDING**: InfluxDB message forwarding. • **MYSQL_FORWARDING**: MySQL message forwarding. • **FUNCTIONGRAPH_FORWARDING**: FunctionGraph message forwarding.
channel_detail	**ChannelDetail** object	Channel configuration details.

Table 1-260 ChannelDetail

Parameter	Type	Description
http_forwarding	**HttpForwarding** object	**Parameter description**: HTTP server message forwarding. This parameter is mandatory when **channel** is set to **HTTP_FORWARDING**.
dis_forwarding	**DisForwarding** object	** Parameter description**: DIS service message forwarding. This parameter is mandatory when **channel** is set to **DIS_FORWARDING**.
obs_forwarding	**ObsForwarding** object	** Parameter description**: OBS service message forwarding. This parameter is mandatory when **channel** is set to **OBS_FORWARDING**.

Parameter	Type	Description
amqp_forwarding	**AmqpForwarding** object	**Parameter description**: AMQP service message forwarding. This parameter is mandatory when **channel** is set to **AMQP_FORWARDING**.
dms_kafka_forwarding	**DmsKafkaForwarding** object	**Parameter description**: Kafka message forwarding. This parameter is mandatory when **channel** is set to **DMS_KAFKA_FORWARDING**.
mysql_forwarding	**MysqlForwarding** object	**Parameter description**: MySQL message forwarding. This parameter is mandatory when **channel** is set to **MYSQL_FORWARDING**.
influxdb_forwarding	**InfluxDBForwarding** object	**Parameter description**: InfluxDB forwarding configuration parameters. This parameter is mandatory when **channel** is set to **INFLUXDB_FORWARDING**.
functiongraph_forwarding	**FunctionGraphForwarding** object	**Parameter description**: FunctionGraph service message forwarding. This parameter is mandatory when **channel** is set to **FUNCTIONGRAPH_FORWARDING**.

Table 1-261 HttpForwarding

Parameter	Type	Description
url	String	**Parameter description**: IP address of the HTTP server for receiving data that meets the rule triggering condition. HTTP is an insecure non-encrypted data transmission mode. You are advised to use HTTPS, which is more secure. Minimum: **1** Maximum: **256**
cert_id	String	**Parameter description**: certificate ID. For details about how to obtain a certificate ID, see **step 3 in Loading the CA Certificate**. Minimum: **1** Maximum: **64**
cn_name	String	**Parameter description**: If **sni_enable** is set to **true**, this parameter is mandatory and the value is the domain name of the server certificate, for example, **domain:8443**. If **sni_enable** is set to **false**, this parameter is left blank by default. Minimum: **1** Maximum: **64**

Parameter	Type	Description
sni_enable	Boolean	**Parameter description**: The HTTPS server and client must support this function. The default value is **false**. If this parameter is set to **true**, the HTTPS client needs to carry **cn_name** when initiating a request. The HTTPS server returns the corresponding certificate based on **cn_name**. If this parameter is set to **false**, this function is disabled.
signature_enable	Boolean	**Parameter description**: whether to enable a signature. The token takes effect only when this parameter is set to **true**. You are advised to set this parameter to **true** to use the token signature to check whether messages are from the platform.
token	String	**Parameter description**: Token used to generate a signature. The client can use the token to generate a signature based on rules and compare the signature with that carried in the push message for security verification. **Value**: The value can contain 3 to 32 characters. Only letters and digits are allowed. For details, see [**Token-based IoT Platform Authentication for HTTP/HTTPS Subscription/Push**]. Minimum: **3** Maximum: **32**

Table 1-262 DisForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the DIS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the DIS service belongs. Minimum: **1** Maximum: **256**

Parameter	Type	Description
stream_name	String	**Parameter description**: DIS stream name. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**
stream_id	String	**Parameter description**: DIS stream ID. Either **stream_id** or **stream_name** must be carried. If both parameters are carried, **stream_id** is used. Minimum: **1** Maximum: **256**

Table 1-263 ObsForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the OBS service is deployed. Minimum: **1** Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the OBS service belongs. Minimum: **1** Maximum: **256**
bucket_name	String	**Parameter description**: OBS bucket name. Minimum: **1** Maximum: **256**
location	String	**Parameter description**: region where the OBS bucket is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
file_path	String	**Parameter description**: custom directory to store files that will be dumped to OBS. Different directory levels are separated by slashes (/). The directory cannot start or end with a slash (/) or contain two or more consecutive slashes (/). **Value**: The value can contain a maximum of 256 characters. Only letters (a–z and A–Z), digits (0–9), underscores (_), hyphens (-), slashes (/), and braces ({}) are allowed. Braces can be used only for template parameters. **Template parameters**: • *{YYYY}*: year • *{MM}*: month • *{DD}*: day • *{HH}*: hour • *{appId}*: application ID • *{deviceId}*: device ID For example, if the custom directory structure is *{YYYY}/{MM}/{DD}/{HH}*, data is generated in the corresponding directory structure **2021>08>11>09** based on the current time during data forwarding.

Table 1-264 AmqpForwarding

Parameter	Type	Description
queue_name	String	**Parameter description**: AMQP queue for receiving data that meets the rule triggering condition. Minimum: **8** Maximum: **256**

Table 1-265 DmsKafkaForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the Kafka service is deployed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
project_id	String	**Parameter description**: ID of the project to which the Kafka service belongs. Minimum: **1** Maximum: **256**
addresses	Array of **SupportPrivateLinkNetAddress** objects	**Parameter description**: list of addresses for Kafka message forwarding.
topic	String	**Parameter description**: topic associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
username	String	**Parameter description**: username associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password associated with the message to be forwarded to Kafka. Minimum: **1** Maximum: **256**
mechanism	String	**Parameter description**: SASL authentication mechanism associated with the message to be forwarded to Kafka. **Options**: • **PAAS**: In this mode, data is not encrypted and transmitted in plaintext, which is insecure. You are advised to use a more secure data encryption mode. • **PLAIN**: SASL/PLAIN authentication. You must enter the username and password. It is a simple username and password verification mechanism. It is not recommended in the SASL_PLAINTEXT scenario. • **SCRAM-SHA-512**: SASL/SCRAM-SHA-512 authentication. You must enter the username and password. This mechanism uses the hash algorithm to generate credentials for usernames and passwords to verify identities. SCRAM-SHA-512 is more secure than PLAIN.

Parameter	Type	Description
security_protocol	String	**Parameter description**: Kafka transmission security protocol. If this parameter is left blank, the default value **SASL_SSL** is used. This parameter does not take effect when **mechanism** is set to **PAAS** or left empty. **Options**: • **SASL_SSL**: Data is encrypted with SSL certificates for high-security transmission. • **SASL_PLAINTEXT**: Data is transmitted in plaintext with username and password authentication. It is recommended that **mechanism** is set to **SCRAM-SHA-512**.

Table 1-266 SupportPrivateLinkNetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: Port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: Domain name of the service. Minimum: **4** Maximum: **255**

Table 1-267 MysqlForwarding

Parameter	Type	Description
address	**NetAddress** object	List of addresses to which the message is forwarded to ROMA.
db_name	String	**Parameter description**: name of the MySQL database to be connected. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
username	String	**Parameter description**: username of the MySQL database to be connected. Minimum: **1** Maximum: **256**
password	String	**Parameter description**: password of the MySQL database to be connected. Minimum: **1** Maximum: **256**
enable_ssl	Boolean	**Parameter description**: Whether the client uses SSL to connect to the server. The default value is **true**. If the value is **false**, the insecure non-encrypted data transmission mode is used. You are advised to use the SSL mode. Default: **true**
table_name	String	**Parameter description**: table name of the MySQL database. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between MySQL databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-268 InfluxDBForwarding

Parameter	Type	Description
address	**NetAddress** object	**Parameter description**: addresses for InfluxDB message forwarding.
db_name	String	**Parameter description**: name of the InfluxDB database to be connected. The database will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
username	String	**Parameter description**: username of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**

Parameter	Type	Description
password	String	**Parameter description**: password of the InfluxDB database to be connected. Minimum: **1** Maximum: **256**
measurement	String	**Parameter description**: measurement of the InfluxDB database. The measurement will be automatically created if it does not exist. Minimum: **1** Maximum: **64**
column_mappings	Array of **ColumnMapping** objects	** Parameter description**: list of mappings between InfluxDB databases and the forwarding data. A maximum of 32 mappings are supported.

Table 1-269 NetAddress

Parameter	Type	Description
ip	String	**Parameter description**: IP address of the service.
port	Integer	**Parameter description**: port number of the service. Minimum: **0** Maximum: **65535**
domain	String	**Parameter description**: domain name of the service. Minimum: **4** Maximum: **255**

Table 1-270 ColumnMapping

Parameter	Type	Description
column_name	String	**Parameter description**: database column name. Minimum: **1** Maximum: **256**

Parameter	Type	Description
json_key	String	**Parameter description**: attribute name of the data to be forwarded. Minimum: **1** Maximum: **256**

Table 1-271 FunctionGraphForwarding

Parameter	Type	Description
func_urn	String	**Parameter description**: Function Uniform Resource Name (URN), which uniquely identifies a function. Minimum: **0** Maximum: **65535**
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**

Example Requests

Modifies the action of a specified rule and pushes the rule to the HTTP server.

```
PUT https://{endpoint}/v5/iot/{project_id}/routing-rule/actions/{action_id}

{
  "channel" : "HTTP_FORWARDING",
  "channel_detail" : {
    "http_forwarding" : {
      "url" : "http://host:port/callbackurltest"
    }
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "action_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "rule_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce1",
  "app_id" : "1a7ffc5cd89c44dd8265b1653d951ce0",
  "channel" : "HTTP_FORWARDING",
  "channel_detail" : {
    "http_forwarding" : {
      "url" : "http://host:port/callbackurltest"
    }
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.8.10 Delete a Rule Action

Function

This API is used by an application to delete a specific rule action from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/routing-rule/actions/{action_id}

Table 1-272 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
action_id	Yes	String	**Parameter description**: rule action ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-273 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

None

Example Requests

Deletes a specified rule action.

```
DELETE https://{endpoint}/v5/iot/{project_id}/routing-rule/actions/{action_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
401	Unauthorized

Status Code	Description
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.9 Transition Data

1.4.9.1 Push a Device Status Change Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device.status** and **event** to **update**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the device status changes.

URI

POST /HTTP URL determined when the application creates a device status change notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-274 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device.status**.
event	Yes	String	**Parameter description**: subscribed event. Set this parameter to **update**.

Parameter	Mandatory	Type	Description
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceStatusUpdateNotifyDataV5** object	**Parameter description**: message to push.

Table 1-275 DeviceStatusUpdateNotifyDataV5

Parameter	Mandatory	Type	Description
header	Yes	**NotifyDataHeader** object	**Parameter description**: message header.
body	Yes	**DeviceStatusUpdate** object	**Parameter description**: message body.

Table 1-276 NotifyDataHeader

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: application ID. Maximum: **256**
device_id	No	String	**Parameter description**: unique device ID, which is allocated by IoTDA during device registration. Maximum: **256**
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **256**
product_id	No	String	**Parameter description**: unique product ID, which is allocated by the platform during product registration. Maximum: **256**
gateway_id	No	String	**Parameter description**: gateway ID, which uniquely identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. Maximum: **256**
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-277 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Table 1-278 DeviceStatusUpdate

Parameter	Mandatory	Type	Description
status	Yes	String	**Parameter description**: current status of the device. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal.
last_online_time	Yes	String	**Parameter description**: last time when the device went online.
status_update_time	Yes	String	**Parameter description**: time when the device was updated to the current status.

Response Parameters

None

Example Requests

Example of a device status change notification.

```
Device status change notification.

{
  "resource" : "device.status",
  "event" : "update",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
```



```
  "header" : {
    "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "product_id" : "ABC123456789",
    "app_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "node_id" : "ABC123456789",
    "tags" : [ {
      "tag_value" : "testTagValue",
      "tag_key" : "testTagName"
    } ]
  }
 },
 "body" : {
  "last_online_time" : "20151212T121212Z",
  "status" : "ONLINE",
  "status_update_time" : "2015-12-12T12:12:12.815Z"
 }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.2 Push a Device Property Reporting Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device.property** and **event** to **report**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when a device reports property data.

URI

POST /HTTP URL determined when the application creates the device property reporting notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-279 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device.property**.
event	Yes	String	**Parameter description**: subscribed resource event. Set this parameter to **report**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DevicePropertyReportNotifyData** object	**Parameter description**: message to push.

Table 1-280 DevicePropertyReportNotifyData

Parameter	Mandatory	Type	Description
header	Yes	**NotifyDataHeader** object	**Parameter description**: message header.

Parameter	Mandatory	Type	Description
body	Yes	DevicePropertyReport object	**Parameter description**: message body.

Table 1-281 NotifyDataHeader

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: application ID. Maximum: **256**
device_id	No	String	**Parameter description**: unique device ID, which is allocated by IoTDA during device registration. Maximum: **256**
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **256**
product_id	No	String	**Parameter description**: unique product ID, which is allocated by the platform during product registration. Maximum: **256**
gateway_id	No	String	**Parameter description**: gateway ID, which uniquely identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. Maximum: **256**

Parameter	Mandatory	Type	Description
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-282 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Table 1-283 DevicePropertyReport

Parameter	Mandatory	Type	Description
services	Yes	Array of **DevicePropertyV5** objects	**Parameter description**: list of services of the device.

Table 1-284 DevicePropertyV5

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: device service ID, which is defined in the product model associated with the device.
properties	Yes	Object	**Parameter description**: data reported by the device.

Parameter	Mandatory	Type	Description
event_time	Yes	String	**Parameter description**: UTC time when the device reports data. The time format depends on the property format reported by the device. The second-level format is yyyyMMdd'T'HHmmss'Z', and the millisecond-level format is yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **20151212T121212Z** or **2020-08-12T12:12:12.333Z**.

Response Parameters

None

Example Requests

Example of a device property reporting notification.

```
Device Property Reporting Notification

{
  "resource" : "device.property",
  "event" : "report",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "header" : {
      "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "product_id" : "ABC123456789",
      "app_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "node_id" : "ABC123456789",
      "tags" : [ {
        "tag_value" : "testTagValue",
        "tag_key" : "testTagName"
      } ]
    }
  },
  "body" : {
    "services" : [ {
      "service_id" : "Battery",
      "properties" : {
        "batteryLevel" : 80
      },
      "event_time" : "20151212T121212Z"
    } ]
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.3 Push a Device Message Status Change Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device.message.status** and **event** to **update**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the device message status changes.

URI

POST /HTTP URL determined when the application creates a device message status change notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-285 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device.message.status**.
event	Yes	String	**Parameter description**: subscribed event. Set this parameter to **update**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.

Parameter	Mandatory	Type	Description
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceMessageStatusUpdateNotifyDataV5** object	**Parameter description**: message to push.

Table 1-286 DeviceMessageStatusUpdateNotifyDataV5

Parameter	Mandatory	Type	Description
header	Yes	**NotifyDataHeader** object	**Parameter description**: message header.
body	Yes	**DeviceMessageStatusUpdate** object	**Parameter description**: message body.

Table 1-287 NotifyDataHeader

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: application ID. Maximum: **256**
device_id	No	String	**Parameter description**: unique device ID, which is allocated by IoTDA during device registration. Maximum: **256**

Parameter	Mandatory	Type	Description
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **256**
product_id	No	String	**Parameter description**: unique product ID, which is allocated by the platform during product registration. Maximum: **256**
gateway_id	No	String	**Parameter description**: gateway ID, which uniquely identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. Maximum: **256**
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-288 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Table 1-289 DeviceMessageStatusUpdate

Parameter	Mandatory	Type	Description
topic	No	String	**Parameter description**: MQTT topic used for message reporting.
message_id	Yes	String	**Parameter description**: sequence number of a message, which uniquely identifies a message.
name	No	String	**Parameter description**: message name.
status	No	String	**Parameter description**: device message status. Options: **PENDING**, **DELIVERED**, **TIMEOUT**, and **FAILED**.
error_info	No	**ErrorInfoDTO** object	**Parameter description**: message delivery failure information.
timestamp	No	String	**Parameter description**: UTC when the message was updated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-290 ErrorInfoDTO

Parameter	Mandatory	Type	Description
error_code	No	String	**Parameter description**: error code. **IOTDA.014016** indicates that the device is offline. **IOTDA.014112** indicates that the device does not subscribe to any topic.
error_msg	No	String	**Parameter description**: error message. The device is offline or does not subscribe to any topic.

Response Parameters

None

Example Requests

Example of a device message status change notification.

Device message status change notification.

```
{
  "resource" : "device.message.status",
  "event" : "update",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "header" : {
      "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "product_id" : "ABC123456789",
      "app_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "node_id" : "ABC123456789",
      "tags" : [ {
        "tag_value" : "testTagValue",
        "tag_key" : "testTagName"
      } ]
    }
  },
  "body" : {
    "error_info" : {
      "error_msg" : "Send to device failed, device not subscribe topic.",
      "error_code" : "IOTDA.014112"
    },
    "name" : "name",
    "topic" : "topic",
    "message_id" : "1235",
    "status" : "DELIVERED",
    "timestamp" : "20151212T121212Z"
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.4 Push a Batch Task Status Change Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **batchtask** and **event** to **update**), the API for **creating a rule**

action, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the batch task status changes.

URI

POST /HTTP URL determined when the application creates a batch task status change notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-291 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **batchtask**.
event	Yes	String	**Parameter description**: subscribed event. Set this parameter to **update**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**BatchTaskUpdateNotifyData** object	**Parameter description**: custom field list of the device.

Table 1-292 BatchTaskUpdateNotifyData

Parameter	Mandatory	Type	Description
body	Yes	**BatchTaskUpdate** object	**Parameter description**: message body.

Table 1-293 BatchTaskUpdate

Parameter	Mandatory	Type	Description
app_id	Yes	String	**Parameter description**: application ID.
task_id	Yes	String	**Parameter description**: batch task ID. The value is returned when the API for creating the batch task is called.
task_type	Yes	String	**Parameter description**: task type. • **firmwareUpgrade**: firmware upgrade • **softwareUpgrade**: software upgrade
status	Yes	String	**Parameter description**: task status. • **Waitting**: The batch task is waiting to be executed. • **Processing**: The batch task is being executed. • **Success**: The batch task is executed. • **PartialSuccess**: Only some subtasks in the batch task are executed. • **Fail**: The batch task fails to be executed. • **Stopped**: The batch task is stopped.
status_desc	Yes	String	**Parameter description**: task status description.

Response Parameters

None

Example Requests

Example of a bulk task status change notification.

```
Batch Task Status Change Notification

{
  "resource" : "batchtask",
  "event" : "update",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "body" : {
      "status_desc" : "status_desc",
      "task_id" : "1a7ffc5c-d89c-44dd-8265",
      "task_type" : "softwareUpgrade",
      "app_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
      "status" : "Waitting"
    }
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.5 Push a Device Message Reporting Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device.message** and **event** to **report**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when a device reports message data.

URI

POST /HTTP URL determined when the application creates the device message reporting notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-294 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device.message**.
event	Yes	String	**Parameter description**: Subscribed resource event. Set this parameter to **report**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceMessageReportNotifyData** object	**Parameter description**: custom field list of the device.

Table 1-295 DeviceMessageReportNotifyData

Parameter	Mandatory	Type	Description
header	Yes	**NotifyDataHeader** object	**Parameter description**: message header.

Parameter	Mandatory	Type	Description
body	Yes	DeviceMessageReport object	**Parameter description**: message body.

Table 1-296 NotifyDataHeader

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: application ID. Maximum: **256**
device_id	No	String	**Parameter description**: unique device ID, which is allocated by IoTDA during device registration. Maximum: **256**
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **256**
product_id	No	String	**Parameter description**: unique product ID, which is allocated by the platform during product registration. Maximum: **256**
gateway_id	No	String	**Parameter description**: gateway ID, which uniquely identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. Maximum: **256**

Parameter	Mandatory	Type	Description
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-297 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Table 1-298 DeviceMessageReport

Parameter	Mandatory	Type	Description
topic	Yes	String	**Parameter description**: MQTT topic used for device reporting.
content	Yes	Object	**Parameter description**: message content.
properties	No	**MqttPropertiesDTO** object	**Parameter description**: property information carried by the device.

Table 1-299 MqttPropertiesDTO

Parameter	Mandatory	Type	Description
correlation_data	No	String	**Parameter description**: data in the request and response modes of MQTT 5.0. This parameter is optional. Devices can use this parameter to configure data in the request and response modes of MQTT. Maximum: **128**
response_topic	No	String	**Parameter description**: response topic in the request and response modes of MQTT 5.0. This parameter is optional. Devices can use this parameter to configure response topic in the request and response modes of MQTT. Maximum: **128**
content_type	No	String	**Parameter description**: content type of the MQTT 5.0 payload. This parameter is optional. Devices can use this parameter to configure the content type of the MQTT payload. Maximum: **128**
user_properties	No	Array of **UserPropDTO** objects	**Parameter description**: custom property. This parameter is optional. Devices can use this parameter to configure custom properties.

Table 1-300 UserPropDTO

Parameter	Mandatory	Type	Description
prop_key	No	String	**Parameter description**: key of the custom property. Maximum: **128**
prop_value	No	String	**Parameter description**: value of the custom property. Maximum: **128**

Response Parameters

None

Example Requests

Example of a device message reporting notification.

Device message reporting notification.

```
{
  "resource" : "device.message",
  "event" : "report",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "header" : {
      "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "product_id" : "ABC123456789",
      "app_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "node_id" : "ABC123456789",
      "tags" : [ {
        "tag_value" : "testTagValue",
        "tag_key" : "testTagName"
      } ]
    }
  },
  "body" : {
    "topic" : "topic",
    "content" : "msg",
    "properties" : {
      "response_topic" : "/device/message/response",
      "content_type" : "text/plain",
      "user_properties" : [ {
        "prop_value" : "propValue1",
        "prop_key" : "propKey1"
      } ],
      "correlation_data" : "messageName"
    }
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.6 Push a Device Addition Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device** and **event** to **create**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the device is added.

URI

POST /HTTP URL determined when the application creates the device addition notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-301 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device**.
event	Yes	String	**Parameter description**: subscribed resource event. Set this parameter to **create**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.

Parameter	Mandatory	Type	Description
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceCreateOrUpdateNotifyData** object	**Parameter description**: message to push.

Table 1-302 DeviceCreateOrUpdateNotifyData

Parameter	Mandatory	Type	Description
body	Yes	**QueryDeviceBase** object	**Parameter description**: message content.

Table 1-303 QueryDeviceBase

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. Maximum: **36**
app_name	No	String	**Parameter description**: resource space name.
device_id	No	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. Maximum: **256**
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **64**

Parameter	Mandatory	Type	Description
gateway_id	No	String	**Parameter description**: gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates. Maximum: **256**
device_name	No	String	**Parameter description**: device name. Maximum: **256**
node_type	No	String	**Parameter description**: device node type. • **ENDPOINT**: indirectly connected device • **GATEWAY**: directly connected device or gateway • **UNKNOWN**: unknown node type
description	No	String	**Parameter description**: device description. Maximum: **2048**
fw_version	No	String	**Parameter description**: firmware version of the device. Maximum: **256**
sw_version	No	String	**Parameter description**: software version of the device. Maximum: **256**
device_sdk_version	No	String	**Parameter description**: SDK version of the device. Maximum: **256**
auth_info	No	**AuthInfo** object	**Parameter description**: access authentication information about the device.
product_id	No	String	**Parameter description**: unique ID of the product associated with the device.

Parameter	Mandatory	Type	Description
product_name	No	String	**Parameter description**: name of the product associated with the device.
status	No	String	**Parameter description**: device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FREEZED**: The device is frozen.
create_time	No	String	**Parameter description**: time when a device was registered on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of device tags.
extension_info	No	Object	**Parameter description**: extended information, which can be customized. If this parameter is specified for a child device during device creation, the platform will notify the gateway of the extended information through the MQTT API used for notifying a gateway of new child device connection.

Table 1-304 AuthInfo

Parameter	Mandatory	Type	Description
auth_type	No	String	**Parameter description**: authentication type. Secret authentication (**SECRET**) and certificate authentication (**CERTIFICATES**) are supported. If secret authentication is used, fill in **secret**. If certificate authentication is used, fill in **fingerprint**. If **auth_type** is not set, secret authentication is used by default.
secure_access	No	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. The default value is **true**. • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Default: **true**
timeout	No	Integer	**Parameter description**: validity period of the device verification code, in seconds. The default value is **0**. If the device has not been connected to the platform within the validity period, the platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the verification code is always valid. Note: This parameter is returned only when **timeout** is modified in the device registration or modification API. Minimum: **0** Maximum: **2147483647** Default: **0**

Table 1-305 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Response Parameters

None

Example Requests

Example of a device addition notification.

```
Device addition notification.

{
  "resource" : "device",
  "event" : "create",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "body" : {
      "device_sdk_version" : "C_v0.5.0",
      "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "create_time" : "20190303T081011Z",
      "description" : "watermeter device",
      "auth_info" : {
        "auth_type" : "SECRET",
        "secure_access" : true,
        "timeout" : 300
      },
      "product_name" : "Thermometer",
      "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "sw_version" : "1.1.0",
      "tags" : [ {
        "tag_value" : "testTagValue",
        "tag_key" : "testTagName"
      } ],
      "extension_info" : {
        "aaa" : "xxx",
        "bbb" : 0
      },
      "app_name" : "testAPP01",
      "device_name" : "dianadevice",
      "node_type" : "ENDPOINT",
      "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
      "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
      "fw_version" : "1.1.0",
      "node_id" : "ABC123456789",
      "status" : "INACTIVE"
    }
```



```
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.7 Push a Device Update Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device** and **event** to **update**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the device is updated.

URI

POST /HTTP URL determined when the application creates the device update notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-306 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device**.
event	Yes	String	**Parameter description**: subscribed event. Set this parameter to **update**.

Parameter	Mandatory	Type	Description
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceCreateOrUpdateNotifyData** object	**Parameter description**: message to push.

Table 1-307 DeviceCreateOrUpdateNotifyData

Parameter	Mandatory	Type	Description
body	Yes	**QueryDeviceBase** object	**Parameter description**: message content.

Table 1-308 QueryDeviceBase

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. Maximum: **36**

Parameter	Mandatory	Type	Description
app_name	No	String	**Parameter description**: resource space name.
device_id	No	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. Maximum: **256**
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **64**
gateway_id	No	String	**Parameter description**: gateway ID, which is the device ID of the parent device. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device with which it associates. Maximum: **256**
device_name	No	String	**Parameter description**: device name. Maximum: **256**
node_type	No	String	**Parameter description**: device node type. • **ENDPOINT**: indirectly connected device • **GATEWAY**: directly connected device or gateway • **UNKNOWN**: unknown node type
description	No	String	**Parameter description**: device description. Maximum: **2048**

Parameter	Mandatory	Type	Description
fw_version	No	String	**Parameter description**: firmware version of the device. Maximum: **256**
sw_version	No	String	**Parameter description**: software version of the device. Maximum: **256**
device_sdk_version	No	String	**Parameter description**: SDK version of the device. Maximum: **256**
auth_info	No	**AuthInfo** object	**Parameter description**: access authentication information about the device.
product_id	No	String	**Parameter description**: unique ID of the product associated with the device.
product_name	No	String	**Parameter description**: name of the product associated with the device.
status	No	String	**Parameter description**: device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **ABNORMAL**: The device is abnormal. • **INACTIVE**: The device is not activated. • **FREEZED**: The device is frozen.
create_time	No	String	**Parameter description**: time when a device was registered on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of device tags.

Parameter	Mandatory	Type	Description
extension_info	No	Object	**Parameter description**: extended information, which can be customized. If this parameter is specified for a child device during device creation, the platform will notify the gateway of the extended information through the MQTT API used for notifying a gateway of new child device connection.

Table 1-309 AuthInfo

Parameter	Mandatory	Type	Description
auth_type	No	String	**Parameter description**: authentication type. Secret authentication (**SECRET**) and certificate authentication (**CERTIFICATES**) are supported. If secret authentication is used, fill in **secret**. If certificate authentication is used, fill in **fingerprint**. If **auth_type** is not set, secret authentication is used by default.
secure_access	No	Boolean	**Parameter description**: whether the device is connected to the platform using a secure protocol. The default value is **true**. • **true**: The device is connected to the platform using a secure protocol. • **false**: The device is connected to the platform using an insecure protocol. Default: **true**

Parameter	Mandatory	Type	Description
timeout	No	Integer	**Parameter description**: validity period of the device verification code, in seconds. The default value is **0**. If the device has not been connected to the platform within the validity period, the platform deletes the registration information of the device. If this parameter is set to **0** (recommended), the verification code is always valid. Note: This parameter is returned only when **timeout** is modified in the device registration or modification API. Minimum: **0** Maximum: **2147483647** Default: **0**

Table 1-310 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Response Parameters

None

Example Requests

Example of a device update notification.

```
Device update notification.

{
  "resource" : "device",
  "event" : "update",
```



```
"event_time" : "20151212T121212Z",
"event_time_ms" : "2015-12-12T12:12:12.000Z",
"request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
"notify_data" : {
  "body" : {
    "device_sdk_version" : "C_v0.5.0",
    "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "create_time" : "20190303T081011Z",
    "description" : "watermeter device",
    "auth_info" : {
      "auth_type" : "SECRET",
      "secure_access" : true,
      "timeout" : 300
    },
    "product_name" : "Thermometer",
    "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "sw_version" : "1.1.0",
    "tags" : [ {
      "tag_value" : "testTagValue",
      "tag_key" : "testTagName"
    } ],
    "extension_info" : {
      "aaa" : "xxx",
      "bbb" : 0
    },
    "app_name" : "testAPP01",
    "device_name" : "dianadevice",
    "node_type" : "ENDPOINT",
    "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
    "fw_version" : "1.1.0",
    "node_id" : "ABC123456789",
    "status" : "INACTIVE"
  }
 }
}
```


Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.8 Push a Device Deletion Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device** and **event** to **delete**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a

rule, the platform pushes the result to the server specified by the rule when the device is deleted.

URI

POST /HTTP URL determined when the application creates the device deletion notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-311 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device**.
event	Yes	String	**Parameter description**: subscribed resource event. Set this parameter to **delete**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceDeleteNotifyData** object	**Parameter description**: message to push.

Table 1-312 DeviceDeleteNotifyData

Parameter	Mandatory	Type	Description
header	Yes	**NotifyDataHeader** object	**Parameter description**: message header.

Table 1-313 NotifyDataHeader

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: application ID. Maximum: **256**
device_id	No	String	**Parameter description**: unique device ID, which is allocated by IoTDA during device registration. Maximum: **256**
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **256**
product_id	No	String	**Parameter description**: unique product ID, which is allocated by the platform during product registration. Maximum: **256**
gateway_id	No	String	**Parameter description**: gateway ID, which uniquely identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. Maximum: **256**

Parameter	Mandatory	Type	Description
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-314 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Response Parameters

None

Example Requests

Example of a device deletion notification.

```
Device deletion notification.

{
  "resource" : "device",
  "event" : "delete",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "header" : {
      "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "product_id" : "ABC123456789",
      "app_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "node_id" : "ABC123456789",
      "tags" : [ {
        "tag_value" : "testTagValue",
        "tag_key" : "testTagName"
      } ]
    }
  }
}
```


Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.9 Push a Product Addition Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **product** and **event** to **create**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the product is added.

URI

POST /HTTP URL determined when the application creates the product addition notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-315 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **product**.
event	Yes	String	**Parameter description**: subscribed resource event. Set this parameter to **create**.

Parameter	Mandatory	Type	Description
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**ProductUpdateNotifyData** object	**Parameter description**: message to push.

Table 1-316 ProductUpdateNotifyData

Parameter	Mandatory	Type	Description
body	Yes	**Product** object	**Parameter description**: message content.

Table 1-317 Product

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID.
app_name	No	String	**Parameter description**: resource space name.

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	No	String	**Parameter description**: product name.
device_type	No	String	**Parameter description**: device type.
protocol_type	No	String	**Parameter description**: protocol used by the device. Options: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, and **ONVIF**.
data_format	No	String	**Parameter description**: format of the data reported by the device. Options: **json** and **binary**.
manufacturer_name	No	String	**Parameter description**: manufacturer name.
industry	No	String	**Parameter description**: industry to which the device belongs.
description	No	String	**Parameter description**: product description.
service_capabilities	No	Array of **ServiceCapability** objects	**Parameter description**: list of service capabilities of the device.
create_time	No	String	**Parameter description**: UTC time when a product was created on the platform. The format is yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-318 ServiceCapability

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: service ID of the device.

Parameter	Mandatory	Type	Description
service_type	Yes	String	**Parameter description**: service type of the device.
properties	No	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service.
commands	No	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service.
events	No	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service.
description	No	String	**Parameter description**: device service description.
option	No	String	**Parameter description**: whether the device service is mandatory. Options: **Master** (main service), **Mandatory** (mandatory service), and **Optional** (optional service). Currently, this field is a non-functional field, which is used only for identification. The default value is **Optional**. Default: **Optional**

Table 1-319 ServiceProperty

Parameter	Mandatory	Type	Description
property_name	Yes	String	**Parameter description**: device property name.
required	No	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**
data_type	Yes	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.

Parameter	Mandatory	Type	Description
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	No	String	**Parameter description**: minimum value of the device property. Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the device property. Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the device property.
step	No	Double	**Parameter description**: step of the device property.
unit	No	String	**Parameter description**: unit of the device property. Maximum: **16**
method	Yes	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.
description	No	String	**Parameter description**: device property description.

Parameter	Mandatory	Type	Description
default_value	No	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-320 ServiceCommand

Parameter	Mandatory	Type	Description
command_name	Yes	String	**Parameter description**: device command name.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.
responses	No	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-321 ServiceCommandResponse

Parameter	Mandatory	Type	Description
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.
response_name	Yes	String	**Parameter description**: name of the device command response.

Table 1-322 ServiceEvent

Parameter	Mandatory	Type	Description
event_type	Yes	String	**Parameter description**: device event type.

Parameter	Mandatory	Type	Description
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-323 ServiceCommandPara

Parameter	Mandatory	Type	Description
para_name	Yes	String	**Parameter description**: parameter name.
required	No	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**
data_type	Yes	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the parameter.
min	No	String	**Parameter description**: minimum value of the parameter. Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the parameter. Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the parameter.
step	No	Double	**Parameter description**: step of the parameter.

Parameter	Mandatory	Type	Description
unit	No	String	**Parameter description**: unit of the parameter. Maximum: **16**
description	No	String	**Parameter description**: parameter description.

Response Parameters

None

Example Requests

Example of a product addition notification.

```
Product Addition Notification

{
  "resource" : "product",
  "event" : "create",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "body" : {
      "app_name" : "testAPP01",
      "protocol_type" : "CoAP",
      "data_format" : "binary",
      "service_capabilities" : [ {
        "service_type" : "temperature",
        "service_id" : "temperature",
        "description" : "temperature",
        "properties" : [ {
          "unit" : "centigrade",
          "min" : "1",
          "method" : "R",
          "max" : "100",
          "data_type" : "decimal",
          "description" : "force",
          "step" : 0.1,
          "default_value" : {
            "color" : "red",
            "size" : 1
          },
          "enum_list" : [ "string" ],
          "required" : true,
          "property_name" : "temperature",
          "max_length" : 100
        } ],
        "commands" : [ {
          "command_name" : "reboot",
          "responses" : [ {
            "response_name" : "ACK",
            "paras" : [ {
              "unit" : "km/h",
              "min" : "1",
              "max" : "100",
              "para_name" : "force",
              "data_type" : "string",
              "description" : "force",
```



```
            "step" : 0.1,
            "enum_list" : [ "string" ],
            "required" : false,
            "max_length" : 100
          } ]
        } ],
        "paras" : [ {
          "unit" : "km/h",
          "min" : "1",
          "max" : "100",
          "para_name" : "force",
          "data_type" : "string",
          "description" : "force",
          "step" : 0.1,
          "enum_list" : [ "string" ],
          "required" : false,
          "max_length" : 100
        } ]
      } ],
      "events" : [ {
        "event_type" : "reboot",
        "paras" : [ {
          "unit" : "km/h",
          "min" : "1",
          "max" : "100",
          "para_name" : "force",
          "data_type" : "string",
          "description" : "force",
          "step" : 0.1,
          "enum_list" : [ "string" ],
          "required" : false,
          "max_length" : 100
        } ]
      } ],
      "option" : "Mandatory"
    } ],
    "create_time" : "20190303T081011Z",
    "product_id" : "5ba24f5ebbe8f56f5a14f605",
    "name" : "Thermometer",
    "description" : "this is a thermometer produced by Huawei",
    "device_type" : "Thermometer",
    "industry" : "smartCity",
    "manufacturer_name" : "ABC",
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
  }
 }
}
```


Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.10 Push a Product Update Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **product** and **event** to **update**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the product is updated.

URI

POST /HTTP URL determined when the application creates the product update notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-324 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **product**.
event	Yes	String	**Parameter description**: subscribed event. Set this parameter to **update**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.

Parameter	Mandatory	Type	Description
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**ProductUpdateNotifyData** object	**Parameter description**: message to push.

Table 1-325 ProductUpdateNotifyData

Parameter	Mandatory	Type	Description
body	Yes	**Product** object	**Parameter description**: message content.

Table 1-326 Product

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID.
app_name	No	String	**Parameter description**: resource space name.
product_id	No	String	**Parameter description**: product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	No	String	**Parameter description**: product name.
device_type	No	String	**Parameter description**: device type.
protocol_type	No	String	**Parameter description**: protocol used by the device. Options: **MQTT**, **CoAP**, **HTTP**, **HTTPS**, **Modbus**, and **ONVIF**.
data_format	No	String	**Parameter description**: format of the data reported by the device. Options: **json** and **binary**.

Parameter	Mandatory	Type	Description
manufacturer_name	No	String	**Parameter description**: manufacturer name.
industry	No	String	**Parameter description**: industry to which the device belongs.
description	No	String	**Parameter description**: product description.
service_capabilities	No	Array of **ServiceCapability** objects	**Parameter description**: list of service capabilities of the device.
create_time	No	String	**Parameter description**: UTC time when a product was created on the platform. The format is yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-327 ServiceCapability

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: service ID of the device.
service_type	Yes	String	**Parameter description**: service type of the device.
properties	No	Array of **ServiceProperty** objects	**Parameter description**: list of properties supported by the device service.
commands	No	Array of **ServiceCommand** objects	**Parameter description**: list of commands supported by the device service.
events	No	Array of **ServiceEvent** objects	**Parameter description**: list of events supported by the device service.
description	No	String	**Parameter description**: device service description.

Parameter	Mandatory	Type	Description
option	No	String	**Parameter description**: whether the device service is mandatory. Options: **Master** (main service), **Mandatory** (mandatory service), and **Optional** (optional service). Currently, this field is a non-functional field, which is used only for identification. The default value is **Optional**. Default: **Optional**

Table 1-328 ServiceProperty

Parameter	Mandatory	Type	Description
property_name	Yes	String	**Parameter description**: device property name.
required	No	Boolean	**Parameter description**: whether the device property is mandatory. The default value is **false**. Default: **false**
data_type	Yes	String	**Parameter description**: data type of the device property. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the device property.
min	No	String	**Parameter description**: minimum value of the device property. Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the device property. Minimum: **1** Maximum: **16**

Parameter	Mandatory	Type	Description
max_length	No	Integer	**Parameter description**: maximum length of the device property.
step	No	Double	**Parameter description**: step of the device property.
unit	No	String	**Parameter description**: unit of the device property. Maximum: **16**
method	Yes	String	**Parameter description**: access mode of the device property. Options: **RWE**, **RW**, **RE**, **WE**, **E**, **W**, and **R**. • **R**: The property value can be read. • **W**: The property value can be written. • **E**: The property value can be subscribed to, that is, an event is reported when the property value changes.
description	No	String	**Parameter description**: device property description.
default_value	No	Object	**Parameter description**: default value of the device property. If the default value is set, it will be written to the desired data of the device shadow when the product is used to create a device. When the device goes online, the default value will be delivered to the device.

Table 1-329 ServiceCommand

Parameter	Mandatory	Type	Description
command_name	Yes	String	**Parameter description**: device command name.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device command parameters.

Parameter	Mandatory	Type	Description
responses	No	Array of **ServiceCommandResponse** objects	**Parameter description**: list of response parameters of the device command.

Table 1-330 ServiceCommandResponse

Parameter	Mandatory	Type	Description
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of response parameters of the device command.
response_name	Yes	String	**Parameter description**: name of the device command response.

Table 1-331 ServiceEvent

Parameter	Mandatory	Type	Description
event_type	Yes	String	**Parameter description**: device event type.
paras	No	Array of **ServiceCommandPara** objects	**Parameter description**: list of device event parameters.

Table 1-332 ServiceCommandPara

Parameter	Mandatory	Type	Description
para_name	Yes	String	**Parameter description**: parameter name.
required	No	Boolean	**Parameter description**: whether the parameter is mandatory. The default value is **false**. Default: **false**

Parameter	Mandatory	Type	Description
data_type	Yes	String	**Parameter description**: data type of the parameter. Options: **int**, **long**, **decimal**, **string**, **DateTime**, **jsonObject**, **enum**, **boolean**, and **string list**.
enum_list	No	Array of strings	**Parameter description**: list of enumerated values of the parameter.
min	No	String	**Parameter description**: minimum value of the parameter. Minimum: **1** Maximum: **16**
max	No	String	**Parameter description**: maximum value of the parameter. Minimum: **1** Maximum: **16**
max_length	No	Integer	**Parameter description**: maximum length of the parameter.
step	No	Double	**Parameter description**: step of the parameter.
unit	No	String	**Parameter description**: unit of the parameter. Maximum: **16**
description	No	String	**Parameter description**: parameter description.

Response Parameters

None

Example Requests

Example of a product update notification.

```
Product update notification

{
  "resource" : "product",
  "event" : "update",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
```



```
"request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
"notify_data" : {
  "body" : {
    "app_name" : "testAPP01",
    "protocol_type" : "CoAP",
    "data_format" : "binary",
    "service_capabilities" : [ {
      "service_type" : "temperature",
      "service_id" : "temperature",
      "description" : "temperature",
      "properties" : [ {
        "unit" : "centigrade",
        "min" : "1",
        "method" : "R",
        "max" : "100",
        "data_type" : "decimal",
        "description" : "force",
        "step" : 0.1,
        "default_value" : {
          "color" : "red",
          "size" : 1
        },
        "enum_list" : [ "string" ],
        "required" : true,
        "property_name" : "temperature",
        "max_length" : 100
      } ],
      "commands" : [ {
        "command_name" : "reboot",
        "responses" : [ {
          "response_name" : "ACK",
          "paras" : [ {
            "unit" : "km/h",
            "min" : "1",
            "max" : "100",
            "para_name" : "force",
            "data_type" : "string",
            "description" : "force",
            "step" : 0.1,
            "enum_list" : [ "string" ],
            "required" : false,
            "max_length" : 100
          } ]
        } ],
        "paras" : [ {
          "unit" : "km/h",
          "min" : "1",
          "max" : "100",
          "para_name" : "force",
          "data_type" : "string",
          "description" : "force",
          "step" : 0.1,
          "enum_list" : [ "string" ],
          "required" : false,
          "max_length" : 100
        } ]
      } ],
      "events" : [ {
        "event_type" : "reboot",
        "paras" : [ {
          "unit" : "km/h",
          "min" : "1",
          "max" : "100",
          "para_name" : "force",
          "data_type" : "string",
          "description" : "force",
          "step" : 0.1,
          "enum_list" : [ "string" ],
          "required" : false,
```

```
          "max_length" : 100
        } ]
      } ],
      "option" : "Mandatory"
    } ],
    "create_time" : "20190303T081011Z",
    "product_id" : "5ba24f5ebbe8f56f5a14f605",
    "name" : "Thermometer",
    "description" : "this is a thermometer produced by Huawei",
    "device_type" : "Thermometer",
    "industry" : "smartCity",
    "manufacturer_name" : "ABC",
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
  }
 }
}
```


Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.11 Push a Product Deletion Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **product** and **event** to **delete**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the product is deleted.

URI

POST /HTTP URL determined when the application creates the product deletion notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-333 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **product**.
event	Yes	String	**Parameter description**: subscribed resource event. Set this parameter to **delete**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2015-12-12T12:12:12.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**ProductDeleteNotifyData** object	**Parameter description**: message to push.

Table 1-334 ProductDeleteNotifyData

Parameter	Mandatory	Type	Description
body	Yes	**DeletedProduct** object	**Parameter description**: message content.

Table 1-335 DeletedProduct

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID.
product_id	No	String	**Parameter description**: product ID, which uniquely identifies a product. It is allocated by the platform after the product is created on the platform.
name	No	String	**Parameter description**: product name.

Response Parameters

None

Example Requests

Example of a product deletion notification.

```
Product deletion notification.

{
  "resource" : "product",
  "event" : "delete",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "body" : {
      "product_id" : "5ba24f5ebbe8f56f5a14f605",
      "name" : "Thermometer",
      "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
    }
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.9.12 Push an Asynchronous Device Command Status Change Notification

Function

After the application calls the API for **creating a rule triggering condition** (**resource** is set to **device.command.status** and **event** to **update**), the API for **creating a rule action**, and the API for **modifying a rule triggering condition**, and activates a rule, the platform pushes the result to the server specified by the rule when the command status changes.

URI

POST /URL (HTTP) determined when the application creates an asynchronous device command status change notification rule. The AMQP channel does not require the URL.

Request Parameters

Table 1-336 Request body parameters

Parameter	Mandatory	Type	Description
resource	Yes	String	**Parameter description**: subscribed resource name. Set this parameter to **device.command.status**.
event	Yes	String	**Parameter description**: subscribed event. Set this parameter to **update**.
event_time	Yes	String	**Parameter description**: UTC time when the resource event was generated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. If necessary, convert the time to display in the local time zone format.

Parameter	Mandatory	Type	Description
event_time_ms	No	String	**Parameter description**: UTC time when a resource event was generated. The value is in the format of yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, **2019-03-03T08:10:11.000Z**. If necessary, convert the time to display in the local time zone format.
request_id	No	String	**Parameter description**: message ID, which is specified by the device or generated by the platform, and is used to trace the service process.
notify_data	Yes	**DeviceCommandStatusUpdateNotifyDataV5** object	**Parameter description**: message to push.

Table 1-337 DeviceCommandStatusUpdateNotifyDataV5

Parameter	Mandatory	Type	Description
header	Yes	**NotifyDataHeader** object	**Parameter description**: message header.
body	Yes	**DeviceCommandStatusUpdate** object	**Parameter description**: message body.

Table 1-338 NotifyDataHeader

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: application ID. Maximum: **256**
device_id	No	String	**Parameter description**: unique device ID, which is allocated by IoTDA during device registration. Maximum: **256**

Parameter	Mandatory	Type	Description
node_id	No	String	**Parameter description**: device node ID. This parameter is set to the IMEI, MAC address, or serial number. Maximum: **256**
product_id	No	String	**Parameter description**: unique product ID, which is allocated by the platform during product registration. Maximum: **256**
gateway_id	No	String	**Parameter description**: gateway ID, which uniquely identifies a gateway. The gateway ID is the same as the device ID if the device is a directly connected device. If the device is an indirectly connected device, the gateway ID is the device ID of the directly connected device (that is, the gateway) with which it associates. Maximum: **256**
tags	No	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-339 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added.
tag_value	No	String	**Parameter description**: tag value.

Table 1-340 DeviceCommandStatusUpdate

Parameter	Mandatory	Type	Description
command_id	Yes	String	**Parameter description**: command ID, which uniquely identifies a command.
created_time	No	String	**Parameter description**: UTC time when the command was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
sent_time	No	String	**Parameter description**: UTC time when the platform sent the command. The value is in the format of yyyyMMdd'T'HHmmss'Z'. If the command was sent immediately, the value was the same as the command creation time. If the command had been cached, the value was the time when the command was actually sent.
delivered_time	No	String	**Parameter description**: UTC time when the device received the command. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
response_time	No	String	**Parameter description**: UTC time when the device responded to the command. The value is in the format of yyyyMMdd'T'HHmmss'Z'.

Parameter	Mandatory	Type	Description
status	No	String	**Parameter description**: status of the delivered command. • **PENDING**: The command is not delivered and is cached on the platform. • **EXPIRED**: The command has expired, that is, the cache time exceeds the value of **expireTime**. • **SENT**: The command is being delivered. • **DELIVERED**: The command has been delivered. • **SUCCESSFUL**: The command has been executed. • **FAILED**: The command fails to be executed. • **TIMEOUT**: After a command is delivered, no response is received from the device or the response times out.
result	No	Object	**Parameter description**: detailed command execution result, which is returned by the device in JSON format.

Response Parameters

None

Example Requests

Asynchronous device command status change notification.

```
Asynchronous device command status change notification.

{
  "resource" : "device.command.status",
  "event" : "update",
  "event_time" : "20151212T121212Z",
  "event_time_ms" : "2015-12-12T12:12:12.000Z",
  "request_id" : "3fe58d5e-8697-4849-a165-7db128f7e776",
  "notify_data" : {
    "header" : {
      "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "product_id" : "ABC123456789",
      "app_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
      "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
```



```
    "node_id" : "ABC123456789",
    "tags" : [ {
      "tag_value" : "testTagValue",
      "tag_key" : "testTagName"
    } ]
  },
  "body" : {
    "result" : {
      "key" : "value"
    },
    "created_time" : "20151212T121212Z",
    "sent_time" : "20151212T121212Z",
    "command_id" : "id",
    "delivered_time" : "20151212T131212Z",
    "response_time" : "20151212T131212Z",
    "status" : "SUCCESSFUL"
  }
 }
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.10 Rule Management

1.4.10.1 Creating a Rule

Function

This API is used by an application to create a rule on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/rules

Table 1-341 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-342 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-343 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	**Parameter description**: rule name. Minimum: **1** Maximum: **128**

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: rule description. Maximum: **256**
condition_group	Yes	**ConditionGroup** object	**Parameter description**: condition group of a rule, including simple and complex rules.
actions	Yes	Array of **RuleAction** objects	**Parameter description**: action list of the rule. A maximum of 10 actions can be configured for a rule.
rule_type	Yes	String	**Parameter description**: rule type. **Options**: • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.
status	No	String	**Parameter description**: rule status. The default value is **active**. **Options**: • **active**: activated. • **inactive**: not activated.
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the rule to create will belong. If this parameter is not specified, the rule to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
device_side	No	**DeviceSide** object	**Parameter description**: information of devices to which the device-side rule is delivered. This parameter is mandatory when **rule_type** is set to **DEVICE_SIDE**.

Table 1-344 ConditionGroup

Parameter	Mandatory	Type	Description
conditions	No	Array of **RuleCondition** objects	**Parameter description**: condition list of the rule. A maximum of 10 conditions can be configured for a rule.
logic	No	String	**Parameter description**: logical relationship between multiple conditions of the rule. The default value is **and**. **Options**: • **and**: All of the conditions are judged. • **or**: One of the conditions is judged.
time_range	No	**TimeRange** object	**Parameter description**: validity period of a condition.

Table 1-345 RuleCondition

Parameter	Mandatory	Type	Description
type	Yes	String	**Parameter description**: type of the rule condition. **Options**: • **DEVICE_DATA**: device property data condition. • **SIMPLE_TIMER**: simple timing condition. • **DAILY_TIMER**: daily scheduled condition. • **DEVICE_LINKAGE_STATUS**: device status condition.
device_property_condition	No	**DeviceDataCondition** object	**Parameter description**: details of the device data condition. This parameter is mandatory when **type** is set to **DEVICE_DATA**.
simple_timer_condition	No	**SimpleTimerType** object	**Parameter description**: details of the simple timing condition. This parameter is mandatory when **type** is set to **SIMPLE_TIMER**.

Parameter	Mandatory	Type	Description
daily_timer_condition	No	**DailyTimerType** object	**Parameter description**: details of the daily scheduled condition. This parameter is mandatory when **type** is set to **DAILY_TIMER**.
device_linkage_status_condition	No	**DeviceLinkageStatusCondition** object	**Parameter description**: details of the device status condition. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_LINKAGE_STATUS**.

Table 1-346 DeviceDataCondition

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device property triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device property triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
filters	No	Array of **PropertyFilter** objects	Data filter criteria.

Table 1-347 PropertyFilter

Parameter	Mandatory	Type	Description
path	Yes	String	**Parameter description**: path of the device properties. The value is in the format of *[service_id]/DataProperty*. For example, the path of the door sensor status is **DoorWindow/status**. Maximum: **128**
operator	Yes	String	**Parameter description**: data comparison operator. **Options**: **>**, **<**, **>=**, **<=**, **=**, **in** (match in specified values), and **between** (value range).

Parameter	Mandatory	Type	Description
value	No	String	**Parameter description**: rvalue of the data comparison expression. When this parameter is used together with the data comparison operator **between**, the rvalue indicates the minimum value and maximum value, which are separated by commas (,). For example, **20,30** indicates that the value is greater than or equal to 20 and less than 30. Maximum: **64**
in_values	No	Array of strings	**Parameter description**: When **operator** is set to **in**, this parameter is used to transfer the rvalue of the comparison expression. Up to 20 values are allowed. Maximum: **128**
strategy	No	**Strategy** object	**Parameter description**: rule triggering policy, which is used to determine whether the last data meets the conditions. If **rule_type** is set to **DEVICE_LINKAGE**, either this parameter must be specified. This parameter is not supported by device-side rules.

Table 1-348 Strategy

Parameter	Mandatory	Type	Description
trigger	No	String	**Parameter description**: rule triggering judgment policy. The default value is **pulse**. **Options**: • **pulse**: If the reported data meets conditions, the rule is triggered and the previous data is not judged. • **reverse**: When the data reported this time meets the conditions, the rule is triggered even if the data reported last time does not meet the conditions.
event_valid_time	No	Integer	**Parameter description**: validity period of the device data, in seconds. The time when the device data is generated is the same as the value of **eventTime** in the reported data. Minimum: **-1**

Table 1-349 SimpleTimerType

Parameter	Mandatory	Type	Description
start_time	Yes	String	**Parameter description**: start time for rule triggering. The UTC time is used. The value is in the format of yyyyMMdd'T'HHmmss'Z'. Maximum: **128**
repeat_interval	Yes	Integer	**Parameter description**: interval for triggering the rule, in seconds. Minimum: **1** Maximum: **31536000**
repeat_count	Yes	Integer	**Parameter description**: number of times that a rule is triggered. Minimum: **1** Maximum: **9999**

Table 1-350 DailyTimerType

Parameter	Mandatory	Type	Description
time	Yes	String	**Parameter description**: time when a rule is triggered. The value is in the format of HH:MM. Maximum: **128**
days_of_week	No	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. Maximum: **128**

Table 1-351 DeviceLinkageStatusCondition

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device status triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device status triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
status_list	No	Array of strings	**Parameter description**: status list. This parameter must be carried for a device status condition. **Options**: • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. Maximum: **128**
duration	No	Integer	**Duration**: device status duration. Value range: 0–60 (minutes). Minimum: **0** Maximum: **60** Default: **0**

Table 1-352 TimeRange

Parameter	Mandatory	Type	Description
start_time	Yes	String	**Parameter description**: start time for rule triggering. The value is in the format of HH:mm.
end_time	Yes	String	**Parameter description**: end time for rule triggering. The value is in the format of HH:mm. If the end time is the same as the start time, the rule is triggered at any time of a day.

Parameter	Mandatory	Type	Description
days_of_week	No	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. The date in the week list indicates the start date.

Table 1-353 RuleAction

Parameter	Mandatory	Type	Description
type	Yes	String	**Parameter description**: rule action type. Device-side rules support only **DEVICE_CMD**. **Options**: • **DEVICE_CMD**: delivering a device command message. • **SMN_FORWARDING**: sending an SMN message. • **DEVICE_ALARM**: reporting a device alarm. This type can only be selected when **condition** contains **DEVICE_DATA**. The action of this type must be unique.
device_command	No	**ActionDeviceCommand** object	Content of the device command message to deliver. This parameter is mandatory when **type** is set to **DEVICE_CMD**.
smn_forwarding	No	**ActionSmnForwarding** object	Structure of the SMN message to send. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **SMN_FORWARDING**.

Parameter	Mandatory	Type	Description
device_alarm	No	**ActionDeviceAlarm** object	Content of the device alarm message to report. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_ALARM**.

Table 1-354 ActionDeviceCommand

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: ID of the device that the command is delivered to. • When a device data rule is created, if **device_id** is empty, the command is delivered to the device for which the rule is triggered. • This parameter cannot be left empty when a scheduled rule is created. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
cmd	Yes	**CMD** object	**Parameter description**: command to deliver.

Table 1-355 CMD

Parameter	Mandatory	Type	Description
command_name	Yes	String	**Parameter description**: command name, which is defined in the product model associated with the device. Maximum: **128**

Parameter	Mandatory	Type	Description
command_body	Yes	Object	**Parameter description**: command parameter in JSON format. An example command delivered to an LwM2M device is **{"value":"1"}**, which is a key-value pair. Each key is a **paraName** parameter in commands in the product model. An example command delivered to an MQTT device is **{"header": {"mode": "ACK","from": "/users/testUser","method": "SET_TEMPERATURE_READ_PERIOD","to":"/devices/{device_id}/services/{service_id}"},"body": {"value" : "1"}}**. • **mode**: indicates whether the device needs to reply an acknowledgment message after receiving a command. The default value is **ACK**. This parameter is mandatory. **ACK** indicates that an acknowledgment message must be replied. **NOACK** indicates that no acknowledgment message is required. Other values are invalid. • **from**: indicates the address of the command sender. This parameter is optional. The formats of requests initiated by the application, application server, and IoT platform are /users/{*userId*}, /{*serviceName*}, and /cloud/{*serviceName*}, respectively. • **to**: indicates the address of the command receiver. The value is in the format of /devices/{*device_id*}/services/{*service_id*}. This parameter is optional.

Parameter	Mandatory	Type	Description
			• **method**: indicates the command name defined in the product model. This parameter is optional. • **body**: indicates the message body of the command. The value consists of key-value pairs. Each key is a **paraName** parameter in commands in the product model. This parameter optional. The specific format depends on the application and device.
service_id	Yes	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. Maximum: **64**
buffer_timeout	No	Integer	**Parameter description**: duration for the platform to cache a device command before delivering it to the device, in seconds. After the duration elapses, the platform does not deliver this command. The default value is **172800** (48 hours). If **buffer_timeout** is set to **0**, the command is delivered to the device immediately regardless of the command delivery mode set on the platform. Minimum: **0** Maximum: **31536000** Default: **172800**

Parameter	Mandatory	Type	Description
response_timeout	No	Integer	**Parameter description**: validity period of a command response, in seconds. If the platform does not receive a response to a command within the time specified by **response_timeout**, the command times out. The default value is **1800**. Minimum: **1** Maximum: **31536000** Default: **1800**
mode	No	String	**Parameter description**: mode in which device commands are delivered. This parameter is valid only when the value of **buffer_timeout** is greater than **0**. • **ACTIVE**: The platform directly delivers commands to devices. • **PASSIVE**: After creating a device command, the platform caches the command. When the device goes online or the execution result of the previous command is reported, the platform delivers this command.

Table 1-356 ActionSmnForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the SMN service is deployed. Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the SMN service belongs.
theme_name	Yes	String	**Parameter description**: SMN topic name. Maximum: **256**

Parameter	Mandatory	Type	Description
topic_urn	Yes	String	**Parameter description**: SMN topic URN. Maximum: **256**
message_content	No	String	**Parameter description**: SMS or email content. Maximum: **1024**
message_template_name	No	String	**Parameter description**: SMN template name.
message_title	Yes	String	**Parameter description**: SMS or email topic. The value is UTF-8 encoded and cannot exceed 521 bytes.

Table 1-357 ActionDeviceAlarm

Parameter	Mandatory	Type	Description
name	Yes	String	**Parameter description**: alarm name. Maximum: **256**
alarm_status	Yes	String	**Parameter description**: alarm status. **Options**: • **fault**: The alarm is reported. • **recovery**: The alarm is cleared.
severity	Yes	String	**Parameter description**: alarm severity. **Options**: **warning**, **minor**, **major**, and **critical**.
dimension	No	String	**Parameter description**: Alarm dimension, which is used together with the alarm name and alarm severity to identify an alarm. Alarms of the user dimension are used by default. Alarms of the device and resource space dimensions are supported. **Options**: • device: device dimension. • app: resource space dimension.

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: alarm description. Maximum: **256**

Table 1-358 DeviceSide

Parameter	Mandatory	Type	Description
device_ids	No	Array of strings	** Parameter description**: IDs of target devices to which the device-side rule is delivered. A unique device ID is allocated by the platform during device registration. Maximum: **128**

Response Parameters

Status code: 201

Table 1-359 Response body parameters

Parameter	Type	Description
rule_id	String	Rule ID. Maximum: **128**
name	String	Rule name. Minimum: **1** Maximum: **128**
description	String	Rule description. Maximum: **256**
condition_group	**ConditionGroup** object	Condition group of a rule, including simple and complex rules.
actions	Array of **RuleAction** objects	Action list of the rule. A maximum of 10 actions can be configured for a rule.
rule_type	String	Rule type. • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.

Parameter	Type	Description
status	String	Rule status. The default value is **active**. • **active**: activated. • **inactive**: not activated.
app_id	String	Resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the rule to create will belong. If this parameter is not specified, the rule to create will belong to the **default resource space**.
edge_node_ids	Array of strings	List of edge node IDs.
last_update_time	String	UTC time when the rule was last updated. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
device_side	**DeviceSide** object	**Parameter description**: information of devices to which the device-side rule is delivered. This parameter is mandatory when **rule_type** is set to **DEVICE_SIDE**.

Table 1-360 ConditionGroup

Parameter	Type	Description
conditions	Array of **RuleCondition** objects	**Parameter description**: condition list of the rule. A maximum of 10 conditions can be configured for a rule.
logic	String	**Parameter description**: logical relationship between multiple conditions of the rule. The default value is **and**. **Options**: • **and**: All of the conditions are judged. • **or**: One of the conditions is judged.
time_range	**TimeRange** object	**Parameter description**: validity period of a condition.

Table 1-361 RuleCondition

Parameter	Type	Description
type	String	**Parameter description**: type of the rule condition. **Options**: • **DEVICE_DATA**: device property data condition. • **SIMPLE_TIMER**: simple timing condition. • **DAILY_TIMER**: daily scheduled condition. • **DEVICE_LINKAGE_STATUS**: device status condition.
device_property_condition	**DeviceDataCondition** object	**Parameter description**: details of the device data condition. This parameter is mandatory when **type** is set to **DEVICE_DATA**.
simple_timer_condition	**SimpleTimerType** object	**Parameter description**: details of the simple timing condition. This parameter is mandatory when **type** is set to **SIMPLE_TIMER**.
daily_timer_condition	**DailyTimerType** object	**Parameter description**: details of the daily scheduled condition. This parameter is mandatory when **type** is set to **DAILY_TIMER**.
device_linkage_status_condition	**DeviceLinkageStatusCondition** object	**Parameter description**: details of the device status condition. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_LINKAGE_STATUS**.

Table 1-362 DeviceDataCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device property triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device property triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
filters	Array of **PropertyFilter** objects	Data filter criteria.

Table 1-363 PropertyFilter

Parameter	Type	Description
path	String	**Parameter description**: path of the device properties. The value is in the format of *[service_id]/DataProperty*. For example, the path of the door sensor status is **DoorWindow/status**. Maximum: **128**
operator	String	**Parameter description**: data comparison operator. **Options**: **>**, **<**, **>=**, **<=**, **=**, **in** (match in specified values), and **between** (value range).
value	String	**Parameter description**: rvalue of the data comparison expression. When this parameter is used together with the data comparison operator **between**, the rvalue indicates the minimum value and maximum value, which are separated by commas (,). For example, **20,30** indicates that the value is greater than or equal to 20 and less than 30. Maximum: **64**
in_values	Array of strings	**Parameter description**: When **operator** is set to **in**, this parameter is used to transfer the rvalue of the comparison expression. Up to 20 values are allowed. Maximum: **128**

Parameter	Type	Description
strategy	**Strategy** object	**Parameter description**: rule triggering policy, which is used to determine whether the last data meets the conditions. If **rule_type** is set to **DEVICE_LINKAGE**, either this parameter must be specified. This parameter is not supported by device-side rules.

Table 1-364 Strategy

Parameter	Type	Description
trigger	String	**Parameter description**: rule triggering judgment policy. The default value is **pulse**. **Options**: • **pulse**: If the reported data meets conditions, the rule is triggered and the previous data is not judged. • **reverse**: When the data reported this time meets the conditions, the rule is triggered even if the data reported last time does not meet the conditions.
event_valid_time	Integer	**Parameter description**: validity period of the device data, in seconds. The time when the device data is generated is the same as the value of **eventTime** in the reported data. Minimum: **-1**

Table 1-365 SimpleTimerType

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The UTC time is used. The value is in the format of yyyyMMdd'T'HHmmss'Z'. Maximum: **128**
repeat_interval	Integer	**Parameter description**: interval for triggering the rule, in seconds. Minimum: **1** Maximum: **31536000**
repeat_count	Integer	**Parameter description**: number of times that a rule is triggered. Minimum: **1** Maximum: **9999**

Table 1-366 DailyTimerType

Parameter	Type	Description
time	String	**Parameter description**: time when a rule is triggered. The value is in the format of HH:MM. Maximum: **128**
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. Maximum: **128**

Table 1-367 DeviceLinkageStatusCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device status triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device status triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
status_list	Array of strings	**Parameter description**: status list. This parameter must be carried for a device status condition. **Options**: • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. Maximum: **128**

Parameter	Type	Description
duration	Integer	**Duration**: device status duration. Value range: 0–60 (minutes). Minimum: **0** Maximum: **60** Default: **0**

Table 1-368 TimeRange

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The value is in the format of HH:mm.
end_time	String	**Parameter description**: end time for rule triggering. The value is in the format of HH:mm. If the end time is the same as the start time, the rule is triggered at any time of a day.
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. The date in the week list indicates the start date.

Table 1-369 RuleAction

Parameter	Type	Description
type	String	**Parameter description**: rule action type. Device-side rules support only **DEVICE_CMD**. **Options**: • **DEVICE_CMD**: delivering a device command message. • **SMN_FORWARDING**: sending an SMN message. • **DEVICE_ALARM**: reporting a device alarm. This type can only be selected when **condition** contains **DEVICE_DATA**. The action of this type must be unique.
device_command	**ActionDeviceCommand** object	Content of the device command message to deliver. This parameter is mandatory when **type** is set to **DEVICE_CMD**.

Parameter	Type	Description
smn_forwarding	**ActionSmnForwarding** object	Structure of the SMN message to send. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **SMN_FORWARDING**.
device_alarm	**ActionDeviceAlarm** object	Content of the device alarm message to report. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_ALARM**.

Table 1-370 ActionDeviceCommand

Parameter	Type	Description
device_id	String	**Parameter description**: ID of the device that the command is delivered to. • When a device data rule is created, if **device_id** is empty, the command is delivered to the device for which the rule is triggered. • This parameter cannot be left empty when a scheduled rule is created. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
cmd	**CMD** object	**Parameter description**: command to deliver.

Table 1-371 CMD

Parameter	Type	Description
command_name	String	**Parameter description**: command name, which is defined in the product model associated with the device. Maximum: **128**

Parameter	Type	Description
command_body	Object	**Parameter description**: command parameter in JSON format. An example command delivered to an LwM2M device is **{"value":"1"}**, which is a key-value pair. Each key is a **paraName** parameter in commands in the product model. An example command delivered to an MQTT device is **{"header": {"mode": "ACK","from": "/users/testUser","method": "SET_TEMPERATURE_READ_PERIOD","to":"/devices/{device_id}/services/{service_id}"},"body": {"value" : "1"}}**. • **mode**: indicates whether the device needs to reply an acknowledgment message after receiving a command. The default value is **ACK**. This parameter is mandatory. **ACK** indicates that an acknowledgment message must be replied. **NOACK** indicates that no acknowledgment message is required. Other values are invalid. • **from**: indicates the address of the command sender. This parameter is optional. The formats of requests initiated by the application, application server, and IoT platform are /users/{*userId*}, /{*serviceName*}, and /cloud/{*serviceName*}, respectively. • **to**: indicates the address of the command receiver. The value is in the format of /devices/{*device_id*}/services/{*service_id*}. This parameter is optional. • **method**: indicates the command name defined in the product model. This parameter is optional. • **body**: indicates the message body of the command. The value consists of key-value pairs. Each key is a **paraName** parameter in commands in the product model. This parameter optional. The specific format depends on the application and device.
service_id	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. Maximum: **64**

Parameter	Type	Description
buffer_timeout	Integer	**Parameter description**: duration for the platform to cache a device command before delivering it to the device, in seconds. After the duration elapses, the platform does not deliver this command. The default value is **172800** (48 hours). If **buffer_timeout** is set to **0**, the command is delivered to the device immediately regardless of the command delivery mode set on the platform. Minimum: **0** Maximum: **31536000** Default: **172800**
response_timeout	Integer	**Parameter description**: validity period of a command response, in seconds. If the platform does not receive a response to a command within the time specified by **response_timeout**, the command times out. The default value is **1800**. Minimum: **1** Maximum: **31536000** Default: **1800**
mode	String	**Parameter description**: mode in which device commands are delivered. This parameter is valid only when the value of **buffer_timeout** is greater than **0**. • **ACTIVE**: The platform directly delivers commands to devices. • **PASSIVE**: After creating a device command, the platform caches the command. When the device goes online or the execution result of the previous command is reported, the platform delivers this command.

Table 1-372 ActionSmnForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the SMN service is deployed. Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the SMN service belongs.

Parameter	Type	Description
theme_name	String	**Parameter description**: SMN topic name. Maximum: **256**
topic_urn	String	**Parameter description**: SMN topic URN. Maximum: **256**
message_content	String	**Parameter description**: SMS or email content. Maximum: **1024**
message_template_name	String	**Parameter description**: SMN template name.
message_title	String	**Parameter description**: SMS or email topic. The value is UTF-8 encoded and cannot exceed 521 bytes.

Table 1-373 ActionDeviceAlarm

Parameter	Type	Description
name	String	**Parameter description**: alarm name. Maximum: **256**
alarm_status	String	**Parameter description**: alarm status. **Options**: • **fault**: The alarm is reported. • **recovery**: The alarm is cleared.
severity	String	**Parameter description**: alarm severity. **Options**: **warning**, **minor**, **major**, and **critical**.
dimension	String	**Parameter description**: Alarm dimension, which is used together with the alarm name and alarm severity to identify an alarm. Alarms of the user dimension are used by default. Alarms of the device and resource space dimensions are supported. **Options**: • device: device dimension. • app: resource space dimension.
description	String	**Parameter description**: alarm description. Maximum: **256**

Table 1-374 DeviceSide

Parameter	Type	Description
device_ids	Array of strings	** Parameter description**: IDs of target devices to which the device-side rule is delivered. A unique device ID is allocated by the platform during device registration. Maximum: **128**

Example Requests

- Creates a cloud rule. An alarm is triggered when a device goes offline.

```
POST https://{endpoint}/v5/iot/{project_id}/rules

{
  "name" : "openLight",
  "description" : "string",
  "condition_group" : {
    "time_range" : {
      "days_of_week" : "2,3,4,5,6",
      "start_time" : "18:00",
      "end_time" : "23:00"
    },
    "logic" : "or",
    "conditions" : [ {
      "device_linkage_status_condition" : {
        "device_id" : "07b69d78-c716-4be6-9545-869920738397",
        "status_list" : [ "OFFLINE" ],
        "duration" : 0
      },
      "type" : "DEVICE_LINKAGE_STATUS"
    } ]
  },
  "actions" : [ {
    "device_alarm" : {
      "severity" : "warning",
      "alarm_status" : "fault",
      "name" : "A device goes offline.",
      "description" : "A device goes offline."
    },
    "type" : "DEVICE_ALARM"
  } ],
  "rule_type" : "DEVICE_LINKAGE",
  "status" : "active",
  "app_id" : "string"
}
```

- Creates a cloud rule. Property reporting triggers command delivery.

```
POST https://{endpoint}/v5/iot/{project_id}/rules

{
  "name" : "openLight",
  "description" : "string",
  "condition_group" : {
    "time_range" : {
      "days_of_week" : "2,3,4,5,6",
      "start_time" : "18:00",
      "end_time" : "23:00"
    },
    "logic" : "or",
    "conditions" : [ {
      "device_property_condition" : {
        "device_id" : "07b69d78-c716-4be6-9545-869920738397",
```

```
      "filters" : [ {
        "path" : "StreetLight/visibility",
        "strategy" : {
          "trigger" : "reverse",
          "event_valid_time" : 300
        },
        "value" : "30",
        "operator" : "<"
      } ]
    },
    "type" : "DEVICE_DATA"
  } ]
},
"actions" : [ {
  "device_command" : {
    "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
    "cmd" : {
      "buffer_timeout" : 0,
      "mode" : "ACTIVE",
      "service_id" : "Switch",
      "command_name" : "SET_LIGHT_SWITCH",
      "command_body" : {
        "value" : 0
      }
    }
  },
  "type" : "DEVICE_CMD"
} ],
"rule_type" : "DEVICE_LINKAGE",
"status" : "active",
"app_id" : "string"
}
```

- Creates a device-side rule. Property reporting triggers command delivery.

```
POST https://{endpoint}/v5/iot/{project_id}/rules

{
  "name" : "openLight",
  "description" : "string",
  "condition_group" : {
    "time_range" : {
      "days_of_week" : "2,3,4,5,6",
      "start_time" : "18:00",
      "end_time" : "23:00"
    },
    "logic" : "or",
    "conditions" : [ {
      "device_property_condition" : {
        "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
        "filters" : [ {
          "path" : "StreetLight/visibility",
          "strategy" : {
            "trigger" : "reverse",
            "event_valid_time" : 300
          },
          "value" : "30",
          "operator" : "<"
        } ]
      },
      "type" : "DEVICE_DATA"
    } ]
  },
  "actions" : [ {
    "device_command" : {
      "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
      "cmd" : {
        "buffer_timeout" : 0,
        "mode" : "ACTIVE",
        "service_id" : "Switch",
        "command_name" : "SET_LIGHT_SWITCH",
```

```
      "command_body" : {
        "value" : 0
      }
    }
  },
  "type" : "DEVICE_CMD"
} ],
"rule_type" : "DEVICE_SIDE",
"device_side" : {
  "device_ids" : [ "3a9e52d9-3ebf-4985-89e9-6d2396748a2f" ]
},
"status" : "active",
"app_id" : "string"
}
```

Example Responses

Status code: 201

Created

```
{
  "rule_id" : "5eb3628d017d9105d0cf9aec",
  "rule_type" : "DEVICE_LINKAGE",
  "name" : "openLight",
  "status" : "active",
  "condition_group" : {
    "conditions" : [ {
      "type" : "DEVICE_DATA",
      "device_property_condition" : {
        "device_id" : "07b69d78-c716-4be6-9545-869920738397",
        "filters" : [ {
          "path" : "StreetLight/visibility",
          "operator" : "<",
          "value" : "30",
          "strategy" : {
            "trigger" : "reverse",
            "event_valid_time" : 300
          }
        } ]
      }
    } ]
  },
  "actions" : [ {
    "type" : "DEVICE_CMD",
    "device_command" : {
      "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
      "cmd" : {
        "command_name" : "SET_LIGHT_SWITCH",
        "command_body" : {
          "value" : 0
        },
        "service_id" : "Switch",
        "buffer_timeout" : 0,
        "response_timeout" : 1800
      }
    }
  } ]
}
```

Status Codes

Status Code	Description
201	Created

Status Code	Description
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.10.2 Query the Rule List

Function

This API is used by an application to query the rule list on the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/rules

Table 1-375 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-376 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can specify this parameter to query rules in the specified resource space. If this parameter is not carried, rules in all resource spaces are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
rule_type	No	String	**Parameter description**: rule type. This parameter is optional. Rules of the specified type will be returned. If this parameter is not specified, all types of rules will be returned. **Options**: • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-377 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-378 Response body parameters

Parameter	Type	Description
marker	String	ID of the last record in this query, which can be used in the next query.
count	Long	Total number of records that meet the query conditions.
rules	Array of **RuleResponse** objects	Rule list.

Table 1-379 RuleResponse

Parameter	Type	Description
rule_id	String	Rule ID. Maximum: **128**
name	String	Rule name. Minimum: **1** Maximum: **128**
description	String	Rule description. Maximum: **256**
condition_group	**ConditionGroup** object	Condition group of a rule, including simple and complex rules.
actions	Array of **RuleAction** objects	Action list of the rule. A maximum of 10 actions can be configured for a rule.
rule_type	String	Rule type. • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.
status	String	Rule status. The default value is **active**. • **active**: activated. • **inactive**: not activated.
app_id	String	Resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the rule to create will belong. If this parameter is not specified, the rule to create will belong to the **default resource space**.
edge_node_ids	Array of strings	List of edge node IDs.
last_update_time	String	UTC time when the rule was last updated. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
device_side	**DeviceSide** object	**Parameter description**: information of devices to which the device-side rule is delivered. This parameter is mandatory when **rule_type** is set to **DEVICE_SIDE**.

Table 1-380 ConditionGroup

Parameter	Type	Description
conditions	Array of **RuleCondition** objects	**Parameter description**: condition list of the rule. A maximum of 10 conditions can be configured for a rule.
logic	String	**Parameter description**: logical relationship between multiple conditions of the rule. The default value is **and**. **Options**: • **and**: All of the conditions are judged. • **or**: One of the conditions is judged.
time_range	**TimeRange** object	**Parameter description**: validity period of a condition.

Table 1-381 RuleCondition

Parameter	Type	Description
type	String	**Parameter description**: type of the rule condition. **Options**: • **DEVICE_DATA**: device property data condition. • **SIMPLE_TIMER**: simple timing condition. • **DAILY_TIMER**: daily scheduled condition. • **DEVICE_LINKAGE_STATUS**: device status condition.
device_property_condition	**DeviceDataCondition** object	**Parameter description**: details of the device data condition. This parameter is mandatory when **type** is set to **DEVICE_DATA**.
simple_timer_condition	**SimpleTimerType** object	**Parameter description**: details of the simple timing condition. This parameter is mandatory when **type** is set to **SIMPLE_TIMER**.
daily_timer_condition	**DailyTimerType** object	**Parameter description**: details of the daily scheduled condition. This parameter is mandatory when **type** is set to **DAILY_TIMER**.
device_linkage_status_condition	**DeviceLinkageStatusCondition** object	**Parameter description**: details of the device status condition. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_LINKAGE_STATUS**.

Table 1-382 DeviceDataCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device property triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device property triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
filters	Array of **PropertyFilter** objects	Data filter criteria.

Table 1-383 PropertyFilter

Parameter	Type	Description
path	String	**Parameter description**: path of the device properties. The value is in the format of *[service_id]/DataProperty*. For example, the path of the door sensor status is **DoorWindow/status**. Maximum: **128**
operator	String	**Parameter description**: data comparison operator. **Options**: >, <, >=, <=, =, **in** (match in specified values), and **between** (value range).

Parameter	Type	Description
value	String	**Parameter description**: rvalue of the data comparison expression. When this parameter is used together with the data comparison operator **between**, the rvalue indicates the minimum value and maximum value, which are separated by commas (,). For example, **20,30** indicates that the value is greater than or equal to 20 and less than 30. Maximum: **64**
in_values	Array of strings	**Parameter description**: When **operator** is set to **in**, this parameter is used to transfer the rvalue of the comparison expression. Up to 20 values are allowed. Maximum: **128**
strategy	**Strategy** object	**Parameter description**: rule triggering policy, which is used to determine whether the last data meets the conditions. If **rule_type** is set to **DEVICE_LINKAGE**, either this parameter must be specified. This parameter is not supported by device-side rules.

Table 1-384 Strategy

Parameter	Type	Description
trigger	String	**Parameter description**: rule triggering judgment policy. The default value is **pulse**. **Options**: • **pulse**: If the reported data meets conditions, the rule is triggered and the previous data is not judged. • **reverse**: When the data reported this time meets the conditions, the rule is triggered even if the data reported last time does not meet the conditions.
event_valid_time	Integer	**Parameter description**: validity period of the device data, in seconds. The time when the device data is generated is the same as the value of **eventTime** in the reported data. Minimum: **-1**

Table 1-385 SimpleTimerType

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The UTC time is used. The value is in the format of yyyyMMdd'T'HHmmss'Z'. Maximum: **128**
repeat_interval	Integer	**Parameter description**: interval for triggering the rule, in seconds. Minimum: **1** Maximum: **31536000**
repeat_count	Integer	**Parameter description**: number of times that a rule is triggered. Minimum: **1** Maximum: **9999**

Table 1-386 DailyTimerType

Parameter	Type	Description
time	String	**Parameter description**: time when a rule is triggered. The value is in the format of HH:MM. Maximum: **128**
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. Maximum: **128**

Table 1-387 DeviceLinkageStatusCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device status triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device status triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
status_list	Array of strings	**Parameter description**: status list. This parameter must be carried for a device status condition. **Options**: • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. Maximum: **128**
duration	Integer	**Duration**: device status duration. Value range: 0–60 (minutes). Minimum: **0** Maximum: **60** Default: **0**

Table 1-388 TimeRange

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The value is in the format of HH:mm.

Parameter	Type	Description
end_time	String	**Parameter description**: end time for rule triggering. The value is in the format of HH:mm. If the end time is the same as the start time, the rule is triggered at any time of a day.
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. The date in the week list indicates the start date.

Table 1-389 RuleAction

Parameter	Type	Description
type	String	**Parameter description**: rule action type. Device-side rules support only **DEVICE_CMD**. **Options**: • **DEVICE_CMD**: delivering a device command message. • **SMN_FORWARDING**: sending an SMN message. • **DEVICE_ALARM**: reporting a device alarm. This type can only be selected when **condition** contains **DEVICE_DATA**. The action of this type must be unique.
device_command	**ActionDeviceCommand** object	Content of the device command message to deliver. This parameter is mandatory when **type** is set to **DEVICE_CMD**.
smn_forwarding	**ActionSmnForwarding** object	Structure of the SMN message to send. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **SMN_FORWARDING**.
device_alarm	**ActionDeviceAlarm** object	Content of the device alarm message to report. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_ALARM**.

Table 1-390 ActionDeviceCommand

Parameter	Type	Description
device_id	String	**Parameter description**: ID of the device that the command is delivered to. • When a device data rule is created, if **device_id** is empty, the command is delivered to the device for which the rule is triggered. • This parameter cannot be left empty when a scheduled rule is created. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
cmd	**CMD** object	**Parameter description**: command to deliver.

Table 1-391 CMD

Parameter	Type	Description
command_name	String	**Parameter description**: command name, which is defined in the product model associated with the device. Maximum: **128**

Parameter	Type	Description
command_body	Object	**Parameter description**: command parameter in JSON format. An example command delivered to an LwM2M device is **{"value":"1"}**, which is a key-value pair. Each key is a **paraName** parameter in commands in the product model. An example command delivered to an MQTT device is **{"header": {"mode": "ACK","from": "/users/testUser","method": "SET_TEMPERATURE_READ_PERIOD","to":"/devices/{device_id}/services/{service_id}"},"body": {"value" : "1"}}**. • **mode**: indicates whether the device needs to reply an acknowledgment message after receiving a command. The default value is **ACK**. This parameter is mandatory. **ACK** indicates that an acknowledgment message must be replied. **NOACK** indicates that no acknowledgment message is required. Other values are invalid. • **from**: indicates the address of the command sender. This parameter is optional. The formats of requests initiated by the application, application server, and IoT platform are /users/{*userId*}, /{*serviceName*}, and /cloud/{*serviceName*}, respectively. • **to**: indicates the address of the command receiver. The value is in the format of /devices/{*device_id*}/services/{*service_id*}. This parameter is optional. • **method**: indicates the command name defined in the product model. This parameter is optional. • **body**: indicates the message body of the command. The value consists of key-value pairs. Each key is a **paraName** parameter in commands in the product model. This parameter optional. The specific format depends on the application and device.
service_id	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. Maximum: **64**

Parameter	Type	Description
buffer_timeout	Integer	**Parameter description**: duration for the platform to cache a device command before delivering it to the device, in seconds. After the duration elapses, the platform does not deliver this command. The default value is **172800** (48 hours). If **buffer_timeout** is set to **0**, the command is delivered to the device immediately regardless of the command delivery mode set on the platform. Minimum: **0** Maximum: **31536000** Default: **172800**
response_timeout	Integer	**Parameter description**: validity period of a command response, in seconds. If the platform does not receive a response to a command within the time specified by **response_timeout**, the command times out. The default value is **1800**. Minimum: **1** Maximum: **31536000** Default: **1800**
mode	String	**Parameter description**: mode in which device commands are delivered. This parameter is valid only when the value of **buffer_timeout** is greater than **0**. • **ACTIVE**: The platform directly delivers commands to devices. • **PASSIVE**: After creating a device command, the platform caches the command. When the device goes online or the execution result of the previous command is reported, the platform delivers this command.

Table 1-392 ActionSmnForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the SMN service is deployed. Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the SMN service belongs.

Parameter	Type	Description
theme_name	String	**Parameter description**: SMN topic name. Maximum: **256**
topic_urn	String	**Parameter description**: SMN topic URN. Maximum: **256**
message_content	String	**Parameter description**: SMS or email content. Maximum: **1024**
message_template_name	String	**Parameter description**: SMN template name.
message_title	String	**Parameter description**: SMS or email topic. The value is UTF-8 encoded and cannot exceed 521 bytes.

Table 1-393 ActionDeviceAlarm

Parameter	Type	Description
name	String	**Parameter description**: alarm name. Maximum: **256**
alarm_status	String	**Parameter description**: alarm status. **Options**: • **fault**: The alarm is reported. • **recovery**: The alarm is cleared.
severity	String	**Parameter description**: alarm severity. **Options**: **warning**, **minor**, **major**, and **critical**.
dimension	String	**Parameter description**: Alarm dimension, which is used together with the alarm name and alarm severity to identify an alarm. Alarms of the user dimension are used by default. Alarms of the device and resource space dimensions are supported. **Options**: • device: device dimension. • app: resource space dimension.
description	String	**Parameter description**: alarm description. Maximum: **256**

Table 1-394 DeviceSide

Parameter	Type	Description
device_ids	Array of strings	** Parameter description**: IDs of target devices to which the device-side rule is delivered. A unique device ID is allocated by the platform during device registration. Maximum: **128**

Example Requests

Queries linkage rules of a specified resource space in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/rules?
app_id={app_id}&limit={limit}&marker={marker}&offset={offset}
```

Example Responses

Status code: 200

OK

```
{
  "marker" : "6336a282cda07a01f7ac5d11",
  "count" : 1,
  "rules" : [ {
    "rule_id" : "5eb3628d017d9105d0cf9aec",
    "name" : "openLight",
    "condition_group" : {
      "conditions" : [ {
        "type" : "DEVICE_DATA",
        "device_property_condition" : {
          "device_id" : "07b69d78-c716-4be6-9545-869920738397",
          "filters" : [ {
            "path" : "StreetLight/visibility",
            "operator" : "<",
            "value" : "30",
            "strategy" : {
              "trigger" : "reverse",
              "event_valid_time" : 300
            }
          } ]
        }
      } ],
      "logic" : "and"
    },
    "actions" : [ {
      "type" : "DEVICE_CMD",
      "device_command" : {
        "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
        "cmd" : {
          "command_name" : "SET_LIGHT_SWITCH",
          "command_body" : {
            "value" : 0
          },
          "service_id" : "Switch",
          "buffer_timeout" : 0,
          "response_timeout" : 1800
        }
      }
    } ],
    "rule_type" : "DEVICE_LINKAGE",
```



```
    "status" : "active",
    "app_id" : "9562bf8541e44361b6ae3a7e9fbe1144",
    "last_update_time" : "20221017T023727Z"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.10.3 Modify a Rule

Function

This API is used by an application to modify the configuration of a specific rule on the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/rules/{rule_id}

Table 1-395 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
rule_id	Yes	String	**Parameter description**: unique rule ID, which is allocated by the platform during rule creation. **Value**: The value can contain a maximum of 32 characters. Only letters and digits are allowed.

Request Parameters

Table 1-396 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-397 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	**Parameter description**: rule name. Minimum: **1** Maximum: **128**
description	No	String	**Parameter description**: rule description. Maximum: **256**
condition_group	Yes	**ConditionGroup** object	**Parameter description**: condition group of a rule, including simple and complex rules.
actions	Yes	Array of **RuleAction** objects	**Parameter description**: action list of the rule. A maximum of 10 actions can be configured for a rule.
rule_type	Yes	String	**Parameter description**: rule type. **Options**: • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.
status	No	String	**Parameter description**: rule status. The default value is **active**. **Options**: • **active**: activated. • **inactive**: not activated.
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the rule to create will belong. If this parameter is not specified, the rule to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
device_side	No	DeviceSide object	**Parameter description**: information of devices to which the device-side rule is delivered. This parameter is mandatory when **rule_type** is set to **DEVICE_SIDE**.

Table 1-398 ConditionGroup

Parameter	Mandatory	Type	Description
conditions	No	Array of RuleCondition objects	**Parameter description**: condition list of the rule. A maximum of 10 conditions can be configured for a rule.
logic	No	String	**Parameter description**: logical relationship between multiple conditions of the rule. The default value is **and**. **Options**: • **and**: All of the conditions are judged. • **or**: One of the conditions is judged.
time_range	No	TimeRange object	**Parameter description**: validity period of a condition.

Table 1-399 RuleCondition

Parameter	Mandatory	Type	Description
type	Yes	String	**Parameter description**: type of the rule condition. **Options**: • **DEVICE_DATA**: device property data condition. • **SIMPLE_TIMER**: simple timing condition. • **DAILY_TIMER**: daily scheduled condition. • **DEVICE_LINKAGE_STATUS**: device status condition.

Parameter	Mandatory	Type	Description
device_property_condition	No	**DeviceDataCondition** object	**Parameter description**: details of the device data condition. This parameter is mandatory when **type** is set to **DEVICE_DATA**.
simple_timer_condition	No	**SimpleTimerType** object	**Parameter description**: details of the simple timing condition. This parameter is mandatory when **type** is set to **SIMPLE_TIMER**.
daily_timer_condition	No	**DailyTimerType** object	**Parameter description**: details of the daily scheduled condition. This parameter is mandatory when **type** is set to **DAILY_TIMER**.
device_linkage_status_condition	No	**DeviceLinkageStatusCondition** object	**Parameter description**: details of the device status condition. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_LINKAGE_STATUS**.

Table 1-400 DeviceDataCondition

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device property triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device property triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
filters	No	Array of **PropertyFilter** objects	Data filter criteria.

Table 1-401 PropertyFilter

Parameter	Mandatory	Type	Description
path	Yes	String	**Parameter description**: path of the device properties. The value is in the format of *[service_id]/DataProperty*. For example, the path of the door sensor status is **DoorWindow/status**. Maximum: **128**
operator	Yes	String	**Parameter description**: data comparison operator. **Options**: **>**, **<**, **>=**, **<=**, **=**, **in** (match in specified values), and **between** (value range).

Parameter	Mandatory	Type	Description
value	No	String	**Parameter description**: rvalue of the data comparison expression. When this parameter is used together with the data comparison operator **between**, the rvalue indicates the minimum value and maximum value, which are separated by commas (,). For example, **20,30** indicates that the value is greater than or equal to 20 and less than 30. Maximum: **64**
in_values	No	Array of strings	**Parameter description**: When **operator** is set to **in**, this parameter is used to transfer the rvalue of the comparison expression. Up to 20 values are allowed. Maximum: **128**
strategy	No	**Strategy** object	**Parameter description**: rule triggering policy, which is used to determine whether the last data meets the conditions. If **rule_type** is set to **DEVICE_LINKAGE**, either this parameter must be specified. This parameter is not supported by device-side rules.

Table 1-402 Strategy

Parameter	Mandatory	Type	Description
trigger	No	String	**Parameter description**: rule triggering judgment policy. The default value is **pulse**. **Options**: • **pulse**: If the reported data meets conditions, the rule is triggered and the previous data is not judged. • **reverse**: When the data reported this time meets the conditions, the rule is triggered even if the data reported last time does not meet the conditions.
event_valid_time	No	Integer	**Parameter description**: validity period of the device data, in seconds. The time when the device data is generated is the same as the value of **eventTime** in the reported data. Minimum: **-1**

Table 1-403 SimpleTimerType

Parameter	Mandatory	Type	Description
start_time	Yes	String	**Parameter description**: start time for rule triggering. The UTC time is used. The value is in the format of yyyyMMdd'T'HHmmss'Z'. Maximum: **128**
repeat_interval	Yes	Integer	**Parameter description**: interval for triggering the rule, in seconds. Minimum: **1** Maximum: **31536000**
repeat_count	Yes	Integer	**Parameter description**: number of times that a rule is triggered. Minimum: **1** Maximum: **9999**

Table 1-404 DailyTimerType

Parameter	Mandatory	Type	Description
time	Yes	String	**Parameter description**: time when a rule is triggered. The value is in the format of HH:MM. Maximum: **128**
days_of_week	No	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. Maximum: **128**

Table 1-405 DeviceLinkageStatusCondition

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device status triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device status triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
status_list	No	Array of strings	**Parameter description**: status list. This parameter must be carried for a device status condition. **Options**: • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. Maximum: **128**
duration	No	Integer	**Duration**: device status duration. Value range: 0–60 (minutes). Minimum: **0** Maximum: **60** Default: **0**

Table 1-406 TimeRange

Parameter	Mandatory	Type	Description
start_time	Yes	String	**Parameter description**: start time for rule triggering. The value is in the format of HH:mm.
end_time	Yes	String	**Parameter description**: end time for rule triggering. The value is in the format of HH:mm. If the end time is the same as the start time, the rule is triggered at any time of a day.

Parameter	Mandatory	Type	Description
days_of_week	No	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. The date in the week list indicates the start date.

Table 1-407 RuleAction

Parameter	Mandatory	Type	Description
type	Yes	String	**Parameter description**: rule action type. Device-side rules support only **DEVICE_CMD**. **Options**: • **DEVICE_CMD**: delivering a device command message. • **SMN_FORWARDING**: sending an SMN message. • **DEVICE_ALARM**: reporting a device alarm. This type can only be selected when **condition** contains **DEVICE_DATA**. The action of this type must be unique.
device_command	No	**ActionDeviceCommand** object	Content of the device command message to deliver. This parameter is mandatory when **type** is set to **DEVICE_CMD**.
smn_forwarding	No	**ActionSmnForwarding** object	Structure of the SMN message to send. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **SMN_FORWARDING**.

Parameter	Mandatory	Type	Description
device_alarm	No	**ActionDeviceAlarm** object	Content of the device alarm message to report. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_ALARM**.

Table 1-408 ActionDeviceCommand

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: ID of the device that the command is delivered to. • When a device data rule is created, if **device_id** is empty, the command is delivered to the device for which the rule is triggered. • This parameter cannot be left empty when a scheduled rule is created. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
cmd	Yes	**CMD** object	**Parameter description**: command to deliver.

Table 1-409 CMD

Parameter	Mandatory	Type	Description
command_name	Yes	String	**Parameter description**: command name, which is defined in the product model associated with the device. Maximum: **128**

Parameter	Mandatory	Type	Description
command_body	Yes	Object	**Parameter description**: command parameter in JSON format. An example command delivered to an LwM2M device is **{"value":"1"}**, which is a key-value pair. Each key is a **paraName** parameter in commands in the product model. An example command delivered to an MQTT device is **{"header": {"mode": "ACK","from": "/users/testUser","method": "SET_TEMPERATURE_READ_PERIOD","to":"/devices/{device_id}/services/{service_id}"},"body": {"value" : "1"}}**. • **mode**: indicates whether the device needs to reply an acknowledgment message after receiving a command. The default value is **ACK**. This parameter is mandatory. **ACK** indicates that an acknowledgment message must be replied. **NOACK** indicates that no acknowledgment message is required. Other values are invalid. • **from**: indicates the address of the command sender. This parameter is optional. The formats of requests initiated by the application, application server, and IoT platform are /users/{*userId*}, /{*serviceName*}, and /cloud/{*serviceName*}, respectively. • **to**: indicates the address of the command receiver. The value is in the format of /devices/{*device_id*}/services/{*service_id*}. This parameter is optional.

Parameter	Mandatory	Type	Description
			• **method**: indicates the command name defined in the product model. This parameter is optional. • **body**: indicates the message body of the command. The value consists of key-value pairs. Each key is a **paraName** parameter in commands in the product model. This parameter optional. The specific format depends on the application and device.
service_id	Yes	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. Maximum: **64**
buffer_timeout	No	Integer	**Parameter description**: duration for the platform to cache a device command before delivering it to the device, in seconds. After the duration elapses, the platform does not deliver this command. The default value is **172800** (48 hours). If **buffer_timeout** is set to **0**, the command is delivered to the device immediately regardless of the command delivery mode set on the platform. Minimum: **0** Maximum: **31536000** Default: **172800**

Parameter	Mandatory	Type	Description
response_timeout	No	Integer	**Parameter description**: validity period of a command response, in seconds. If the platform does not receive a response to a command within the time specified by **response_timeout**, the command times out. The default value is **1800**. Minimum: **1** Maximum: **31536000** Default: **1800**
mode	No	String	**Parameter description**: mode in which device commands are delivered. This parameter is valid only when the value of **buffer_timeout** is greater than **0**. • **ACTIVE**: The platform directly delivers commands to devices. • **PASSIVE**: After creating a device command, the platform caches the command. When the device goes online or the execution result of the previous command is reported, the platform delivers this command.

Table 1-410 ActionSmnForwarding

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: region where the SMN service is deployed. Maximum: **256**
project_id	Yes	String	**Parameter description**: ID of the project to which the SMN service belongs.
theme_name	Yes	String	**Parameter description**: SMN topic name. Maximum: **256**

Parameter	Mandatory	Type	Description
topic_urn	Yes	String	**Parameter description**: SMN topic URN. Maximum: **256**
message_content	No	String	**Parameter description**: SMS or email content. Maximum: **1024**
message_template_name	No	String	**Parameter description**: SMN template name.
message_title	Yes	String	**Parameter description**: SMS or email topic. The value is UTF-8 encoded and cannot exceed 521 bytes.

Table 1-411 ActionDeviceAlarm

Parameter	Mandatory	Type	Description
name	Yes	String	**Parameter description**: alarm name. Maximum: **256**
alarm_status	Yes	String	**Parameter description**: alarm status. **Options**: • **fault**: The alarm is reported. • **recovery**: The alarm is cleared.
severity	Yes	String	**Parameter description**: alarm severity. **Options**: **warning**, **minor**, **major**, and **critical**.
dimension	No	String	**Parameter description**: Alarm dimension, which is used together with the alarm name and alarm severity to identify an alarm. Alarms of the user dimension are used by default. Alarms of the device and resource space dimensions are supported. **Options**: • device: device dimension. • app: resource space dimension.

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: alarm description. Maximum: **256**

Table 1-412 DeviceSide

Parameter	Mandatory	Type	Description
device_ids	No	Array of strings	** Parameter description**: IDs of target devices to which the device-side rule is delivered. A unique device ID is allocated by the platform during device registration. Maximum: **128**

Response Parameters

Status code: 200

Table 1-413 Response body parameters

Parameter	Type	Description
rule_id	String	Rule ID. Maximum: **128**
name	String	Rule name. Minimum: **1** Maximum: **128**
description	String	Rule description. Maximum: **256**
condition_group	**ConditionGroup** object	Condition group of a rule, including simple and complex rules.
actions	Array of **RuleAction** objects	Action list of the rule. A maximum of 10 actions can be configured for a rule.
rule_type	String	Rule type. • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.

Parameter	Type	Description
status	String	Rule status. The default value is **active**. • **active**: activated. • **inactive**: not activated.
app_id	String	Resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the rule to create will belong. If this parameter is not specified, the rule to create will belong to the **default resource space**.
edge_node_ids	Array of strings	List of edge node IDs.
last_update_time	String	UTC time when the rule was last updated. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
device_side	**DeviceSide** object	**Parameter description**: information of devices to which the device-side rule is delivered. This parameter is mandatory when **rule_type** is set to **DEVICE_SIDE**.

Table 1-414 ConditionGroup

Parameter	Type	Description
conditions	Array of **RuleCondition** objects	**Parameter description**: condition list of the rule. A maximum of 10 conditions can be configured for a rule.
logic	String	**Parameter description**: logical relationship between multiple conditions of the rule. The default value is **and**. **Options**: • **and**: All of the conditions are judged. • **or**: One of the conditions is judged.
time_range	**TimeRange** object	**Parameter description**: validity period of a condition.

Table 1-415 RuleCondition

Parameter	Type	Description
type	String	**Parameter description**: type of the rule condition. **Options**: • **DEVICE_DATA**: device property data condition. • **SIMPLE_TIMER**: simple timing condition. • **DAILY_TIMER**: daily scheduled condition. • **DEVICE_LINKAGE_STATUS**: device status condition.
device_property_condition	**DeviceDataCondition** object	**Parameter description**: details of the device data condition. This parameter is mandatory when **type** is set to **DEVICE_DATA**.
simple_timer_condition	**SimpleTimerType** object	**Parameter description**: details of the simple timing condition. This parameter is mandatory when **type** is set to **SIMPLE_TIMER**.
daily_timer_condition	**DailyTimerType** object	**Parameter description**: details of the daily scheduled condition. This parameter is mandatory when **type** is set to **DAILY_TIMER**.
device_linkage_status_condition	**DeviceLinkageStatusCondition** object	**Parameter description**: details of the device status condition. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_LINKAGE_STATUS**.

Table 1-416 DeviceDataCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device property triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device property triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
filters	Array of **PropertyFilter** objects	Data filter criteria.

Table 1-417 PropertyFilter

Parameter	Type	Description
path	String	**Parameter description**: path of the device properties. The value is in the format of *[service_id]/DataProperty*. For example, the path of the door sensor status is **DoorWindow/status**. Maximum: **128**
operator	String	**Parameter description**: data comparison operator. **Options**: **>**, **<**, **>=**, **<=**, **=**, **in** (match in specified values), and **between** (value range).
value	String	**Parameter description**: rvalue of the data comparison expression. When this parameter is used together with the data comparison operator **between**, the rvalue indicates the minimum value and maximum value, which are separated by commas (,). For example, **20,30** indicates that the value is greater than or equal to 20 and less than 30. Maximum: **64**
in_values	Array of strings	**Parameter description**: When **operator** is set to **in**, this parameter is used to transfer the rvalue of the comparison expression. Up to 20 values are allowed. Maximum: **128**

Parameter	Type	Description
strategy	**Strategy** object	**Parameter description**: rule triggering policy, which is used to determine whether the last data meets the conditions. If **rule_type** is set to **DEVICE_LINKAGE**, either this parameter must be specified. This parameter is not supported by device-side rules.

Table 1-418 Strategy

Parameter	Type	Description
trigger	String	**Parameter description**: rule triggering judgment policy. The default value is **pulse**. **Options**: • **pulse**: If the reported data meets conditions, the rule is triggered and the previous data is not judged. • **reverse**: When the data reported this time meets the conditions, the rule is triggered even if the data reported last time does not meet the conditions.
event_valid_time	Integer	**Parameter description**: validity period of the device data, in seconds. The time when the device data is generated is the same as the value of **eventTime** in the reported data. Minimum: **-1**

Table 1-419 SimpleTimerType

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The UTC time is used. The value is in the format of yyyyMMdd'T'HHmmss'Z'. Maximum: **128**
repeat_interval	Integer	**Parameter description**: interval for triggering the rule, in seconds. Minimum: **1** Maximum: **31536000**
repeat_count	Integer	**Parameter description**: number of times that a rule is triggered. Minimum: **1** Maximum: **9999**

Table 1-420 DailyTimerType

Parameter	Type	Description
time	String	**Parameter description**: time when a rule is triggered. The value is in the format of HH:MM. Maximum: **128**
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. Maximum: **128**

Table 1-421 DeviceLinkageStatusCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device status triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device status triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
status_list	Array of strings	**Parameter description**: status list. This parameter must be carried for a device status condition. **Options**: • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. Maximum: **128**

Parameter	Type	Description
duration	Integer	**Duration**: device status duration. Value range: 0–60 (minutes). Minimum: **0** Maximum: **60** Default: **0**

Table 1-422 TimeRange

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The value is in the format of HH:mm.
end_time	String	**Parameter description**: end time for rule triggering. The value is in the format of HH:mm. If the end time is the same as the start time, the rule is triggered at any time of a day.
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. The date in the week list indicates the start date.

Table 1-423 RuleAction

Parameter	Type	Description
type	String	**Parameter description**: rule action type. Device-side rules support only **DEVICE_CMD**. **Options**: • **DEVICE_CMD**: delivering a device command message. • **SMN_FORWARDING**: sending an SMN message. • **DEVICE_ALARM**: reporting a device alarm. This type can only be selected when **condition** contains **DEVICE_DATA**. The action of this type must be unique.
device_command	**ActionDeviceCommand** object	Content of the device command message to deliver. This parameter is mandatory when **type** is set to **DEVICE_CMD**.

Parameter	Type	Description
smn_forwarding	**ActionSmnForwarding** object	Structure of the SMN message to send. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **SMN_FORWARDING**.
device_alarm	**ActionDeviceAlarm** object	Content of the device alarm message to report. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_ALARM**.

Table 1-424 ActionDeviceCommand

Parameter	Type	Description
device_id	String	**Parameter description**: ID of the device that the command is delivered to. • When a device data rule is created, if **device_id** is empty, the command is delivered to the device for which the rule is triggered. • This parameter cannot be left empty when a scheduled rule is created. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
cmd	**CMD** object	**Parameter description**: command to deliver.

Table 1-425 CMD

Parameter	Type	Description
command_name	String	**Parameter description**: command name, which is defined in the product model associated with the device. Maximum: **128**

Parameter	Type	Description
command_body	Object	**Parameter description**: command parameter in JSON format. An example command delivered to an LwM2M device is **{"value":"1"}**, which is a key-value pair. Each key is a **paraName** parameter in commands in the product model. An example command delivered to an MQTT device is **{"header": {"mode": "ACK","from": "/users/testUser","method": "SET_TEMPERATURE_READ_PERIOD","to":"/devices/{device_id}/services/{service_id}"},"body": {"value" : "1"}}**. • **mode**: indicates whether the device needs to reply an acknowledgment message after receiving a command. The default value is **ACK**. This parameter is mandatory. **ACK** indicates that an acknowledgment message must be replied. **NOACK** indicates that no acknowledgment message is required. Other values are invalid. • **from**: indicates the address of the command sender. This parameter is optional. The formats of requests initiated by the application, application server, and IoT platform are /users/{*userId*}, /{*serviceName*}, and /cloud/{*serviceName*}, respectively. • **to**: indicates the address of the command receiver. The value is in the format of /devices/{*device_id*}/services/{*service_id*}. This parameter is optional. • **method**: indicates the command name defined in the product model. This parameter is optional. • **body**: indicates the message body of the command. The value consists of key-value pairs. Each key is a **paraName** parameter in commands in the product model. This parameter optional. The specific format depends on the application and device.
service_id	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. Maximum: **64**

Parameter	Type	Description
buffer_timeout	Integer	**Parameter description**: duration for the platform to cache a device command before delivering it to the device, in seconds. After the duration elapses, the platform does not deliver this command. The default value is **172800** (48 hours). If **buffer_timeout** is set to **0**, the command is delivered to the device immediately regardless of the command delivery mode set on the platform. Minimum: **0** Maximum: **31536000** Default: **172800**
response_timeout	Integer	**Parameter description**: validity period of a command response, in seconds. If the platform does not receive a response to a command within the time specified by **response_timeout**, the command times out. The default value is **1800**. Minimum: **1** Maximum: **31536000** Default: **1800**
mode	String	**Parameter description**: mode in which device commands are delivered. This parameter is valid only when the value of **buffer_timeout** is greater than **0**. • **ACTIVE**: The platform directly delivers commands to devices. • **PASSIVE**: After creating a device command, the platform caches the command. When the device goes online or the execution result of the previous command is reported, the platform delivers this command.

Table 1-426 ActionSmnForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the SMN service is deployed. Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the SMN service belongs.

Parameter	Type	Description
theme_name	String	**Parameter description**: SMN topic name. Maximum: **256**
topic_urn	String	**Parameter description**: SMN topic URN. Maximum: **256**
message_content	String	**Parameter description**: SMS or email content. Maximum: **1024**
message_template_name	String	**Parameter description**: SMN template name.
message_title	String	**Parameter description**: SMS or email topic. The value is UTF-8 encoded and cannot exceed 521 bytes.

Table 1-427 ActionDeviceAlarm

Parameter	Type	Description
name	String	**Parameter description**: alarm name. Maximum: **256**
alarm_status	String	**Parameter description**: alarm status. **Options**: • **fault**: The alarm is reported. • **recovery**: The alarm is cleared.
severity	String	**Parameter description**: alarm severity. **Options**: **warning**, **minor**, **major**, and **critical**.
dimension	String	**Parameter description**: Alarm dimension, which is used together with the alarm name and alarm severity to identify an alarm. Alarms of the user dimension are used by default. Alarms of the device and resource space dimensions are supported. **Options**: • device: device dimension. • app: resource space dimension.
description	String	**Parameter description**: alarm description. Maximum: **256**

Table 1-428 DeviceSide

Parameter	Type	Description
device_ids	Array of strings	** Parameter description**: IDs of target devices to which the device-side rule is delivered. A unique device ID is allocated by the platform during device registration. Maximum: **128**

Example Requests

Modifies the linkage rule. The rule is executed from 18:00 to 24:00 on Tuesday. When the value of **visibility** is less than 30, 19:00, or the specified time arrives, an alarm is generated and a command is delivered to the device.

```
PUT https://{endpoint}/v5/iot/{project_id}/rules/{rule_id}

{
  "name" : "openLight",
  "description" : "string",
  "condition_group" : {
    "time_range" : {
      "days_of_week" : 2,
      "start_time" : "18:00",
      "end_time" : "23:00"
    },
    "logic" : "or",
    "conditions" : [ {
      "device_property_condition" : {
        "device_id" : "07b69d78-c716-4be6-9545-869920738397",
        "product_id" : "074abacf-cdb1-4f52-8c88-5864e50d332c",
        "filters" : [ {
          "path" : "StreetLight/visibility",
          "strategy" : {
            "event_valid_time" : 300,
            "trigger" : "reverse"
          },
          "value" : "30",
          "operator" : "<"
        } ]
      },
      "daily_timer_condition" : {
        "days_of_week" : 2,
        "time" : "19:00"
      },
      "type" : "DEVICE_DATA",
      "simple_timer_condition" : {
        "start_time" : "20190122T141500Z",
        "repeat_interval" : 1,
        "repeat_count" : 1
      }
    } ]
  },
  "actions" : [ {
    "device_alarm" : {
      "severity" : "warning",
      "alarm_status" : "fault",
      "name" : "The device property is abnormal.",
      "description" : "****The device property is abnormal."
    },
    "type" : "DEVICE_CMD",
    "smn_forwarding" : {
```



```
    "message_content" : "message_content",
    "project_id" : "project_id",
    "message_title" : "message_title",
    "theme_name" : "theme_name",
    "region_name" : "region_name",
    "topic_urn" : "topic_urn"
  },
  "device_command" : {
    "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
    "cmd" : {
      "buffer_timeout" : 0,
      "mode" : "string",
      "response_timeout" : 1800,
      "command_body" : {
        "value" : 0
      },
      "service_id" : "Switch",
      "command_name" : "SET_LIGHT_SWITCH"
    }
  }
 } ],
 "rule_type" : "DEVICE_LINKAGE",
 "status" : "string",
 "app_id" : "string"
}
```

Example Responses

Status code: 200

OK

```
{
 "rule_id" : "5eb3628d017d9105d0cf9aec",
 "name" : "openLight",
 "condition_group" : {
  "conditions" : [ {
   "type" : "DEVICE_DATA",
   "device_property_condition" : {
    "device_id" : "07b69d78-c716-4be6-9545-869920738397",
    "filters" : [ {
     "path" : "StreetLight/visibility",
     "operator" : "<",
     "value" : "30",
     "strategy" : {
      "trigger" : "reverse",
      "event_valid_time" : 300
     }
    } ]
   }
  } ],
  "logic" : "and"
 },
 "actions" : [ {
  "type" : "DEVICE_CMD",
  "device_command" : {
   "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
   "cmd" : {
    "command_name" : "SET_LIGHT_SWITCH",
    "command_body" : {
     "value" : 0
    },
    "service_id" : "Switch",
    "buffer_timeout" : 0,
    "response_timeout" : 1800
   }
  }
 } ],
 "rule_type" : "DEVICE_LINKAGE",
```



```
  "status" : "active",
  "app_id" : "9562bf8541e44361b6ae3a7e9fbe1144",
  "last_update_time" : "20221017T025425Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.10.4 Query a Rule

Function

This API is used by an application to query the configuration of a specific rule on the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/rules/{rule_id}

Table 1-429 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
rule_id	Yes	String	**Parameter description**: unique rule ID, which is allocated by the platform during rule creation. **Value**: The value can contain a maximum of 32 characters. Only letters and digits are allowed.

Request Parameters

Table 1-430 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-431 Response body parameters

Parameter	Type	Description
rule_id	String	Rule ID. Maximum: **128**
name	String	Rule name. Minimum: **1** Maximum: **128**
description	String	Rule description. Maximum: **256**
condition_group	**ConditionGroup** object	Condition group of a rule, including simple and complex rules.
actions	Array of **RuleAction** objects	Action list of the rule. A maximum of 10 actions can be configured for a rule.
rule_type	String	Rule type. • **DEVICE_LINKAGE**: cloud linkage rule. • **DEVICE_SIDE**: Device-side rule.
status	String	Rule status. The default value is **active**. • **active**: activated. • **inactive**: not activated.
app_id	String	Resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the rule to create will belong. If this parameter is not specified, the rule to create will belong to the **default resource space**.
edge_node_ids	Array of strings	List of edge node IDs.
last_update_time	String	UTC time when the rule was last updated. The value is in the format of yyyyMMdd'T'HHmmss'Z'.
device_side	**DeviceSide** object	**Parameter description**: information of devices to which the device-side rule is delivered. This parameter is mandatory when **rule_type** is set to **DEVICE_SIDE**.

Table 1-432 ConditionGroup

Parameter	Type	Description
conditions	Array of **RuleCondition** objects	**Parameter description**: condition list of the rule. A maximum of 10 conditions can be configured for a rule.
logic	String	**Parameter description**: logical relationship between multiple conditions of the rule. The default value is **and**. **Options**: • **and**: All of the conditions are judged. • **or**: One of the conditions is judged.
time_range	**TimeRange** object	**Parameter description**: validity period of a condition.

Table 1-433 RuleCondition

Parameter	Type	Description
type	String	**Parameter description**: type of the rule condition. **Options**: • **DEVICE_DATA**: device property data condition. • **SIMPLE_TIMER**: simple timing condition. • **DAILY_TIMER**: daily scheduled condition. • **DEVICE_LINKAGE_STATUS**: device status condition.
device_property_condition	**DeviceDataCondition** object	**Parameter description**: details of the device data condition. This parameter is mandatory when **type** is set to **DEVICE_DATA**.
simple_timer_condition	**SimpleTimerType** object	**Parameter description**: details of the simple timing condition. This parameter is mandatory when **type** is set to **SIMPLE_TIMER**.
daily_timer_condition	**DailyTimerType** object	**Parameter description**: details of the daily scheduled condition. This parameter is mandatory when **type** is set to **DAILY_TIMER**.
device_linkage_status_condition	**DeviceLinkageStatusCondition** object	**Parameter description**: details of the device status condition. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_LINKAGE_STATUS**.

Table 1-434 DeviceDataCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device property triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device property triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
filters	Array of **PropertyFilter** objects	Data filter criteria.

Table 1-435 PropertyFilter

Parameter	Type	Description
path	String	**Parameter description**: path of the device properties. The value is in the format of *[service_id]/DataProperty*. For example, the path of the door sensor status is **DoorWindow/status**. Maximum: **128**
operator	String	**Parameter description**: data comparison operator. **Options**: >, <, >=, <=, =, **in** (match in specified values), and **between** (value range).

Parameter	Type	Description
value	String	**Parameter description**: rvalue of the data comparison expression. When this parameter is used together with the data comparison operator **between**, the rvalue indicates the minimum value and maximum value, which are separated by commas (,). For example, **20,30** indicates that the value is greater than or equal to 20 and less than 30. Maximum: **64**
in_values	Array of strings	**Parameter description**: When **operator** is set to **in**, this parameter is used to transfer the rvalue of the comparison expression. Up to 20 values are allowed. Maximum: **128**
strategy	**Strategy** object	**Parameter description**: rule triggering policy, which is used to determine whether the last data meets the conditions. If **rule_type** is set to **DEVICE_LINKAGE**, either this parameter must be specified. This parameter is not supported by device-side rules.

Table 1-436 Strategy

Parameter	Type	Description
trigger	String	**Parameter description**: rule triggering judgment policy. The default value is **pulse**. **Options**: • **pulse**: If the reported data meets conditions, the rule is triggered and the previous data is not judged. • **reverse**: When the data reported this time meets the conditions, the rule is triggered even if the data reported last time does not meet the conditions.
event_valid_time	Integer	**Parameter description**: validity period of the device data, in seconds. The time when the device data is generated is the same as the value of **eventTime** in the reported data. Minimum: **-1**

Table 1-437 SimpleTimerType

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The UTC time is used. The value is in the format of yyyyMMdd'T'HHmmss'Z'. Maximum: **128**
repeat_interval	Integer	**Parameter description**: interval for triggering the rule, in seconds. Minimum: **1** Maximum: **31536000**
repeat_count	Integer	**Parameter description**: number of times that a rule is triggered. Minimum: **1** Maximum: **9999**

Table 1-438 DailyTimerType

Parameter	Type	Description
time	String	**Parameter description**: time when a rule is triggered. The value is in the format of HH:MM. Maximum: **128**
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. Maximum: **128**

Table 1-439 DeviceLinkageStatusCondition

Parameter	Type	Description
device_id	String	**Parameter description**: device ID. The ID is unique and is allocated by the platform during device registration. If this parameter is specified, device status triggering is based on the specified device. This parameter and **product_id** cannot be both empty. If this parameter and **product_id** are both specified, the value of this parameter is used for filtering. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. If this parameter is specified and **device_id** is empty, all devices of the product are matched for device status triggering. This parameter and **device_id** cannot be both empty. Maximum: **128**
status_list	Array of strings	**Parameter description**: status list. This parameter must be carried for a device status condition. **Options**: • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. Maximum: **128**
duration	Integer	**Duration**: device status duration. Value range: 0–60 (minutes). Minimum: **0** Maximum: **60** Default: **0**

Table 1-440 TimeRange

Parameter	Type	Description
start_time	String	**Parameter description**: start time for rule triggering. The value is in the format of HH:mm.

Parameter	Type	Description
end_time	String	**Parameter description**: end time for rule triggering. The value is in the format of HH:mm. If the end time is the same as the start time, the rule is triggered at any time of a day.
days_of_week	String	**Parameter description**: list of days in a week. Days are separated by commas (,). The value **1** indicates Sunday, the value **2** indicates Monday, and the rest can be deduced by analogy. By default, every day is included. The date in the week list indicates the start date.

Table 1-441 RuleAction

Parameter	Type	Description
type	String	**Parameter description**: rule action type. Device-side rules support only **DEVICE_CMD**. **Options**: • **DEVICE_CMD**: delivering a device command message. • **SMN_FORWARDING**: sending an SMN message. • **DEVICE_ALARM**: reporting a device alarm. This type can only be selected when **condition** contains **DEVICE_DATA**. The action of this type must be unique.
device_command	**ActionDeviceCommand** object	Content of the device command message to deliver. This parameter is mandatory when **type** is set to **DEVICE_CMD**.
smn_forwarding	**ActionSmnForwarding** object	Structure of the SMN message to send. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **SMN_FORWARDING**.
device_alarm	**ActionDeviceAlarm** object	Content of the device alarm message to report. This parameter is mandatory when **rule_type** is set to **DEVICE_LINKAGE** and **type** is set to **DEVICE_ALARM**.

Table 1-442 ActionDeviceCommand

Parameter	Type	Description
device_id	String	**Parameter description**: ID of the device that the command is delivered to. • When a device data rule is created, if **device_id** is empty, the command is delivered to the device for which the rule is triggered. • This parameter cannot be left empty when a scheduled rule is created. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
cmd	**CMD** object	**Parameter description**: command to deliver.

Table 1-443 CMD

Parameter	Type	Description
command_name	String	**Parameter description**: command name, which is defined in the product model associated with the device. Maximum: **128**

Parameter	Type	Description
command_body	Object	**Parameter description**: command parameter in JSON format. An example command delivered to an LwM2M device is **{"value":"1"}**, which is a key-value pair. Each key is a **paraName** parameter in commands in the product model. An example command delivered to an MQTT device is **{"header": {"mode": "ACK","from": "/users/testUser","method": "SET_TEMPERATURE_READ_PERIOD","to":"/devices/{device_id}/services/{service_id}"},"body": {"value" : "1"}}**. • **mode**: indicates whether the device needs to reply an acknowledgment message after receiving a command. The default value is **ACK**. This parameter is mandatory. **ACK** indicates that an acknowledgment message must be replied. **NOACK** indicates that no acknowledgment message is required. Other values are invalid. • **from**: indicates the address of the command sender. This parameter is optional. The formats of requests initiated by the application, application server, and IoT platform are /users/{*userId*}, /{*serviceName*}, and /cloud/{*serviceName*}, respectively. • **to**: indicates the address of the command receiver. The value is in the format of /devices/{*device_id*}/services/{*service_id*}. This parameter is optional. • **method**: indicates the command name defined in the product model. This parameter is optional. • **body**: indicates the message body of the command. The value consists of key-value pairs. Each key is a **paraName** parameter in commands in the product model. This parameter optional. The specific format depends on the application and device.
service_id	String	**Parameter description**: ID of the device service that the device command belongs to, which is defined in the product model associated with the device. Maximum: **64**

Parameter	Type	Description
buffer_timeout	Integer	**Parameter description**: duration for the platform to cache a device command before delivering it to the device, in seconds. After the duration elapses, the platform does not deliver this command. The default value is **172800** (48 hours). If **buffer_timeout** is set to **0**, the command is delivered to the device immediately regardless of the command delivery mode set on the platform. Minimum: **0** Maximum: **31536000** Default: **172800**
response_timeout	Integer	**Parameter description**: validity period of a command response, in seconds. If the platform does not receive a response to a command within the time specified by **response_timeout**, the command times out. The default value is **1800**. Minimum: **1** Maximum: **31536000** Default: **1800**
mode	String	**Parameter description**: mode in which device commands are delivered. This parameter is valid only when the value of **buffer_timeout** is greater than **0**. • **ACTIVE**: The platform directly delivers commands to devices. • **PASSIVE**: After creating a device command, the platform caches the command. When the device goes online or the execution result of the previous command is reported, the platform delivers this command.

Table 1-444 ActionSmnForwarding

Parameter	Type	Description
region_name	String	**Parameter description**: region where the SMN service is deployed. Maximum: **256**
project_id	String	**Parameter description**: ID of the project to which the SMN service belongs.

Parameter	Type	Description
theme_name	String	**Parameter description**: SMN topic name. Maximum: **256**
topic_urn	String	**Parameter description**: SMN topic URN. Maximum: **256**
message_content	String	**Parameter description**: SMS or email content. Maximum: **1024**
message_template_name	String	**Parameter description**: SMN template name.
message_title	String	**Parameter description**: SMS or email topic. The value is UTF-8 encoded and cannot exceed 521 bytes.

Table 1-445 ActionDeviceAlarm

Parameter	Type	Description
name	String	**Parameter description**: alarm name. Maximum: **256**
alarm_status	String	**Parameter description**: alarm status. **Options**: • **fault**: The alarm is reported. • **recovery**: The alarm is cleared.
severity	String	**Parameter description**: alarm severity. **Options**: **warning**, **minor**, **major**, and **critical**.
dimension	String	**Parameter description**: Alarm dimension, which is used together with the alarm name and alarm severity to identify an alarm. Alarms of the user dimension are used by default. Alarms of the device and resource space dimensions are supported. **Options**: • device: device dimension. • app: resource space dimension.
description	String	**Parameter description**: alarm description. Maximum: **256**

Table 1-446 DeviceSide

Parameter	Type	Description
device_ids	Array of strings	** Parameter description**: IDs of target devices to which the device-side rule is delivered. A unique device ID is allocated by the platform during device registration. Maximum: **128**

Example Requests

Queries a device rule.

```
GET https://{endpoint}/v5/iot/{project_id}/rules/{rule_id}
```

Example Responses

Status code: 200

OK

```
{
  "rule_id" : "5eb3628d017d9105d0cf9aec",
  "name" : "openLight",
  "condition_group" : {
    "conditions" : [ {
      "type" : "DEVICE_DATA",
      "device_property_condition" : {
        "device_id" : "07b69d78-c716-4be6-9545-869920738397",
        "filters" : [ {
          "path" : "StreetLight/visibility",
          "operator" : "<",
          "value" : "30",
          "strategy" : {
            "trigger" : "reverse",
            "event_valid_time" : 300
          }
        } ]
      }
    } ],
    "logic" : "and"
  },
  "actions" : [ {
    "type" : "DEVICE_CMD",
    "device_command" : {
      "device_id" : "3a9e52d9-3ebf-4985-89e9-6d2396748a2f",
      "cmd" : {
        "command_name" : "SET_LIGHT_SWITCH",
        "command_body" : {
          "value" : 0
        },
        "service_id" : "Switch",
        "buffer_timeout" : 0,
        "response_timeout" : 1800
      }
    }
  } ],
  "rule_type" : "DEVICE_LINKAGE",
  "status" : "active",
  "app_id" : "9562bf8541e44361b6ae3a7e9fbe1144",
  "last_update_time" : "20221017T025425Z"
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.10.5 Delete a Rule

Function

This API is used by an application to delete a specific rule from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/rules/{rule_id}

Table 1-447 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
rule_id	Yes	String	**Parameter description**: unique rule ID, which is allocated by the platform during rule creation. **Value**: The value can contain a maximum of 32 characters. Only letters and digits are allowed.

Request Parameters

Table 1-448 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a device rule.


```
DELETE https://{endpoint}/v5/iot/{project_id}/rules/{rule_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.10.6 Modify the Rule Status

Function

This API is used by an application to modify the status of a specific rule on the platform to activate or deactivate the rule.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/rules/{rule_id}/status

Table 1-449 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
rule_id	Yes	String	**Parameter description**: rule ID. **Value**: The value can contain a maximum of 32 characters. Only letters and digits are allowed.

Request Parameters

Table 1-450 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-451 Request body parameters

Parameter	Mandatory	Type	Description
status	Yes	String	**Parameter description**: rule status. **Options**: • **active**: activated. • **inactive**: not activated.

Response Parameters

Status code: 200

Table 1-452 Response body parameters

Parameter	Type	Description
status	String	**Parameter description**: rule status. **Options**: • **active**: activated. • **inactive**: not activated.

Example Requests

Changes the status of the rule to **Active**.

```
PUT https://{endpoint}/v5/iot/{project_id}/rules/{rule_id}/status

{
  "status" : "active"
}
```

Example Responses

Status code: 200

OK

```
{
  "status" : "active"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.11 Device Shadow

Introduction:

A device shadow is a JSON file used to store and obtain device status.

- Each device has only one device shadow, which is uniquely identified by the device ID.

- A device shadow stores the properties reported by a device and the properties expected by an application.

- You can obtain and set the device statuses through the device shadow, regardless of whether the device is online.

1.4.11.1 Query a Device Shadow

Function

This API is used by an application to query the device shadow of a specific device, including the desired device properties (in the desired section) and the latest properties reported by the device (in the reported section).

Introduction: A device shadow is a JSON file used to store and obtain device status.

- Each device has only one device shadow, which is uniquely identified by the device ID.
- A device shadow stores the properties (status) reported by a device and the properties (status) expected by an application.
- You can obtain and set the device properties through the device shadow, regardless of whether the device is online.
- After a device goes online or reports data, if the reported properties are different from the desired ones, the desired properties are delivered to the device. The desired properties can be delivered only when they have been defined in the product model and **method** is specified as **W**.

Restrictions: Keys in the device shadow JSON file cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). Otherwise, the device shadow file cannot be updated.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/devices/{device_id}/shadow

Table 1-453 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-454 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-455 Response body parameters

Parameter	Type	Description
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.
shadow	Array of **DeviceShadowData** objects	Device shadow data structure.

Table 1-456 DeviceShadowData

Parameter	Type	Description
service_id	String	Device service ID, which is defined in the product model associated with the device. Minimum: **1** Maximum: **256**
desired	**DeviceShadowProperties** object	Latest desired properties delivered to the device. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model.
reported	**DeviceShadowProperties** object	Latest properties reported by the device. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model.
version	Long	Version of the device shadow. When this parameter is carried, the platform will check whether the value is the same as the current shadow version. The initial value is **0**.

Table 1-457 DeviceShadowProperties

Parameter	Type	Description
properties	Object	Properties stored in the device shadow. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model. Only one-layer structure is supported, as shown in the example. **Note**: Keys in the JSON file cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). Keys that contain these special characters will cause device shadow update failures.
event_time	String	Time when the event occurred. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. Maximum: **256**

Example Requests

Query the device shadow of a specified device.

```
GET https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/shadow
```

Example Responses

Status code: 200

OK

```
{
  "device_id" : "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "properties" : {
        "temperature" : "60"
      },
      "event_time" : "20151212T121212Z"
    },
    "reported" : {
      "properties" : {
        "temperature" : "60"
      },
      "event_time" : "20151212T121212Z"
    },
    "version" : 1
  } ]
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	INTERNAL SERVER ERROR

Error Codes

See **Error Codes**.

1.4.11.2 Configure Desired Properties in a Device Shadow

Function

This API is used by an application to configure properties in the desired section of a device shadow. When the device goes online or reports properties, the desired properties are delivered to the device.

Introduction: A device shadow is a JSON file used to store and obtain device status.

- Each device has only one device shadow, which is uniquely identified by the device ID.
- A device shadow stores the properties (status) reported by a device and the properties (status) expected by an application.
- You can obtain and set the device properties through the device shadow, regardless of whether the device is online.
- After a device goes online or reports data, if the reported properties are different from the desired ones, the desired properties are delivered to the device. The desired properties can be delivered only when they have been defined in the product model and **method** is specified as **W**.
- This API can be used to configure desired shadow data of a single device. For details about how to configure desired shadow data of multiple devices, see **Creating a Batch Task**.

Restrictions: Keys in the device shadow JSON file cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). Otherwise, the device shadow file cannot be updated.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/devices/{device_id}/shadow

Table 1-458 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id][node_id]*. ***Value****: The value can contain a maximum of 128 characters. Only letters, digits, underscores (*_*), and hyphens (-) are allowed.*

Request Parameters

Table 1-459 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-460 Request body parameters

Parameter	Mandatory	Type	Description
shadow	No	Array of **UpdateDesired** objects	**Parameter description**: data structure of the desired properties in the device shadow.

Table 1-461 UpdateDesired

Parameter	Mandatory	Type	Description
service_id	Yes	String	**Parameter description**: device service ID, which is defined in the product model associated with the device. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
desired	Yes	Object	**Parameter description**: desired properties in the device shadow. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model. Only one-layer structure is supported, as shown in the example request. To delete all desired properties, leave the value empty (for example, **"desired":{}**). To delete a desired property, set the value of the property to **null** (for example, **{"temperature":null}**).
version	No	Long	**Parameter description**: version of the device shadow. When this parameter is carried, the platform will check whether the value is the same as the current shadow version. The initial value is **0**.

Response Parameters

Status code: 200

Table 1-462 Response body parameters

Parameter	Type	Description
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.
shadow	Array of **DeviceShadowData** objects	Device shadow data structure.

Table 1-463 DeviceShadowData

Parameter	Type	Description
service_id	String	Device service ID, which is defined in the product model associated with the device. Minimum: **1** Maximum: **256**
desired	**DeviceShadowProperties** object	Latest desired properties delivered to the device. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model.
reported	**DeviceShadowProperties** object	Latest properties reported by the device. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model.
version	Long	Version of the device shadow. When this parameter is carried, the platform will check whether the value is the same as the current shadow version. The initial value is **0**.

Table 1-464 DeviceShadowProperties

Parameter	Type	Description
properties	Object	Properties stored in the device shadow. The data is JSON data and is in the format of key-value pairs. Each key is a parameter name of a property (**property_name**) in the product model. Only one-layer structure is supported, as shown in the example. **Note**: Keys in the JSON file cannot contain periods (.), dollar signs ($), and the null character (hexadecimal ASCII code 00). Keys that contain these special characters will cause device shadow update failures.
event_time	String	Time when the event occurred. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. Maximum: **256**

Example Requests

- Configures the expected data of the device shadow. The expected value of **temperature** is **60**.


```
PUT https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/shadow

{
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "temperature" : "60"
    },
    "version" : 1
  } ]
}
```

- Delete the expected value of the **temperature** property from the device shadow.

```
PUT https://{endpoint}/v5/iot/{project_id}/devices/{device_id}/shadow

{
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "temperature" : null
    },
    "version" : 2
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "device_id" : "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",
  "shadow" : [ {
    "service_id" : "WaterMeter",
    "desired" : {
      "properties" : {
        "temperature" : "60"
      },
      "event_time" : "20151212T121212Z"
    },
    "reported" : {
      "properties" : {
        "temperature" : "60"
      },
      "event_time" : "20151212T121212Z"
    },
    "version" : 2
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	FORBIDDEN

Status Code	Description
404	Not Found
409	CONFLICT
500	INTERNAL SERVER ERROR

Error Codes

See **Error Codes**.

1.4.12 Group Management

1.4.12.1 Create a Device Group

Function

This API is used by an application to create a device group. Up to 1000 device groups, including parent device groups and child device groups, can be created in a Huawei Cloud account. The maximum number of group levels is five. It means that the relationship tree of groups can have up to five levels.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-group

Table 1-465 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-466 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-467 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	**Parameter description**: device group name, which must be unique in a single resource space. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: device group description. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-. Minimum: **1** Maximum: **64**
super_group_id	No	String	** Parameter description**: parent device group ID. If this parameter is carried, a child device group is created in the device group. Dynamic groups do not support this parameter. **Value**: The value can contain a maximum of 36 characters, including hexadecimal strings and hyphens (-).
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the device group to create will belong. If this parameter is not specified, the device group to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
group_type	No	String	**Parameter description**: device group type. A static device group is used by default. When a dynamic device group is used, enter the dynamic device group rule.
dynamic_group_rule	No	String	**Parameter description**: The syntax of dynamic device group rules is the same as that of advanced search. You only need to fill in the **where** clause.

Response Parameters

Status code: 201

Table 1-468 Response body parameters

Parameter	Type	Description
group_id	String	Device group ID. The ID is unique and is allocated by the platform during device group creation.
name	String	Device group name, which must be unique in a single resource space.
description	String	Device group description.
super_group_id	String	ID of the parent device group.
group_type	String	Device group type, which can be dynamic or static.
dynamic_group_rule	String	Dynamic device group rule.

Example Requests

- Creates a static device group named **GroupA**.

```
POST https://{endpoint}/v5/iot/{project_id}/device-group

{
  "name" : "GroupA",
  "description" : "GroupA",
  "super_group_id" : "04ed32dc1b0025b52fe3c01a27c2b0a8",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "group_type" : "STATIC"
}
```

- Creates a dynamic device group named **GroupA**.

```
POST https://{endpoint}/v5/iot/{project_id}/device-group

{
  "name" : "GroupA",
  "description" : "GroupA",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "dynamic_group_rule" : "product_id = '63fef97897bacf7a56438cba'",
  "group_type" : "DYNAMIC"
}
```

Example Responses

Status code: 201

Created


```
{
  "group_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
  "name" : "GroupA",
  "description" : "GroupA",
  "super_group_id" : "04ed32dc1b0025b52fe3c01a27c2b0a8",
  "group_type" : "STATIC",
  "dynamic_group_rule" : null
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.12.2 Query the Device Group List

Function

This API is used by an application to query the device group list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-group

Table 1-469 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-470 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**
last_modified_time	No	String	**Parameter description**: Modification records of a device group after the time specified by **last_modified_time** are queried. **Value**: The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. It is used by users who have multiple resource spaces to query the product list of a specified resource space. If this parameter is not carried, the list of all products of the user will be returned. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
group_type	No	String	**Parameter description**: device group type. A static device group is used by default. When a dynamic device group is used, enter the dynamic device group rule.
name	No	String	**Parameter description**: device group name, which must be unique in a single resource space. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-. Minimum: **1** Maximum: **64**

Request Parameters

Table 1-471 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-472 Response body parameters

Parameter	Type	Description
device_groups	Array of **DeviceGroupResponseSummary** objects	Device group information list.
page	**Page** object	Pagination information of the query results.

Table 1-473 DeviceGroupResponseSummary

Parameter	Type	Description
group_id	String	Device group ID. The ID is unique and is allocated by the platform during device group creation.
name	String	Device group name, which must be unique in a single resource space.
description	String	Device group description.
super_group_id	String	ID of the parent device group.
group_type	String	**Parameter description**: device group type. A static device group is used by default. When a dynamic device group is used, enter the dynamic device group rule.

Table 1-474 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries a device group list.

```
GET https://{endpoint}/v5/iot/{project_id}/device-group
```

Example Responses

Status code: 200

OK

```
{
  "device_groups" : [ {
    "group_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
    "name" : "GroupA",
    "description" : "GroupA",
    "super_group_id" : "04ed32dc1b0025b52fe3c01a27c2b0a8",
    "group_type" : "STATIC"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.12.3 Query a Device Group

Function

This API is used by an application to query details about a specific device group.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-group/{group_id}

Table 1-475 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
group_id	Yes	String	**Parameter description**: device group ID. The ID is unique and is allocated by the platform during device group creation. **Value**: The value can contain a maximum of 36 characters, including hexadecimal strings and hyphens (-).

Request Parameters

Table 1-476 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-477 Response body parameters

Parameter	Type	Description
group_id	String	Device group ID. The ID is unique and is allocated by the platform during device group creation.
name	String	Device group name, which must be unique in a single resource space.
description	String	Device group description.
super_group_id	String	ID of the parent device group.

Parameter	Type	Description
group_type	String	Device group type, which can be dynamic or static.
dynamic_group_rule	String	Dynamic device group rule.

Example Requests

Queries details about a specified device group.

```
GET https://{endpoint}/v5/iot/{project_id}/device-group/{group_id}
```

Example Responses

Status code: 200

OK

```
{
  "group_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
  "name" : "GroupA",
  "description" : "GroupA",
  "super_group_id" : "04ed32dc1b0025b52fe3c01a27c2b0a8",
  "group_type" : "STATIC"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
404	Not Found
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.12.4 Modify a Device Group

Function

This API is used by an application to modify a specific device group on the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/device-group/{group_id}

Table 1-478 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
group_id	Yes	String	**Parameter description**: device group ID. The ID is unique and is allocated by the platform during device group creation. **Value**: The value can contain a maximum of 36 characters, including hexadecimal strings and hyphens (-).

Request Parameters

Table 1-479 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-480 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	**Parameter description**: device group name, which must be unique in a single resource space. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,&%@!-. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: device group description. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and question marks (?) are allowed. '#().,& %@!-. Minimum: **1** Maximum: **64**

Response Parameters

Status code: 200

Table 1-481 Response body parameters

Parameter	Type	Description
group_id	String	Device group ID. The ID is unique and is allocated by the platform during device group creation.
name	String	Device group name, which must be unique in a single resource space.
description	String	Device group description.
super_group_id	String	ID of the parent device group.
group_type	String	Device group type, which can be dynamic or static.
dynamic_group_rule	String	Dynamic device group rule.

Example Requests

Changes the name of the device group to **GroupA**.

```
PUT https://{endpoint}/v5/iot/{project_id}/device-group/{group_id}

{
  "name" : "GroupA",
  "description" : "GroupA"
}
```

Example Responses

Status code: 200

OK

```
{
  "group_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
  "name" : "GroupA",
  "description" : "GroupA",
  "super_group_id" : "04ed32dc1b0025b52fe3c01a27c2b0a8",
  "group_type" : "STATIC",
  "dynamic_group_rule" : null
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.12.5 Delete a Device Group

Function

This API is used by an application to delete a device group.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/device-group/{group_id}

Table 1-482 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
group_id	Yes	String	**Parameter description**: device group ID. The ID is unique and is allocated by the platform during device group creation. **Value**: The value can contain a maximum of 36 characters, including hexadecimal strings and hyphens (-).

Request Parameters

Table 1-483 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a device group.

```
DELETE https://{endpoint}/v5/iot/{project_id}/device-group/{group_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.12.6 Manage Devices in a Device Group

Function

This API is used by an application to manage devices in a device group. Up to 20,000 devices can be added to a device group. A device can be added to up to 10 device groups.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-group/{group_id}/action

Table 1-484 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
group_id	Yes	String	**Parameter description**: device group ID. The ID is unique and is allocated by the platform during device group creation. **Value**: The value can contain a maximum of 36 characters, including hexadecimal strings and hyphens (-).

Table 1-485 Query Parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	**Parameter description**: operation type. You can add or delete a device. **Options**: • **addDevice**: Add a registered device to a specific device group. • **removeDevice**: Remove a device from a specific device group. This operation only removes the relationship between the device and the device group. The device still exists on the platform.
device_id	Yes	String	**Parameter description**: device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by the platform. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-486 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

- Manages devices in a device group and adds devices to the device group.

  ```
  POST https://{endpoint}/v5/iot/{project_id}/device-group/{group_id}/action?
  action_id=addDevice&device_id=yourDeviceId
  ```

- Manages devices in a device group and removes devices from the device group.

  ```
  POST https://{endpoint}/v5/iot/{project_id}/device-group/{group_id}/action?
  action_id=removeDevice&device_id=yourDeviceId
  ```

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.12.7 Query Devices in a Device Group

Function

This API is used by an application to query the list of devices in a specific group.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-group/{group_id}/devices

Table 1-487 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
group_id	Yes	String	**Parameter description**: device group ID. The ID is unique and is allocated by the platform during device group creation. **Value**: The value can contain a maximum of 36 characters, including hexadecimal strings and hyphens (-).

Table 1-488 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-489 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-490 Response body parameters

Parameter	Type	Description
devices	Array of **SimplifyDevice** objects	Device list.
page	**Page** object	Pagination information of the query results.

Table 1-491 SimplifyDevice

Parameter	Type	Description
device_id	String	Device ID, used to uniquely identify a device. The value of this parameter is specified during device registration or allocated by the platform. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*.
node_id	String	Device identifier. This parameter is set to the IMEI, MAC address, or serial number.
device_name	String	Device name.
product_id	String	Unique ID of the product associated with the device.

Table 1-492 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries devices in a device group.

```
POST https://{endpoint}/v5/iot/{project_id}/device-group/{group_id}/devices
```

Example Responses

Status code: 200

OK

```
{
  "devices" : [ {
    "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "node_id" : "ABC123456789",
    "device_name" : "dianadevice",
    "product_id" : "b640f4c203b7910fc3cbd446ed437cbd"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.13 Tag Management

You can add the same tag to a type of resources with the same properties for easy management. Currently, only devices can be tagged.

Note: Tags prefixed with **iot_** are reserved tags and unavailable to users. Up to 10 tags can be bound to a resource.

1.4.13.1 Bind Tags

Function

This API is used by an application to bind tags to a specific resource. Currently, only devices can be tagged.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/tags/bind-resource

Table 1-493 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-494 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-495 Request body parameters

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Parameter description**: type of the resource that tags are to be bound to. **Options**: • **device**: device.
resource_id	Yes	String	**Parameter description**: ID of the resource that tags are to be bound to. For example, if **resource_type** is set to **device**, set this parameter to the device ID. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
tags	Yes	Array of **TagV5DTO** objects	**Parameter description**: list of tags to be bound to a specific resource. Each tag key must be unique in the tag list. Up to 10 tags can be bound to a resource.

Table 1-496 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.
tag_value	No	String	**Parameter description**: tag value. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Response Parameters

None

Example Requests

Binds the tag to the device whose ID is **d4922d8a**.

```
POST https://{endpoint}/v5/iot/{project_id}/tags/bind-resource

{
  "resource_type" : "device",
  "resource_id" : "d4922d8a",
  "tags" : [ {
    "tag_key" : "testTagName",
    "tag_value" : "testTagValue"
  } ]
}
```


Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.13.2 Unbind Tags

Function

This API is used by an application to unbind tags from a specific resource. Currently, only devices can be tagged.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/tags/unbind-resource

Table 1-497 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-498 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-499 Request body parameters

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Parameter description**: type of the resource that tags are to be bound to. **Options**: • **device**: device.
resource_id	Yes	String	**Parameter description**: ID of the resource that tags are to be bound to. For example, if **resource_type** is set to **device**, set this parameter to the device ID. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
tag_keys	Yes	Array of strings	**Parameter description**: keys of tags to be unbound from the resource. Each tag key must be unique in the tag list. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Response Parameters

None

Example Requests

Unbinds the **testkey** tag from the device whose ID is **d4922d8a**.

```
POST https://{endpoint}/v5/iot/{project_id}/tags/unbind-resource

{
  "resource_type" : "device",
  "resource_id" : "d4922d8a",
  "tag_keys" : [ "testTag" ]
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.13.3 Query Resources by Tag

Function

This API is used by an application to query resources with specific tags. Currently, only devices can be tagged.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/tags/query-resources

Table 1-500 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-501 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-502 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-503 Request body parameters

Parameter	Mandatory	Type	Description
resource_type	Yes	String	**Parameter description**: type of resources to query. Currently, only devices are supported.
tags	Yes	Array of **TagV5DTO** objects	**Parameter description**: tag list. Specify tag key-value pairs. If multiple tags are specified, resources with any of the specified tags are returned.

Table 1-504 TagV5DTO

Parameter	Mandatory	Type	Description
tag_key	Yes	String	**Parameter description**: tag key, which is unique for a resource. If the specified key already exists, the value of the existing tag is overwritten. If the specified key does not exist, a new tag is added. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
tag_value	No	String	**Parameter description**: tag value. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed.

Response Parameters

Status code: 200

Table 1-505 Response body parameters

Parameter	Type	Description
resources	Array of **ResourceDTO** objects	Resource list.
page	**Page** object	Pagination information of the query results.

Table 1-506 ResourceDTO

Parameter	Type	Description
resource_id	String	Resource ID. For example, if **resource_type** is set to **device**, set this parameter to the device ID.

Table 1-507 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries devices bound to a specified tag.

```
POST https://{endpoint}/v5/iot/{project_id}/tags/query-resources

{
```



```
  "resource_type" : "device",
  "tags" : [ {
    "tag_key" : "testTagName",
    "tag_value" : "testTagValue"
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "resources" : [ {
    "resource_id" : "d4922d8a"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.14 Instance Management

IoTDA instance management.

1.4.14.1 Create an IoTDA Instance

Function

This API is used to create an IoTDA instance. For details about the supported instance specifications, see **Specifications**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/iotda-instances

Table 1-508 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.

Request Parameters

Table 1-509 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Table 1-510 Request body parameters

Parameter	Mandatory	Type	Description
instance_type	Yes	String	**Parameter description**: type of the instance to be created. For details about instance types, see **Specifications**. **Options**: • **standard** • **enterprise**

Parameter	Mandatory	Type	Description
flavor	Yes	**Flavor** object	**Parameter description**: specifications of an IoTDA instance.
name	Yes	String	**Parameter description**: instance name. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
charge_info	Yes	**ChargeInfo** object	**Parameter description**: billing information of an IoTDA instance.
description	No	String	**Parameter description**: description of an IoTDA instance. **Value**: The value can contain a maximum of 256 characters. Use only letters, digits, and special characters (_,.&-).
enterprise_project_id	No	String	**Parameter description**: enterprise project ID. If this parameter is set to a specific enterprise project ID or **0** (default enterprise project ID), the enterprise project feature is supported. You can obtain the value from Enterprise Project Management Service (EPS). **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-). Default: **0**
tags	No	Array of **Tag** objects	**Parameter description**: tag information of an IoTDA instance. Array Length: **0 - 20**
additional_params	No	**AdditionalParams** object	Additional parameter information of the enterprise edition.

Table 1-511 Flavor

Parameter	Mandatory	Type	Description
type	Yes	String	**Parameter description**: specification name of the IoTDA instance to create. For details, see **Specifications**.
size	No	Integer	**Parameter description**: unit number of the standard IoTDA instances to create. For details, see **Specifications**. This parameter is mandatory when **instance_type** is set to **standard**. Minimum: **1** Maximum: **200**

Table 1-512 ChargeInfo

Parameter	Mandatory	Type	Description
charge_mode	Yes	String	**Parameter description**: payment mode of the instance. **Options**: • **prePaid**: yearly/monthly • **postPaid**: pay-per-use
period_type	No	String	**Parameter description**: period type of the subscribed IoTDA instance. This parameter is valid and mandatory only when **charge_mode** is set to **prePaid**. **Options**: • **month** • **year**

Parameter	Mandatory	Type	Description
period_num	No	Integer	**Parameter description**: number of periods for subscribing to an IoTDA instance. This parameter is valid and mandatory only when **charge_mode** is set to **prePaid**. **Value**: If **period_type** is set to **month**, the value ranges from 1 to 9. If **period_type** is set to **year**, the value ranges from 1 to 3. Minimum: **1** Maximum: **9**
is_auto_renew	No	Boolean	**Parameter description**: whether to automatically renew the subscription, specified during yearly/monthly instance creation. The renewal period is the same as the original period, and the payment is automatically made during the renewal. **Options**: • **true** • **false** (default value) Default: **false**
is_auto_pay	No	Boolean	**Parameter description**: whether to automatically pay from the customer's account. This parameter can be specified when a yearly/monthly instance is created. This parameter does not affect the payment mode of automatic renewal. **Value**: **true**: automatic payment. Fees are automatically deducted from the account balance. **false**: default value. Orders are submitted but not paid. Default: **false**

Table 1-513 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	**Parameter description**: tag key. Use letters, digits, spaces, and special characters (_.:=+-@). No space is allowed at the beginning or end. Minimum: **1** Maximum: **128**
value	No	String	**Parameter description**: tag value, which can be an empty string or null. Use letters, digits, spaces, and special characters (_.:=+-@). Minimum: **0** Maximum: **255**

Table 1-514 AdditionalParams

Parameter	Mandatory	Type	Description
vpc_id	Yes	String	**Parameter description**: VPC ID of the instance of the enterprise edition.
subnet_id	Yes	String	**Parameter description**: subnet ID of the instance of the enterprise edition.
security_group_id	Yes	String	**Parameter description**: security group ID of the instance of the enterprise edition. Ensure that port 22 (Linux SSH login), port 3389 (Windows remote login), and ICMP (Ping) have been enabled for the selected security group.
smn_topic_urn	No	String	**Parameter description**: SMN topic URN. When an instance of the enterprise edition is successfully created, the platform sends notifications through this topic.

Parameter	Mandatory	Type	Description
ciphering_algorithm	No	String	**Parameter description**: encryption algorithm supported by the instance. **Options**: • **COMMON_ALGORITHM**: common encryption algorithm (international cryptographic algorithms such as RSA and SHA-256) • **SM_ALGORITHM**: SM series commercial encryption algorithms (Chinese cryptographic algorithms such as SM2, SM3, and SM4) Default: **COMMON_ALGORITHM**
port_info	Yes	**Port** object	Port information of the enterprise edition. This parameter is mandatory when you create an enterprise instance.

Table 1-515 Port

Parameter	Mandatory	Type	Description
app_https_port	No	Integer	**Parameter description**: HTTPS port for application access. The default value is **443**. Minimum: **0** Maximum: **65535** Default: **443**
app_amqps_port	No	Integer	**Parameter description**: AMQP port for application access. The default value is **5671**. Minimum: **0** Maximum: **65535** Default: **5671**

Parameter	Mandatory	Type	Description
device_coap_port	No	Integer	**Parameter description**: CoAP port for device access. The default value is **5683**. Minimum: **0** Maximum: **65535** Default: **5683**
device_coaps_port	No	Integer	**Parameter description**: CoAPS port for device access. The default value is **5684**. Minimum: **0** Maximum: **65535** Default: **5684**
device_mqtt_port	No	Integer	**Parameter description**: MQTT port for device access. The default value is **1883**. Minimum: **0** Maximum: **65535** Default: **1883**
device_mqtts_port	No	Integer	**Parameter description**: MQTTS port for device access. The default value is **8883**. Minimum: **0** Maximum: **65535** Default: **8883**
device_https_port	No	Integer	**Parameter description**: HTTPS port for device access. The default value is **443**. Minimum: **0** Maximum: **65535** Default: **443**

Response Parameters

Status code: 200

Table 1-516 Response body parameters

Parameter	Type	Description
instance_type	String	**Parameter description**: instance type. **Options**: • **standard** • **enterprise**
instance_id	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).
name	String	**Parameter description**: instance name. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
flavor	**Flavor** object	**Parameter description**: specifications of an IoTDA instance.
status	String	**Parameter description**: instance status. **Options**: • **CREATING**: The instance is being created. • **ACTIVE**: The instance is normal. • **FROZEN**: The instance is frozen. • **TRADING**: The instance is in a transaction. • **MODIFYING**: The instance class is being changed. • **FAILED**: The instance fails to be created. Minimum: **0** Maximum: **64**
charge_info	**ChargeInfo** object	**Parameter description**: billing information of an IoTDA instance.
description	String	**Parameter description**: description of an IoTDA instance. **Value**: The value can contain a maximum of 256 characters. Use only letters, digits, and special characters (_.&-).
enterprise_project_id	String	**Parameter description**: enterprise project ID.
tags	Array of **Tag** objects	**Parameter description**: tag information of an IoTDA instance. This field has values if the instance has tags. Or, it is left empty. Array Length: **0 - 20**

Parameter	Type	Description
order_id	String	**Parameter description**: order ID. This parameter is returned when a yearly/monthly instance is created. Minimum: **0** Maximum: **64**
additional_params	**AdditionalParams** object	Additional parameter information of the enterprise edition.

Table 1-517 Flavor

Parameter	Type	Description
type	String	**Parameter description**: specification name of the IoTDA instance to create. For details, see **Specifications**.
size	Integer	**Parameter description**: unit number of the standard IoTDA instances to create. For details, see **Specifications**. This parameter is mandatory when **instance_type** is set to **standard**. Minimum: **1** Maximum: **200**

Table 1-518 ChargeInfo

Parameter	Type	Description
charge_mode	String	**Parameter description**: payment mode of the instance. **Options**: • **prePaid**: yearly/monthly • **postPaid**: pay-per-use
period_type	String	**Parameter description**: period type of the subscribed IoTDA instance. This parameter is valid and mandatory only when **charge_mode** is set to **prePaid**. **Options**: • **month** • **year**

Parameter	Type	Description
period_num	Integer	**Parameter description**: number of periods for subscribing to an IoTDA instance. This parameter is valid and mandatory only when **charge_mode** is set to **prePaid**. **Value**: If **period_type** is set to **month**, the value ranges from 1 to 9. If **period_type** is set to **year**, the value ranges from 1 to 3. Minimum: **1** Maximum: **9**
is_auto_renew	Boolean	**Parameter description**: whether to automatically renew the subscription, specified during yearly/monthly instance creation. The renewal period is the same as the original period, and the payment is automatically made during the renewal. **Options**: • **true** • **false** (default value) Default: **false**
is_auto_pay	Boolean	**Parameter description**: whether to automatically pay from the customer's account. This parameter can be specified when a yearly/monthly instance is created. This parameter does not affect the payment mode of automatic renewal. **Value**: **true**: automatic payment. Fees are automatically deducted from the account balance. **false**: default value. Orders are submitted but not paid. Default: **false**

Table 1-519 Tag

Parameter	Type	Description
key	String	**Parameter description**: tag key. Use letters, digits, spaces, and special characters (_.:=+-@). No space is allowed at the beginning or end. Minimum: **1** Maximum: **128**
value	String	**Parameter description**: tag value, which can be an empty string or null. Use letters, digits, spaces, and special characters (_.:=+-@). Minimum: **0** Maximum: **255**

Table 1-520 AdditionalParams

Parameter	Type	Description
vpc_id	String	**Parameter description**: VPC ID of the instance of the enterprise edition.
subnet_id	String	**Parameter description**: subnet ID of the instance of the enterprise edition.
security_group_id	String	**Parameter description**: security group ID of the instance of the enterprise edition. Ensure that port 22 (Linux SSH login), port 3389 (Windows remote login), and ICMP (Ping) have been enabled for the selected security group.
smn_topic_urn	String	**Parameter description**: SMN topic URN. When an instance of the enterprise edition is successfully created, the platform sends notifications through this topic.
ciphering_algorithm	String	**Parameter description**: encryption algorithm supported by the instance. **Options**: • **COMMON_ALGORITHM**: common encryption algorithm (international cryptographic algorithms such as RSA and SHA-256) • **SM_ALGORITHM**: SM series commercial encryption algorithms (Chinese cryptographic algorithms such as SM2, SM3, and SM4) Default: **COMMON_ALGORITHM**
port_info	**Port** object	Port information of the enterprise edition. This parameter is mandatory when you create an enterprise instance.

Table 1-521 Port

Parameter	Type	Description
app_https_port	Integer	**Parameter description**: HTTPS port for application access. The default value is **443**. Minimum: **0** Maximum: **65535** Default: **443**

Parameter	Type	Description
app_amqps_port	Integer	**Parameter description**: AMQP port for application access. The default value is **5671**. Minimum: **0** Maximum: **65535** Default: **5671**
device_coap_port	Integer	**Parameter description**: CoAP port for device access. The default value is **5683**. Minimum: **0** Maximum: **65535** Default: **5683**
device_coaps_port	Integer	**Parameter description**: CoAPS port for device access. The default value is **5684**. Minimum: **0** Maximum: **65535** Default: **5684**
device_mqtt_port	Integer	**Parameter description**: MQTT port for device access. The default value is **1883**. Minimum: **0** Maximum: **65535** Default: **1883**
device_mqtts_port	Integer	**Parameter description**: MQTTS port for device access. The default value is **8883**. Minimum: **0** Maximum: **65535** Default: **8883**
device_https_port	Integer	**Parameter description**: HTTPS port for device access. The default value is **443**. Minimum: **0** Maximum: **65535** Default: **443**

Example Requests

- Create an instance. Specifications: standard intermediate-frequency units, 2 units, and yearly billing.

```
POST https://{endpoint}/v5/iot/{project_id}/iotda-instances

{
  "instance_type" : "standard",
  "flavor" : {
    "type" : "iotda.standard.s1",
    "size" : 2
```



```
  },
  "name" : "iotda_instance",
  "charge_info" : {
    "charge_mode" : "prePaid",
    "period_type" : "year",
    "period_num" : 1,
    "is_auto_renew" : "true",
    "is_auto_pay" : "false"
  },
  "description" : "IoTDA instance for production.",
  "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d",
  "tags" : [ {
    "key" : "testTagName",
    "value" : "testTagValue"
  } ]
}
```

- Create an instance. Specifications: standard intermediate-frequency units, 2 units, and pay-per-use billing.

```
POST https://{endpoint}/v5/iot/{project_id}/iotda-instances

{
  "instance_type" : "standard",
  "flavor" : {
    "type" : "iotda.standard.s2",
    "size" : 2
  },
  "name" : "iotda_instance",
  "charge_info" : {
    "charge_mode" : "postPaid"
  },
  "description" : "IoTDA instance for production.",
  "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d"
}
```

- Create an instance. Specifications: enterprise edition and yearly billing.

```
POST https://{endpoint}/v5/iot/{project_id}/iotda-instances

{
  "instance_type" : "enterprise",
  "flavor" : {
    "type" : "iotda.enterprise.1000tps.10wonlinedevice"
  },
  "name" : "iotda_enterprise_instance",
  "charge_info" : {
    "charge_mode" : "prePaid",
    "period_type" : "year",
    "period_num" : 1,
    "is_auto_renew" : "true",
    "is_auto_pay" : "false"
  },
  "description" : "IoTDA instance for production.",
  "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d",
  "additional_params" : {
    "vpc_id" : "40926909-d411-45dd-a8b4-1ebc36512345",
    "subnet_id" : "088c7d8a-f49e-4f5f-bd33-ce9b6b712345",
    "security_group_id" : "55980b43-f006-4dbf-ab58-9b0d9e712345",
    "smn_topic_urn" : "urn:smn:cn-
north-7:08aaee8ae000d5182f26c00199812345:iotda_instance_create",
    "ciphering_algorithm" : "COMMON_ALGORITHM",
    "port_info" : {
      "app_https_port" : "443",
      "app_amqps_port" : "5671",
      "device_coap_port" : "5683",
      "device_coaps_port" : "5684",
      "device_mqtt_port" : "1883",
      "device_mqtts_port" : "8883",
      "device_https_port" : "443"
    }
```



```
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "instance_type" : "standard",
  "instance_id" : "8561675c-d8a3-4956-9884-9cf9cbdd3134",
  "flavor" : {
    "type" : "iotda.standard.s2",
    "size" : 2
  },
  "status" : "CREATING",
  "charge_info" : {
    "charge_mode" : "prePaid",
    "period_type" : "year",
    "period_num" : 1,
    "is_auto_renew" : true,
    "is_auto_pay" : false
  },
  "description" : "IoTDA instance for production.",
  "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d",
  "tags" : [ {
    "key" : "testTagName",
    "value" : "testTagValue"
  } ],
  "order_id" : "CS22121614500ABCD",
  "additional_params" : null
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.14.2 This API is used to query the instance list.

Function

This API is used to query the list of IoTDA instances.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/iotda-instances

Table 1-522 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.

Table 1-523 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. - To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. Value: an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: an integer ranging from 1 to 500. The default value is **500**. Minimum: **1** Maximum: **500** Default: **500**

Parameter	Mandatory	Type	Description
marker	No	String	ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. Default: **ffffffffffffffff**
name	No	String	**Parameter description**: IoTDA instance name. The fuzzy match is used. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
instance_type	No	String	**Parameter description**: instance type. **Options**: • **standard** • **enterprise**

Request Parameters

Table 1-524 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Response Parameters

Status code: 200

Table 1-525 Response body parameters

Parameter	Type	Description
count	Integer	Total number of IoTDA instances. Minimum: **1** Maximum: **10000**
marker	String	ID of the last record in this query, which can be used in the next query. Minimum: **0** Maximum: **20**
instances	Array of **QueryInstanceSimplify** objects	Details list of IoTDA instances. Array Length: **0 - 10000**

Table 1-526 QueryInstanceSimplify

Parameter	Type	Description
instance_type	String	**Parameter description**: instance type. **Options**: • **standard** • **enterprise**
instance_id	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).
name	String	**Parameter description**: instance name. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
charge_mode	String	**Parameter description**: payment mode of the instance. **Options**: • **prePaid**: yearly/monthly • **postPaid**: pay-per-use
flavor	**Flavor** object	**Parameter description**: specifications of an IoTDA instance.
status	String	**Parameter description**: instance status. **Options**: • **CREATING**: The instance is being created. • **ACTIVE**: The instance is normal. • **FROZEN**: The instance is frozen. • **TRADING**: The instance is in a transaction. • **MODIFYING**: The instance class is being changed. • **FAILED**: The instance fails to be created. Minimum: **0** Maximum: **64**
create_time	String	**Parameter description**: time when the instance is created. Example: 2023-01-28T06:57:52Z. Minimum: **0** Maximum: **64**
update_time	String	**Parameter description**: last update time of the instance. Example: 2023-01-28T06:57:52Z. Minimum: **0** Maximum: **64**
enterprise_project_id	String	**Parameter description**: enterprise project ID.

Table 1-527 Flavor

Parameter	Type	Description
type	String	**Parameter description**: specification name of the IoTDA instance to create. For details, see **Specifications**.
size	Integer	**Parameter description**: unit number of the standard IoTDA instances to create. For details, see **Specifications**. This parameter is mandatory when **instance_type** is set to **standard**. Minimum: **1** Maximum: **200**

Example Requests

This API is used to query all instances in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/iotda-instances
```

Example Responses

Status code: 200

OK

```
{
  "count" : 1,
  "marker" : 1805431584415871305,
  "instances" : [ {
    "instance_type" : "standard",
    "instance_id" : "8561675c-d8a3-4956-9884-9cf9cbdd3134",
    "name" : "iotda_instance",
    "flavor" : {
      "type" : "iotda.standard.s1",
      "size" : 1
    },
    "status" : "ACTIVE",
    "create_time" : "2023-01-28T06:57:52Z",
    "update_time" : "2023-01-28T06:57:52Z",
    "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request

Status Code	Description
401	Unauthorized

Error Codes

See **Error Codes**.

1.4.14.3 Query Instance Details

Function

Querying details about an IoTDA instance.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/iotda-instances/{instance_id}

Table 1-528 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-529 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Response Parameters

Status code: 200

Table 1-530 Response body parameters

Parameter	Type	Description
instance_type	String	**Parameter description**: instance type. **Options**: • **standard** • **enterprise**
instance_id	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).
charge_mode	String	**Parameter description**: payment mode of the instance. **Options**: • **prePaid**: yearly/monthly • **postPaid**: pay-per-use
name	String	**Parameter description**: instance name. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
flavor	**Flavor** object	**Parameter description**: specifications of an IoTDA instance.

Parameter	Type	Description
status	String	**Parameter description**: instance status. **Options**: • **CREATING**: The instance is being created. • **ACTIVE**: The instance is normal. • **FROZEN**: The instance is frozen. • **TRADING**: The instance is in a transaction. • **MODIFYING**: The instance class is being changed. • **FAILED**: The instance fails to be created. Minimum: **0** Maximum: **64**
description	String	**Parameter description**: description of an IoTDA instance. **Value**: The value can contain a maximum of 256 characters. Use only letters, digits, and special characters (_,.&-).
access_infos	Array of **AccessInfo** objects	**Parameter description**: access information of an IoTDA instance. Array Length: **0 - 10**
create_time	String	**Parameter description**: time when the instance is created. Example: 2023-01-28T06:57:52Z. Minimum: **0** Maximum: **64**
update_time	String	**Parameter description**: last update time of the instance. Example: 2023-01-28T06:57:52Z. Minimum: **0** Maximum: **64**
enterprise_project_id	String	**Parameter description**: enterprise project ID.
tags	Array of **Tag** objects	**Parameter description**: tag information of an IoTDA instance. This field has values if the instance has tags. Or, it is left empty. Array Length: **0 - 20**
order_id	String	**Parameter description**: order ID. This parameter is returned when a yearly/monthly instance is created. Minimum: **0** Maximum: **64**

Parameter	Type	Description
operate_window	**OperateWindow** object	Instance maintenance time window. You can specify the time window during which the specifications are modified.
additional_params	**AdditionalParamsResp** object	Additional parameter information of the enterprise edition.

Table 1-531 Flavor

Parameter	Type	Description
type	String	**Parameter description**: specification name of the IoTDA instance to create. For details, see **Specifications**.
size	Integer	**Parameter description**: unit number of the standard IoTDA instances to create. For details, see **Specifications**. This parameter is mandatory when **instance_type** is set to **standard**. Minimum: **1** Maximum: **200**

Table 1-532 AccessInfo

Parameter	Type	Description
type	String	**Parameter description**: access address type. For example, if HTTPS is used for application access, the value is **APP_HTTPS**. If MQTT is used for device access, the value is **DEVICE_MQTT**. **Options**: • **APP_HTTPS**: HTTPS for application access • **APP_AMQP**: AMQP for application access • **APP_MQTT**: MQTT for application access • **DEVICE_COAP**: CoAP for device access • **DEVICE_MQTT**: MQTT for device access • **DEVICE_HTTPS**: HTTPS for device access
port	Integer	**Parameter description**: secure access port of the application or device of the instance. Minimum: **0** Maximum: **65535**

Parameter	Type	Description
non_tls_port	Integer	**Parameter description**: non-secure access port of the application or device of the instance. If null is returned, the access address of this type does not support non-secure port access. Minimum: **0** Maximum: **65535**
websocket_port	Integer	**Parameter description**: WebSocket-based MQTT access port. If null is returned, the access address of this type does not support WebSocket port access. Minimum: **0** Maximum: **65535**
domain_name	String	**Parameter description**: access domain name of the instance.
private_addresses	Array of strings	**Parameter description**: private network access address list of the instance. Array Length: **0 - 10**
public_address	Array of strings	**Parameter description**: public access address of the instance. Array Length: **0 - 10**
ipv6_address	Array of strings	**Parameter description**: IPv6 access address list of the instance. Array Length: **0 - 10**
ip_whitelist	**IPWhiteList** object	IP address whitelist. The IP address whitelist can be configured only for the APP_HTTPS protocol of the enterprise instances.

Table 1-533 IPWhiteList

Parameter	Type	Description
enable	Boolean	**Parameter description**: whether to enable the IP address whitelist access control.
allow_list	Array of **IpAllowList** objects	List of IP addresses that are allowed to access the enterprise instances. Array Length: **0 - 15**

Table 1-534 IpAllowList

Parameter	Type	Description
address	String	**Parameter description**: whitelist IP address. Minimum: **7** Maximum: **128**
description	String	**Parameter description**: description. Minimum: **0** Maximum: **128**

Table 1-535 Tag

Parameter	Type	Description
key	String	**Parameter description**: tag key. Use letters, digits, spaces, and special characters (_.:=+-@). No space is allowed at the beginning or end. Minimum: **1** Maximum: **128**
value	String	**Parameter description**: tag value, which can be an empty string or null. Use letters, digits, spaces, and special characters (_.:=+-@). Minimum: **0** Maximum: **255**

Table 1-536 OperateWindow

Parameter	Type	Description
start_time	String	**Parameter description**: start time of the change time window. The value is UTC time in HH:mm format.
end_time	String	**Parameter description**: end time of the change time window. The value is UTC time in HH:mm format.

Table 1-537 AdditionalParamsResp

Parameter	Type	Description
vpc_id	String	**Parameter description**: VPC ID of the instance of the enterprise edition. Minimum: **0** Maximum: **64**
subnet_id	String	**Parameter description**: subnet ID of the instance of the enterprise edition. Minimum: **0** Maximum: **64**
security_group_id	String	**Parameter description**: security group ID of the instance of the enterprise edition. Minimum: **0** Maximum: **64**
ciphering_algorithm	String	**Parameter description**: encryption algorithm supported by the instance. **Options**: • **COMMON_ALGORITHM**: common encryption algorithm (international cryptographic algorithms such as RSA and SHA-256) • **SM_ALGORITHM**: SM series commercial encryption algorithms (Chinese cryptographic algorithms such as SM2, SM3, and SM4) Default: **COMMON_ALGORITHM**
forwarding_info	**ForwardingInfo** object	SNAT configuration information of the instance of the enterprise edition.

Table 1-538 ForwardingInfo

Parameter	Type	Description
eip	String	**Parameter description**: EIP bound to the NAT gateway. Minimum: **7** Maximum: **128**
enable_snat	Boolean	**Parameter description**: whether to enable the SNAT configuration. **Options**: • **true** • **false**

Example Requests

Querying an instance.

```
GET https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}
```

Example Responses

Status code: 200

OK


```
{
  "instance_type" : "standard",
  "instance_id" : "8561675c-d8a3-4956-9884-9cf9cbdd3134",
  "charge_mode" : "prePaid",
  "name" : "iotda_instance",
  "flavor" : {
    "type" : "iotda.standard.s1",
    "size" : 1
  },
  "status" : "ACTIVE",
  "create_time" : "2023-01-28T06:57:52Z",
  "update_time" : "2023-01-28T06:58:52Z",
  "description" : "IoTDA instance for production.",
  "access_infos" : [ {
    "type" : "APP_AMQP",
    "port" : 5671,
    "non_tls_port" : null,
    "websocket_port" : null,
    "domain_name" : "example.myhuaweicloud.com",
    "public_address" : [ ],
    "ipv6_address" : [ ],
    "ip_whitelist" : null
  }, {
    "type" : "APP_HTTPS",
    "port" : 443,
    "non_tls_port" : null,
    "websocket_port" : null,
    "domain_name" : "example1.myhuaweicloud.com",
    "public_address" : [ ],
    "ipv6_address" : [ ],
    "ip_whitelist" : null
  }, {
    "type" : "DEVICE_MQTT",
    "port" : 8883,
    "non_tls_port" : 1883,
    "websocket_port" : 443,
    "domain_name" : "example2.myhuaweicloud.com",
    "public_address" : [ ],
    "ipv6_address" : [ ],
    "ip_whitelist" : null
  }, {
    "type" : "DEVICE_COAP",
    "port" : 5684,
    "non_tls_port" : 5683,
    "websocket_port" : null,
    "domain_name" : "example3.myhuaweicloud.com",
    "public_address" : [ ],
    "ipv6_address" : [ ],
    "ip_whitelist" : null
  } ],
  "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d",
  "tags" : [ {
    "key" : "testTagName",
    "value" : "testTagValue"
  } ],
  "order_id" : "CS22121614500ABCD",
```



```
  "operate_window" : {
    "start_time" : "22:00",
    "end_time" : "24:00"
  },
  "additional_params" : null
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized

Error Codes

See **Error Codes**.

1.4.14.4 Modify Instance Information

Function

Modifying IoTDA instance information.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/iotda-instances/{instance_id}

Table 1-539 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-540 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Table 1-541 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	**Parameter description**: instance name. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
description	No	String	**Parameter description**: description of an IoTDA instance. **Value**: The value can contain a maximum of 256 characters. Use only letters, digits, and special characters (_,.&-).
operate_window	No	**OperateWindow** object	Instance maintenance time window. You can specify the time window during which the specifications are modified.

Parameter	Mandatory	Type	Description
forwarding_info	No	**UpdateForwardingInfo** object	SNAT configuration of the enterprise instance. After the configuration is enabled, the enterprise instance can communicate externally over public networks. Only enterprise instances support data transfer configuration modification.
access_info	No	**UpdateAccessInfo** object	Access information of IoTDA instances. Used for connecting applications and devices to the IoT platform. Only enterprise instances support custom access information.

Table 1-542 OperateWindow

Parameter	Mandatory	Type	Description
start_time	Yes	String	**Parameter description**: start time of the change time window. The value is UTC time in HH:mm format.
end_time	Yes	String	**Parameter description**: end time of the change time window. The value is UTC time in HH:mm format.

Table 1-543 UpdateForwardingInfo

Parameter	Mandatory	Type	Description
enable_snat	Yes	Boolean	**Parameter description**: whether to enable the SNAT configuration. After SNAT is enabled for an enterprise instance, the instance can communicate externally over public networks. SNAT can be configured only for enterprise instances. After SNAT is enabled, it cannot be disabled. **Options**: • **true**

Table 1-544 UpdateAccessInfo

Parameter	Mandatory	Type	Description
access_type	Yes	String	**Parameter description**: access address type. For example, if HTTPS is used for application access, the value is **APP_HTTPS**. If MQTT is used for device access, the value is **DEVICE_MQTT**. **Options**: • **APP_HTTPS**: HTTPS for application access • **APP_AMQP**: AMQP for application access • **APP_MQTT**: MQTT for application access • **DEVICE_COAP**: CoAP for device access • **DEVICE_MQTT**: MQTT for device access • **DEVICE_HTTPS**: HTTPS for device access
domain_name	No	String	**Parameter description**: access domain name. If the domain name needs to be updated, this parameter is carried.
public_addresses_enable	No	Boolean	**Parameter description**: whether to configure the public network access address. The value can be **true** or **false**. **Options**: • **true**: Configure a public network access address, which is automatically allocated by the platform. After an address is allocated, it cannot be modified or deleted.
ip_whitelist	No	**IPWhiteList** object	IP address whitelist. The IP address whitelist can be configured only for the APP_HTTPS protocol of the enterprise instances.

Table 1-545 IPWhiteList

Parameter	Mandatory	Type	Description
enable	Yes	Boolean	**Parameter description**: whether to enable the IP address whitelist access control.
allow_list	No	Array of **IpAllowList** objects	List of IP addresses that are allowed to access the enterprise instances. Array Length: **0 - 15**

Table 1-546 IpAllowList

Parameter	Mandatory	Type	Description
address	Yes	String	**Parameter description**: whitelist IP address. Minimum: **7** Maximum: **128**
description	No	String	**Parameter description**: description. Minimum: **0** Maximum: **128**

Response Parameters

Status code: 200

Table 1-547 Response body parameters

Parameter	Type	Description
instance_type	String	**Parameter description**: instance type. **Options**: • **standard** • **enterprise**
instance_id	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Parameter	Type	Description
charge_mode	String	**Parameter description**: payment mode of the instance. **Options**: • **prePaid**: yearly/monthly • **postPaid**: pay-per-use
name	String	**Parameter description**: instance name. **Value**: Enter 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
flavor	**Flavor** object	**Parameter description**: specifications of an IoTDA instance.
status	String	**Parameter description**: instance status. **Options**: • **CREATING**: The instance is being created. • **ACTIVE**: The instance is normal. • **FROZEN**: The instance is frozen. • **TRADING**: The instance is in a transaction. • **MODIFYING**: The instance class is being changed. • **FAILED**: The instance fails to be created. Minimum: **0** Maximum: **64**
description	String	**Parameter description**: description of an IoTDA instance. **Value**: The value can contain a maximum of 256 characters. Use only letters, digits, and special characters (_,.&-).
access_infos	Array of **AccessInfo** objects	**Parameter description**: access information of an IoTDA instance. Array Length: **0 - 10**
create_time	String	**Parameter description**: time when the instance is created. Example: 2023-01-28T06:57:52Z. Minimum: **0** Maximum: **64**
update_time	String	**Parameter description**: last update time of the instance. Example: 2023-01-28T06:57:52Z. Minimum: **0** Maximum: **64**
enterprise_project_id	String	**Parameter description**: enterprise project ID.

Parameter	Type	Description
tags	Array of **Tag** objects	**Parameter description**: tag information of an IoTDA instance. This field has values if the instance has tags. Or, it is left empty. Array Length: **0 - 20**
order_id	String	**Parameter description**: order ID. This parameter is returned when a yearly/monthly instance is created. Minimum: **0** Maximum: **64**
operate_window	**OperateWindow** object	Instance maintenance time window. You can specify the time window during which the specifications are modified.
additional_params	**AdditionalParamsResp** object	Additional parameter information of the enterprise edition.

Table 1-548 Flavor

Parameter	Type	Description
type	String	**Parameter description**: specification name of the IoTDA instance to create. For details, see **Specifications**.
size	Integer	**Parameter description**: unit number of the standard IoTDA instances to create. For details, see **Specifications**. This parameter is mandatory when **instance_type** is set to **standard**. Minimum: **1** Maximum: **200**

Table 1-549 AccessInfo

Parameter	Type	Description
type	String	**Parameter description**: access address type. For example, if HTTPS is used for application access, the value is **APP_HTTPS**. If MQTT is used for device access, the value is **DEVICE_MQTT**. **Options**: • **APP_HTTPS**: HTTPS for application access • **APP_AMQP**: AMQP for application access • **APP_MQTT**: MQTT for application access • **DEVICE_COAP**: CoAP for device access • **DEVICE_MQTT**: MQTT for device access • **DEVICE_HTTPS**: HTTPS for device access
port	Integer	**Parameter description**: secure access port of the application or device of the instance. Minimum: **0** Maximum: **65535**
non_tls_port	Integer	**Parameter description**: non-secure access port of the application or device of the instance. If null is returned, the access address of this type does not support non-secure port access. Minimum: **0** Maximum: **65535**
websocket_port	Integer	**Parameter description**: WebSocket-based MQTT access port. If null is returned, the access address of this type does not support WebSocket port access. Minimum: **0** Maximum: **65535**
domain_name	String	**Parameter description**: access domain name of the instance.
private_addresses	Array of strings	**Parameter description**: private network access address list of the instance. Array Length: **0 - 10**
public_address	Array of strings	**Parameter description**: public access address of the instance. Array Length: **0 - 10**
ipv6_address	Array of strings	**Parameter description**: IPv6 access address list of the instance. Array Length: **0 - 10**

Parameter	Type	Description
ip_whitelist	**IPWhiteList** object	IP address whitelist. The IP address whitelist can be configured only for the APP_HTTPS protocol of the enterprise instances.

Table 1-550 IPWhiteList

Parameter	Type	Description
enable	Boolean	**Parameter description**: whether to enable the IP address whitelist access control.
allow_list	Array of **IpAllowList** objects	List of IP addresses that are allowed to access the enterprise instances. Array Length: **0 - 15**

Table 1-551 IpAllowList

Parameter	Type	Description
address	String	**Parameter description**: whitelist IP address. Minimum: **7** Maximum: **128**
description	String	**Parameter description**: description. Minimum: **0** Maximum: **128**

Table 1-552 Tag

Parameter	Type	Description
key	String	**Parameter description**: tag key. Use letters, digits, spaces, and special characters (_.:=+-@). No space is allowed at the beginning or end. Minimum: **1** Maximum: **128**
value	String	**Parameter description**: tag value, which can be an empty string or null. Use letters, digits, spaces, and special characters (_.:=+-@). Minimum: **0** Maximum: **255**

Table 1-553 OperateWindow

Parameter	Type	Description
start_time	String	**Parameter description**: start time of the change time window. The value is UTC time in HH:mm format.
end_time	String	**Parameter description**: end time of the change time window. The value is UTC time in HH:mm format.

Table 1-554 AdditionalParamsResp

Parameter	Type	Description
vpc_id	String	**Parameter description**: VPC ID of the instance of the enterprise edition. Minimum: **0** Maximum: **64**
subnet_id	String	**Parameter description**: subnet ID of the instance of the enterprise edition. Minimum: **0** Maximum: **64**
security_group_id	String	**Parameter description**: security group ID of the instance of the enterprise edition. Minimum: **0** Maximum: **64**
ciphering_algorithm	String	**Parameter description**: encryption algorithm supported by the instance. **Options**: • **COMMON_ALGORITHM**: common encryption algorithm (international cryptographic algorithms such as RSA and SHA-256) • **SM_ALGORITHM**: SM series commercial encryption algorithms (Chinese cryptographic algorithms such as SM2, SM3, and SM4) Default: **COMMON_ALGORITHM**
forwarding_info	**ForwardingInfo** object	SNAT configuration information of the instance of the enterprise edition.

Table 1-555 ForwardingInfo

Parameter	Type	Description
eip	String	**Parameter description**: EIP bound to the NAT gateway. Minimum: **7** Maximum: **128**
enable_snat	Boolean	**Parameter description**: whether to enable the SNAT configuration. **Options**: • **true** • **false**

Example Requests

- Modifying the instance name, description, and operation time window.

```
PUT https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}

{
  "name" : "iotda_instance01",
  "description" : "IoTDA instance for test.",
  "operate_window" : {
    "start_time" : "18:00",
    "end_time" : "22:00"
  }
}
```

- Enabling SNAT configuration on data transfer of public networks for enterprise instances.

```
PUT https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}

{
  "forwarding_info" : {
    "enable_snat" : "true"
  }
}
```

- Modifying the access information of an enterprise instance.

```
PUT https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}

{
  "access_info" : {
    "access_type" : "APP_HTTPS",
    "domain_name" : "example.myhuaweicloud.com",
    "public_addresses_enable" : true
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "instance_type" : "standard",
  "instance_id" : "8561675c-d8a3-4956-9884-9cf9cbdd3134",
  "charge_mode" : "prePaid",
  "name" : "iotda_instance",
```



```
 "flavor" : {
   "type" : "iotda.standard.s1",
   "size" : 1
 },
 "status" : "ACTIVE",
 "create_time" : "2023-01-28T06:57:52Z",
 "update_time" : "2023-01-28T06:58:52Z",
 "description" : "IoTDA instance for production.",
 "access_infos" : [ {
   "type" : "APP_AMQP",
   "port" : 5671,
   "non_tls_port" : null,
   "websocket_port" : null,
   "domain_name" : "example.myhuaweicloud.com",
   "public_address" : [ ],
   "ipv6_address" : [ ],
   "ip_whitelist" : null
 }, {
   "type" : "APP_HTTPS",
   "port" : 443,
   "non_tls_port" : null,
   "websocket_port" : null,
   "domain_name" : "example1.myhuaweicloud.com",
   "public_address" : [ ],
   "ipv6_address" : [ ],
   "ip_whitelist" : null
 }, {
   "type" : "DEVICE_MQTT",
   "port" : 8883,
   "non_tls_port" : 1883,
   "websocket_port" : 443,
   "domain_name" : "example2.myhuaweicloud.com",
   "public_address" : [ ],
   "ipv6_address" : [ ],
   "ip_whitelist" : null
 }, {
   "type" : "DEVICE_COAP",
   "port" : 5684,
   "non_tls_port" : 5683,
   "websocket_port" : null,
   "domain_name" : "example3.myhuaweicloud.com",
   "public_address" : [ ],
   "ipv6_address" : [ ],
   "ip_whitelist" : null
 } ],
 "enterprise_project_id" : "d22e47e9-cfad-4254-8a29-d2a56a07681d",
 "tags" : [ {
   "key" : "testTagName",
   "value" : "testTagValue"
 } ],
 "order_id" : "CS22121614500ABCD",
 "operate_window" : {
   "start_time" : "22:00",
   "end_time" : "24:00"
 },
 "additional_params" : null
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request

Status Code	Description
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.14.5 Delete an Instance

Function

This API is used to delete an IoTDA instance with the pay-per-use billing mode.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/iotda-instances/{instance_id}

Table 1-556 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-557 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Response Parameters

None

Example Requests

Deleting a specified instance.

```
DELETE https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.14.6 Modify Instance Specifications

Function

Modifying the specifications of the IoTDA instance.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/iotda-instances/{instance_id}/resize

Table 1-558 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-559 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Table 1-560 Request body parameters

Parameter	Mandatory	Type	Description
flavor	Yes	**Flavor** object	**Parameter description**: specifications of an IoTDA instance.
is_auto_pay	No	Boolean	**Parameter description**: whether to automatically pay from the customer's account. This parameter can be specified during specifications change of a yearly/monthly instance. This parameter does not affect the payment mode of automatic renewal. **Value**: **true**: automatic payment. Fees are automatically deducted from the account balance. **false**: default value. Orders are submitted but not paid. Default: **false**
delay	No	Boolean	**Parameter description**: whether to changing the billing information of a device instance in the delayed mode. To delay the change, you need to set the change time window of the instance first. **Options**: • **true**: The change is delayed. The specification change task will be executed in the specified change time window. • **false**: The specification change task is executed immediately. Default: **false**

Table 1-561 Flavor

Parameter	Mandatory	Type	Description
type	Yes	String	**Parameter description**: specification name of the IoTDA instance to create. For details, see **Specifications**.

Parameter	Mandatory	Type	Description
size	No	Integer	**Parameter description**: unit number of the standard IoTDA instances to create. For details, see **Specifications**. This parameter is mandatory when **instance_type** is set to **standard**. Minimum: **1** Maximum: **200**

Response Parameters

Status code: 200

Table 1-562 Response body parameters

Parameter	Type	Description
order_id	String	**Parameter description**: order ID. This parameter is returned when a yearly/monthly instance is modified. This parameter is left blank when a pay-per-use instance is modified. For details about how to check the order details, see [**Querying Order Details**]. Minimum: **0** Maximum: **64**

Example Requests

Modifying the specifications of a standard instance. Target: 2 standard intermediate-frequency units.

```
PUT https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}/resize

{
  "flavor" : {
    "type" : "iotda.standard.s2",
    "size" : 2
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "order_id" : "CS22121614500ABCD"
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.14.7 Change the Billing Mode of an Instance

Function

Changing the billing mode of an IoTDA instance. The pay-per-use billing mode can be changed to yearly/monthly. Available only when the instance specifications support the yearly/monthly billing mode. For details about the supported instance specifications, see **Specifications**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/iotda-instances/{instance_id}/change-charge-mode

Table 1-563 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-564 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Table 1-565 Request body parameters

Parameter	Mandatory	Type	Description
period_type	Yes	String	**Parameter description**: period type of the subscribed IoTDA instance. This parameter is valid and mandatory only when **charge_mode** is set to **prePaid**. **Options**: • **month** • **year**
period_num	Yes	Integer	**Parameter description**: number of periods for subscribing to an IoTDA instance. This parameter is valid and mandatory only when **charge_mode** is set to **prePaid**. **Value**: If **period_type** is set to **month**, the value ranges from 1 to 9. If **period_type** is set to **year**, the value ranges from 1 to 3. Minimum: **1** Maximum: **9**

Parameter	Mandatory	Type	Description
is_auto_renew	No	Boolean	**Parameter description**: whether to automatically renew the subscription, specified during yearly/monthly instance creation. The renewal period is the same as the original period, and the payment is automatically made during the renewal. **Options**: • **true** • **false** (default value) Default: **false**
is_auto_pay	No	Boolean	**Parameter description**: whether to automatically pay from the customer's account. This parameter can be specified when a yearly/monthly instance is created. This parameter does not affect the payment mode of automatic renewal. **Value**: **true**: automatic payment. Fees are automatically deducted from the account balance. **false**: default value. Orders are submitted but not paid. Default: **false**

Response Parameters

Status code: 200

Table 1-566 Response body parameters

Parameter	Type	Description
order_id	String	**Parameter description**: order ID. Minimum: **0** Maximum: **64**

Example Requests

Changing the billing mode of the instance to yearly. After the order expires, the instance is automatically renewed.


```
PUT https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}/change-charge-mode

{
  "period_type" : "year",
  "period_num" : 1,
  "is_auto_renew" : true
}
```

Example Responses

Status code: 200

OK

```
{
  "order_id" : "CS22121614500ABCD"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.14.8 Adding an Instance Tag

Function

Adding an instance tag.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/iotda-instances/{instance_id}/bind-tags

Table 1-567 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-568 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Table 1-569 Request body parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of **Tag** objects	Instance tags. Array Length: **1 - 20**

Table 1-570 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	**Parameter description**: tag key. Use letters, digits, spaces, and special characters (_.:=+-@). No space is allowed at the beginning or end. Minimum: **1** Maximum: **128**
value	No	String	**Parameter description**: tag value, which can be an empty string or null. Use letters, digits, spaces, and special characters (_.:=+-@). Minimum: **0** Maximum: **255**

Response Parameters

None

Example Requests

```
POST https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}/bind-tags

{
  "tags" : [ {
    "key" : "testTagName",
    "value" : "testTagValue"
  } ]
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.14.9 Delete an Instance Tag

Function

Deleting an instance tag.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/iotda-instances/{instance_id}/unbind-tags

Table 1-571 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details, see **Obtaining a Project ID**.
instance_id	Yes	String	**Parameter description**: instance ID. **Value**: Enter up to 36 characters, including lowercase letters (a-f), digits, and hyphens (-).

Request Parameters

Table 1-572 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**. Minimum: **0** Maximum: **1024000**

Table 1-573 Request body parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of **Tag** objects	Instance tags. Array Length: **1 - 20**

Table 1-574 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	**Parameter description**: tag key. Use letters, digits, spaces, and special characters (_.:=+-@). No space is allowed at the beginning or end. Minimum: **1** Maximum: **128**
value	No	String	**Parameter description**: tag value, which can be an empty string or null. Use letters, digits, spaces, and special characters (_.:=+-@). Minimum: **0** Maximum: **255**

Response Parameters

None

Example Requests

```
POST https://{endpoint}/v5/iot/{project_id}/iotda-instances/{instance_id}/unbind-tags

{
  "tags" : [ {
    "key" : "testTagName",
    "value" : "testTagValue"
  } ]
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden

Error Codes

See **Error Codes**.

1.4.15 Resource Space Management

A resource space is a top-layer concept designed for isolating resources of a single account on the IoT platform. These resources include devices, products, and rules.

1.4.15.1 Query the Resource Space List

Function

The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. This API is used by an application to query the resource space list.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/apps

Table 1-575 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**. Minimum: **0** Maximum: **32**

Table 1-576 Query Parameters

Parameter	Mandatory	Type	Description
default_app	No	Boolean	**Parameter description**: default resource space identifier. If this parameter is not carried, all resource spaces are queried. **Options**: • **true**: queries the default resource space. • **false**: queries the non-default resource space.

Request Parameters

Table 1-577 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. The token can be obtained by calling the IAM API. Minimum: **0** Maximum: **1024000**

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **0** Maximum: **36**

Response Parameters

Status code: 200

Table 1-578 Response body parameters

Parameter	Type	Description
applications	Array of **ApplicationDTO** objects	Resource space list. Array Length: **0 - 500**

Table 1-579 ApplicationDTO

Parameter	Type	Description
app_id	String	Resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. Minimum: **1** Maximum: **64**
app_name	String	Resource space name. Minimum: **1** Maximum: **64**

Parameter	Type	Description
create_time	String	Time when the resource space was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. Minimum: **1** Maximum: **64**
default_app	Boolean	Whether the resource space is the default resource space.

Example Requests

Queries resources in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/apps
```

Example Responses

Status code: 200

Successful response

```
{
  "applications" : [ {
    "app_id" : "0ab87ceecbfc49acbcc8d5acdef3c68c",
    "app_name" : "testApp",
    "create_time" : "20151212T121212Z",
    "default_app" : true
  } ]
}
```

Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.15.2 Create a Resource Space

Function

The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. This API is used by an application to create a resource space.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/apps

Table 1-580 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**. Minimum: **0** Maximum: **32**

Request Parameters

Table 1-581 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. The token can be obtained by calling the IAM API. Minimum: **0** Maximum: **1024000**

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **0** Maximum: **36**

Table 1-582 Request body parameters

Parameter	Mandatory	Type	Description
app_name	Yes	String	**Parameter description**: resource space name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **1** Maximum: **64**

Response Parameters

Status code: 201

Table 1-583 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. Minimum: **1** Maximum: **64**
app_name	String	Resource space name. Minimum: **1** Maximum: **64**
create_time	String	Time when the resource space was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. Minimum: **1** Maximum: **64**
default_app	Boolean	Whether the resource space is the default resource space.

Example Requests

Creates a resource space named **testapp**.

```
POST https://{endpoint}/v5/iot/{project_id}/apps

{
  "app_name" : "testApp"
}
```

Example Responses

Status code: 201

Created

```
{
  "app_id" : "0ab87ceecbfc49acbcc8d5acdef3c68c",
  "app_name" : "testApp",
  "create_time" : "20151212T121212Z",
  "default_app" : true
}
```


Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.15.3 Query a Resource Space

Function

The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. This API is used by an application to query details about a specific resource space.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/apps/{app_id}

Table 1-584 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**. Minimum: **0** Maximum: **32**

Parameter	Mandatory	Type	Description
app_id	Yes	String	**Parameter description**: resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-585 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. The token can be obtained by calling the IAM API. Minimum: **0** Maximum: **1024000**
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **0** Maximum: **36**

Response Parameters

Status code: 200

Table 1-586 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. Minimum: **1** Maximum: **64**
app_name	String	Resource space name. Minimum: **1** Maximum: **64**
create_time	String	Time when the resource space was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. Minimum: **1** Maximum: **64**
default_app	Boolean	Whether the resource space is the default resource space.

Example Requests

Queries resource space details.

```
GET https://{endpoint}/v5/iot/{project_id}/apps/{app_id}
```

Example Responses

Status code: 200

OK

```
{
  "app_id" : "0ab87ceecbfc49acbcc8d5acdef3c68c",
  "app_name" : "testApp",
  "create_time" : "20151212T121212Z",
  "default_app" : true
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.15.4 Delete a Resource Space

Function

This API is used to delete a specific resource space. Deleting a resource space is a high-risk operation. After the resource space is deleted, resources such as products and devices in the space will be unavailable. Exercise caution when performing this operation.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/apps/{app_id}

Table 1-587 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**. Minimum: **0** Maximum: **32**

Parameter	Mandatory	Type	Description
app_id	Yes	String	**Parameter description**: resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-588 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. The token can be obtained by calling the IAM API. Minimum: **0** Maximum: **1024000**
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **0** Maximum: **36**

Response Parameters

None

Example Requests

Deleting a resource space.

```
DELETE https://{endpoint}/v5/iot/{project_id}/apps/{app_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.15.5 Update Resource Spaces

Function

This API is used by an application to update the name of a resource space.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/apps/{app_id}

Table 1-589 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**. Minimum: **0** Maximum: **32**
app_id	Yes	String	**Parameter description**: resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-590 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. The token can be obtained by calling the IAM API. Minimum: **0** Maximum: **1024000**

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **0** Maximum: **36**

Table 1-591 Request body parameters

Parameter	Mandatory	Type	Description
app_name	No	String	**Parameter description**: resource space name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **1** Maximum: **64**

Response Parameters

Status code: 200

Table 1-592 Response body parameters

Parameter	Type	Description
app_id	String	Resource space ID, which uniquely identifies a resource space and is allocated by the platform during resource space creation. The resource space corresponds to the original application of the platform. The meaning of the resource space in the platform is the same as that of the application. The only difference lies in the name. Minimum: **1** Maximum: **64**
app_name	String	Resource space name. Minimum: **1** Maximum: **64**
create_time	String	Time when the resource space was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**. Minimum: **1** Maximum: **64**
default_app	Boolean	Whether the resource space is the default resource space.

Example Requests

```
PUT https://{endpoint}/v5/iot/{project_id}/apps/{app_id}

{
  "app_name" : "testApp"
}
```

Example Responses

Status code: 200

OK

```
{
  "app_id" : "0ab87ceecbfc49acbcc8d5acdef3c68c",
  "app_name" : "testApp",
  "create_time" : "20151212T121212Z",
  "default_app" : true
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16 Batch Task

Batch task APIs are used by applications to perform batch operations on devices connected to the platform.

1.4.16.1 Create a Batch Task

Function

This API is used by an application to create a batch task to perform batch operations on multiple devices. You can upgrade software and firmware, create, modify, delete, freeze, and unfreeze devices, and create command and message tasks in batches.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/batchtasks

Table 1-593 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-594 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Table 1-595 Request body parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the batch task to create will belong. If this parameter is not specified, the batch task to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
task_name	Yes	String	**Parameter description**: batch task name. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, and underscores (_) are allowed. Minimum: **1** Maximum: **128**

Parameter	Mandatory	Type	Description
task_type	Yes	String	**Parameter description**: batch task type. **Options**: • **softwareUpgrade**: software upgrade task • **firmwareUpgrade**: firmware upgrade task • **createDevices**: batch device creation task • **deleteDevices**: batch device deletion task • **freezeDevices**: batch device freezing task • **unfreezeDevices**: batch device unfreezing task • **createCommands**: task for creating synchronous commands in batches • **createAsyncCommands**: task for creating asynchronous commands in batches • **createMessages**: batch message creation task • **updateDeviceShadows**: task for configuring device shadows in batch • **updateDevices**: batch device update task

Parameter	Mandatory	Type	Description
task_mode	No	String	**Parameter description**: batch task mode. Currently, only the gateway mode is supported. This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, if the device to be upgraded is a child device of a gateway, the version information obtaining notification and upgrade notification delivered by the IoT platform carry **task_id** (ID of the software/firmware upgrade batch task) and **sub_device_count** (number of child devices to be upgraded in a batch task). **Value**: **GATEWAY** (gateway mode).
task_ext_info	No	Object	**Parameter description**: extended information of batch tasks This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, this parameter is carried in the version information obtaining notification and upgrade notification delivered by the IoT platform. **Value**: a maximum of 512 characters. Maximum: **512**

Parameter	Mandatory	Type	Description
targets	No	Array of strings	**Parameter description**: batch task targets. Set this parameter to a device ID list, which can include up to 30,000 device IDs. This parameter is supported when **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **deleteDevices**, **freezeDevices**, **unfreezeDevices**, **createCommands**, **createAsyncCommands**, **updateDeviceShadows**, or **createMessages**. If the **targets**, **targets_filter**, and **document_source** parameters are all specified, only one parameter takes effect. The platform uses **targets**, **targets_filter**, and **document_source** in descending order of priority. **Value**: device ID list. The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
targets_filter	No	Map<String,Object>	**Parameter description**: parameters for filtering task targets. The value is in JSON format (key-value pairs). Supported filter key is **group_ids** and its value is a list of group IDs, for example, **["e495cf17-ff79-4294-8f64-4d367919d665"]**. Devices matching the filter will become the task targets. This parameter is supported when **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **deleteDevices**, **freezeDevices**, **unfreezeDevices**, **createCommands**, **createAsyncCommands**, **updateDeviceShadows**, or **createMessages**. If the **targets**, **targets_filter**, and **document_source** parameters are all specified, only one parameter takes effect. The platform uses **targets**, **targets_filter**, and **document_source** in descending order of priority.

Parameter	Mandatory	Type	Description
document	No	Object	**Parameter description**: task execution data file, in JSON format (key-value pairs). This parameter is supported when **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **createCommands**, **createAsyncCommands**, **createMessages**, or **updateDeviceShadows**. • When **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**: Set the key to **package_id** and its value to IDs of the software or firmware packages uploaded to the platform. IDs can be obtained from the software library on the platform. Example: **{"package_id": "32822e5744a45ede319d2c50"}**. • When **task_type** is set to **createCommands**: Enter synchronous command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"}}**. For details, see **Synchronous Device Command APIs**. • When **task_type** is set to **createAsyncCommands**: Set asynchronous command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"},"expire_time":0,"send_strategy":"immediately"}**. For details, see **Asynchronous Device Command APIs**. • When **task_type** is set to **createMessages**: Set

Parameter	Mandatory	Type	Description
			message delivery parameters, for example, **{"message_id":"99b32da9-cd17-4cdf-a286-f6e849cbc364","name":"messageName","message":"HelloWorld","encoding":"none","payload_format":"standard","topic":"messageDown","topic_full_name":"/device/message/down"}**. For details, see **Delivering a Message to a Device**. • When **task_type** is set to **updateDeviceShadows**: Set device shadow parameters, for example, **{"shadow":[{"service_id":"WaterMeter","desired":{"temperature": "60"}}]}**. For details, see **Configuring Desired Properties in a Device Shadow**. Restrictions: 1. The **service_id** and **desired** parameters are mandatory. 2. Up to five service IDs can be configured and each service ID must be unique. 3. The size of the configuration content cannot exceed 10 KB.
task_policy	No	**TaskPolicy** object	**Parameter description**: task policy request details.

Parameter	Mandatory	Type	Description
document_source	No	String	**Parameter description**: ID of the uploaded batch task file. This parameter is supported when **task_type** is set to **createDevices**, **deleteDevices**, **freezeDevices**, or **unfreezeDevices**. Before using this parameter, call the API for uploading a batch task file to obtain the file ID. For details about file templates, see **Template for batch device registration**, **Template for batch device update**, **Template for batch device deletion**, **Template for batch device freezing**, and **Template for batch device unfreezing**. If the **targets**, **targets_filter**, and **document_source** parameters are all specified, only one parameter takes effect. The platform uses **targets**, **targets_filter**, and **document_source** in descending order of priority.

Table 1-596 TaskPolicy

Parameter	Mandatory	Type	Description
schedule_time	No	String	**Parameter description**: time when the task is executed. **Value**: The task is executed within 7 days. If this parameter is not specified, the task is executed immediately. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Mandatory	Type	Description
retry_count	No	Integer	**Parameter description**: number of automatic retries upon failure for subtasks. **Value**: This parameter is mandatory if **retry_interval** is specified. Up to five retries are supported. Minimum: **1** Maximum: **5**
retry_interval	No	Integer	**Parameter description**: interval for automatic retries upon failure for subtasks, in minutes. **Value**: The maximum value is **1440** (24 hours). If this parameter is not passed, subtasks will not be retried. This parameter is mandatory if **retry_count** is specified. Minimum: **0** Maximum: **1440**

Response Parameters

Status code: 201

Table 1-597 Response body parameters

Parameter	Type	Description
task_id	String	Batch task ID. It is allocated by the platform during batch task creation.
task_name	String	Batch task name.

Parameter	Type	Description
task_type	String	Batch task type. Options: • **softwareUpgrade**: software upgrade task • **firmwareUpgrade**: firmware upgrade task • **createDevices**: batch device creation task • **deleteDevices**: batch device deletion task • **freezeDevices**: batch device freezing task • **unfreezeDevices**: batch device unfreezing task • **createCommands**: task for creating synchronous commands in batches • **createAsyncCommands**: task for creating asynchronous commands in batches • **createMessages**: batch message creation task • **updateDeviceShadows**: task for configuring device shadows in batch • **updateDevices**: batch device update task
task_mode	String	**Parameter description**: batch task mode. Currently, only the gateway mode is supported. This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, if the device to be upgraded is a child device of a gateway, the version information obtaining notification and upgrade notification delivered by the IoT platform carry **task_id** (ID of the software/firmware upgrade batch task) and **sub_device_count** (number of child devices to be upgraded in a batch task). **Value**: **GATEWAY** (gateway mode).
task_ext_info	Object	**Parameter description**: extended information of batch tasks This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, this parameter is carried in the version information obtaining notification and upgrade notification delivered by the IoT platform. **Value**: a maximum of 512 characters. Maximum: **512**

Parameter	Type	Description
targets	Array of strings	Batch task targets. When **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **deleteDevices**, **freezeDevices**, **unfreezeDevices**, **createCommands**, **createAsyncCommands**, **createMessages**, or **updateDeviceShadows**, set this parameter to a device ID list.
targets_filter	Map<String,Object>	Parameters for filtering task targets. The value is in JSON format (key-value pairs). Supported filter key is **group_ids** and its value is a list of group IDs, for example, **["e495cf17-ff79-4294-8f64-4d367919d665"]**. Devices matching the filter will become the task targets.
document	Object	Task execution data file in JSON format. When **task_type** is set to **softwareUpgrade** or **firmwareUpgrade**, the JSON file contains the key-value pair. Set **key** to **package_id** and **value** to IDs of the software or firmware packages uploaded on the platform. IDs can be obtained from the software library on the platform. When **task_type** is set to **createCommands**, the JSON file contains command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"}}**. For details, see **Synchronous Device Command APIs**. When **task_type** is set to **createAsyncCommands**, the JSON file contains command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"},"expire_time":0,"send_strategy":"immediately"}**. For details, see **Asynchronous Device Command APIs**. When **task_type** is set to **updateDeviceShadows**, the JSON file contains command parameters, for example, **{"shadow": [{"service_id": "WaterMeter","desired": {"temperature": "60"}}]}**. For details, see **Configuring Desired Properties in a Device Shadow**.
task_policy	**TaskPolicy** object	Policies for executing the task.

Parameter	Type	Description
status	String	Indicates the status of a batch task. This parameter is optional. Options: • **Initializing**: The task is being initialized. • **Waitting**: The task is waiting to be executed. • **Processing**: The task is being executed. • **Success**: The task is executed. • **Fail**: The task execution failed. • **PartialSuccess**: Only some subtasks in the batch task are executed. • **Stopped**: The task is stopped. • **Stopping**: The task is being stopped.
status_desc	String	Description of the batch task status (including the error information of the parent task).
task_progress	**TaskProgress** object	Subtask execution statistics.
create_time	String	Time when the batch task was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-598 TaskPolicy

Parameter	Type	Description
schedule_time	String	**Parameter description**: time when the task is executed. **Value**: The task is executed within 7 days. If this parameter is not specified, the task is executed immediately. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
retry_count	Integer	**Parameter description**: number of automatic retries upon failure for subtasks. **Value**: This parameter is mandatory if **retry_interval** is specified. Up to five retries are supported. Minimum: **1** Maximum: **5**

Parameter	Type	Description
retry_interval	Integer	**Parameter description**: interval for automatic retries upon failure for subtasks, in minutes. **Value**: The maximum value is **1440** (24 hours). If this parameter is not passed, subtasks will not be retried. This parameter is mandatory if **retry_count** is specified. Minimum: **0** Maximum: **1440**

Table 1-599 TaskProgress

Parameter	Type	Description
total	Integer	Total number of subtasks.
processing	Integer	Number of subtasks that are being executed.
success	Integer	Number of subtasks that are executed.
fail	Integer	Number of subtasks that fail.
waitting	Integer	Number of subtasks to execute.
fail_wait_retry	Integer	Number of subtasks waiting for retry.
stopped	Integer	Number of stopped subtasks.
removed	Integer	Indicates the number of removed subtasks.

Example Requests

- Creates a batch software upgrade task for a specified device.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks

{
  "app_id" : "Ev8FVvCfOdQDzrFrxSOemiw_aMca",
  "task_name" : "BatchSoftwareUpgradeTask",
  "task_type" : "softwareUpgrade",
  "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
  "document" : {
    "package_id" : "32822e5744a45ede319d2c50"
  },
  "task_policy" : {
    "schedule_time" : "20151212T121212Z",
    "retry_count" : 5,
    "retry_interval" : 60
  }
}
```

- Creates a batch software upgrade task for a specified device.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks

{
  "app_id" : "Ev8FVvCfOdQDzrFrxSOemiw_aMca",
  "task_name" : "BatchSoftwareUpgradeTask",
```



```
  "task_type" : "softwareUpgrade",
  "document" : {
    "package_id" : "32822e5744a45ede319d2c50"
  },
  "targets_filter" : {
    "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
  },
  "task_policy" : {
    "schedule_time" : "20151212T121212Z",
    "retry_count" : 5,
    "retry_interval" : 60
  }
}
```

- Creates a batch device registration task for a specified device.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks

{
  "app_id" : "Ev8FVvCfOdQDzrFrxSOemiw_aMca",
  "task_name" : "BatchCreateDevicesTask",
  "task_type" : "createDevices",
  "task_policy" : {
    "schedule_time" : "20151212T121212Z",
    "retry_count" : 5,
    "retry_interval" : 60
  },
  "document_source" : "jeQDJQZltU8iKgFFoW060F5SGZka"
}
```

Example Responses

Status code: 201

Created

```
{
  "task_id" : "5c8ba99030344005c02316ad",
  "task_name" : "testname",
  "task_type" : "softwareUpgrade",
  "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
  "targets_filter" : {
    "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
  },
  "document" : {
    "package_id" : "32822e5744a45ede319d2c50"
  },
  "task_policy" : {
    "schedule_time" : "20151212T121212Z",
    "retry_count" : 5,
    "retry_interval" : 60
  },
  "status" : "Success",
  "status_desc" : "string",
  "task_progress" : {
    "total" : 0,
    "processing" : 0,
    "success" : 0,
    "fail" : 0,
    "waitting" : 0,
    "fail_wait_retry" : 0,
    "stopped" : 0
  },
  "create_time" : "20151212T121212Z"
}
```


Status Codes

Status Code	Description
201	Created
400	BAD REQUEST
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.2 Query the Batch Task List

Function

This API is used by an application to query the batch task list on the IoT platform. Each task includes the task content, task status, and task completion statistics.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/batchtasks

Table 1-600 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-601 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can specify this parameter to query tasks in a specified resource space. If this parameter is not carried, tasks in all resource spaces are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
task_type	Yes	String	**Parameter description**: batch task type. **Options**: • **softwareUpgrade**: software upgrade task • **firmwareUpgrade**: firmware upgrade task • **createDevices**: batch device creation task • **deleteDevices**: batch device deletion task • **freezeDevices**: batch device freezing task • **unfreezeDevices**: batch device unfreezing task • **createCommands**: task for creating synchronous commands in batches • **createAsyncCommands**: task for creating asynchronous commands in batches • **createMessages**: batch message creation task • **updateDeviceShadows**: task for configuring device shadows in batch • **updateDevices**: batch device update task

Parameter	Mandatory	Type	Description
status	No	String	**Parameter description**: Indicates the status of the batch task. This parameter is optional. Options: • **Initializing**: The task is being initialized. • **Waitting**: The task is waiting to be executed. • **Processing**: The task is being executed. • **Success**: The task is executed. • **Fail**: The task execution failed. • **PartialSuccess**: Only some subtasks in the batch task are executed. • **Stopped**: The task is stopped. • **Stopping**: The task is being stopped.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-602 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Response Parameters

Status code: 200

Table 1-603 Response body parameters

Parameter	Type	Description
batchtasks	Array of **Task** objects	Batch task list.
page	**Page** object	Pagination information of the query results.

Table 1-604 Task

Parameter	Type	Description
task_id	String	Batch task ID. It is allocated by the platform during batch task creation.
task_name	String	Batch task name.
task_type	String	Batch task type. Options: • **softwareUpgrade**: software upgrade task • **firmwareUpgrade**: firmware upgrade task • **createDevices**: batch device creation task • **deleteDevices**: batch device deletion task • **freezeDevices**: batch device freezing task • **unfreezeDevices**: batch device unfreezing task • **createCommands**: task for creating synchronous commands in batches • **createAsyncCommands**: task for creating asynchronous commands in batches • **createMessages**: batch message creation task • **updateDeviceShadows**: task for configuring device shadows in batch • **updateDevices**: batch device update task
task_mode	String	**Parameter description**: batch task mode. Currently, only the gateway mode is supported. This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, if the device to be upgraded is a child device of a gateway, the version information obtaining notification and upgrade notification delivered by the IoT platform carry **task_id** (ID of the software/firmware upgrade batch task) and **sub_device_count** (number of child devices to be upgraded in a batch task). **Value**: **GATEWAY** (gateway mode).
task_ext_info	Object	**Parameter description**: extended information of batch tasks This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, this parameter is carried in the version information obtaining notification and upgrade notification delivered by the IoT platform. **Value**: a maximum of 512 characters. Maximum: **512**

Parameter	Type	Description
targets	Array of strings	Batch task targets. When **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **deleteDevices**, **freezeDevices**, **unfreezeDevices**, **createCommands**, **createAsyncCommands**, **createMessages**, or **updateDeviceShadows**, set this parameter to a device ID list.
targets_filter	Map<String,Object>	Parameters for filtering task targets. The value is in JSON format (key-value pairs). Supported filter key is **group_ids** and its value is a list of group IDs, for example, **["e495cf17-ff79-4294-8f64-4d367919d665"]**. Devices matching the filter will become the task targets.
document	Object	Task execution data file in JSON format. When **task_type** is set to **softwareUpgrade** or **firmwareUpgrade**, the JSON file contains the key-value pair. Set **key** to **package_id** and **value** to IDs of the software or firmware packages uploaded on the platform. IDs can be obtained from the software library on the platform. When **task_type** is set to **createCommands**, the JSON file contains command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"}}**. For details, see **Synchronous Device Command APIs**. When **task_type** is set to **createAsyncCommands**, the JSON file contains command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"},"expire_time":0,"send_strategy":"immediately"}**. For details, see **Asynchronous Device Command APIs**. When **task_type** is set to **updateDeviceShadows**, the JSON file contains command parameters, for example, **{"shadow": [{"service_id": "WaterMeter","desired": {"temperature": "60"}}]}**. For details, see **Configuring Desired Properties in a Device Shadow**.
task_policy	**TaskPolicy** object	Policies for executing the task.

Parameter	Type	Description
status	String	Indicates the status of a batch task. This parameter is optional. Options: • **Initializing**: The task is being initialized. • **Waitting**: The task is waiting to be executed. • **Processing**: The task is being executed. • **Success**: The task is executed. • **Fail**: The task execution failed. • **PartialSuccess**: Only some subtasks in the batch task are executed. • **Stopped**: The task is stopped. • **Stopping**: The task is being stopped.
status_desc	String	Description of the batch task status (including the error information of the parent task).
task_progress	**TaskProgress** object	Subtask execution statistics.
create_time	String	Time when the batch task was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-605 TaskPolicy

Parameter	Type	Description
schedule_time	String	**Parameter description**: time when the task is executed. **Value**: The task is executed within 7 days. If this parameter is not specified, the task is executed immediately. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
retry_count	Integer	**Parameter description**: number of automatic retries upon failure for subtasks. **Value**: This parameter is mandatory if **retry_interval** is specified. Up to five retries are supported. Minimum: **1** Maximum: **5**

Parameter	Type	Description
retry_interval	Integer	**Parameter description**: interval for automatic retries upon failure for subtasks, in minutes. **Value**: The maximum value is **1440** (24 hours). If this parameter is not passed, subtasks will not be retried. This parameter is mandatory if **retry_count** is specified. Minimum: **0** Maximum: **1440**

Table 1-606 TaskProgress

Parameter	Type	Description
total	Integer	Total number of subtasks.
processing	Integer	Number of subtasks that are being executed.
success	Integer	Number of subtasks that are executed.
fail	Integer	Number of subtasks that fail.
waitting	Integer	Number of subtasks to execute.
fail_wait_retry	Integer	Number of subtasks waiting for retry.
stopped	Integer	Number of stopped subtasks.
removed	Integer	Indicates the number of removed subtasks.

Table 1-607 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries all batch tasks in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/batchtasks
```

Example Responses

Status code: 200

OK


```
{
  "batchtasks" : [ {
    "task_id" : "5c8ba99030344005c02316ad",
    "task_name" : "testname",
    "task_type" : "softwareUpgrade",
    "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
    "targets_filter" : {
      "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
    },
    "document" : {
      "package_id" : "32822e5744a45ede319d2c50"
    },
    "task_policy" : {
      "schedule_time" : "20151212T121212Z",
      "retry_count" : 5,
      "retry_interval" : 60
    },
    "status" : "Success",
    "status_desc" : "string",
    "task_progress" : {
      "total" : 0,
      "processing" : 0,
      "success" : 0,
      "fail" : 0,
      "waitting" : 0,
      "fail_wait_retry" : 0,
      "stopped" : 0
    },
    "create_time" : "20151212T121212Z"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.3 Query a Batch Task

Function

This API is used by an application to query a specific batch task on the platform, including the task content, task status, task completion statistics, and subtask list.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/batchtasks/{task_id}

Table 1-608 Path Parameters

Parameter	Mandatory	Type	Description
task_id	Yes	String	**Parameter description**: batch task ID. It is allocated by the platform during batch task creation. **Value**: The value can contain a maximum of 24 characters. Only lowercase letters a to f and digits are allowed.
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-609 Query Parameters

Parameter	Mandatory	Type	Description
task_detail_status	No	String	**Parameter description**: Indicates the execution status of a subtask. This parameter is optional. Options: • **Success**: The task is executed. • **Fail**: The task execution failed. • **Processing**: The task is being executed. • **FailWaitRetry**: The task is waiting for retry after execution failure. • **Stopped**: The task is stopped. • **Waitting**: The task is waiting to be executed. • **Removed**: The task is removed.
target	No	String	**Parameter description**: Indicates the target of a batch operation. When **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **deleteDevices**, **freezeDevices**, **unfreezeDevices**, **createCommands**, **createAsyncCommands**, **createMessages**, or **updateDeviceShadows**, set this parameter to **device_id**. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-610 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Response Parameters

Status code: 200

Table 1-611 Response body parameters

Parameter	Type	Description
batchtask	**Task** object	Basic information about the batch task.
task_details	Array of **TaskDetail** objects	Subtask list.
page	**Page** object	Pagination information of the query results.

Table 1-612 Task

Parameter	Type	Description
task_id	String	Batch task ID. It is allocated by the platform during batch task creation.
task_name	String	Batch task name.
task_type	String	Batch task type. Options: • **softwareUpgrade**: software upgrade task • **firmwareUpgrade**: firmware upgrade task • **createDevices**: batch device creation task • **deleteDevices**: batch device deletion task • **freezeDevices**: batch device freezing task • **unfreezeDevices**: batch device unfreezing task • **createCommands**: task for creating synchronous commands in batches • **createAsyncCommands**: task for creating asynchronous commands in batches • **createMessages**: batch message creation task • **updateDeviceShadows**: task for configuring device shadows in batch • **updateDevices**: batch device update task
task_mode	String	**Parameter description**: batch task mode. Currently, only the gateway mode is supported. This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, if the device to be upgraded is a child device of a gateway, the version information obtaining notification and upgrade notification delivered by the IoT platform carry **task_id** (ID of the software/firmware upgrade batch task) and **sub_device_count** (number of child devices to be upgraded in a batch task). **Value**: **GATEWAY** (gateway mode).
task_ext_info	Object	**Parameter description**: extended information of batch tasks This parameter is supported when **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**. In the software/firmware upgrade scenario, this parameter is carried in the version information obtaining notification and upgrade notification delivered by the IoT platform. **Value**: a maximum of 512 characters. Maximum: **512**

Parameter	Type	Description
targets	Array of strings	Batch task targets. When **task_type** is set to **firmwareUpgrade**, **softwareUpgrade**, **deleteDevices**, **freezeDevices**, **unfreezeDevices**, **createCommands**, **createAsyncCommands**, **createMessages**, or **updateDeviceShadows**, set this parameter to a device ID list.
targets_filter	Map<String,Object>	Parameters for filtering task targets. The value is in JSON format (key-value pairs). Supported filter key is **group_ids** and its value is a list of group IDs, for example, **["e495cf17-ff79-4294-8f64-4d367919d665"]**. Devices matching the filter will become the task targets.
document	Object	Task execution data file in JSON format. When **task_type** is set to **softwareUpgrade** or **firmwareUpgrade**, the JSON file contains the key-value pair. Set **key** to **package_id** and **value** to IDs of the software or firmware packages uploaded on the platform. IDs can be obtained from the software library on the platform. When **task_type** is set to **createCommands**, the JSON file contains command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"}}**. For details, see **Synchronous Device Command APIs**. When **task_type** is set to **createAsyncCommands**, the JSON file contains command parameters, for example, **{"service_id":"water","command_name":"ON_OFF","paras":{"value":"ON"},"expire_time":0,"send_strategy":"immediately"}**. For details, see **Asynchronous Device Command APIs**. When **task_type** is set to **updateDeviceShadows**, the JSON file contains command parameters, for example, **{"shadow": [{"service_id": "WaterMeter","desired": {"temperature": "60"}}]}**. For details, see **Configuring Desired Properties in a Device Shadow**.
task_policy	**TaskPolicy** object	Policies for executing the task.

Parameter	Type	Description
status	String	Indicates the status of a batch task. This parameter is optional. Options: • **Initializing**: The task is being initialized. • **Waitting**: The task is waiting to be executed. • **Processing**: The task is being executed. • **Success**: The task is executed. • **Fail**: The task execution failed. • **PartialSuccess**: Only some subtasks in the batch task are executed. • **Stopped**: The task is stopped. • **Stopping**: The task is being stopped.
status_desc	String	Description of the batch task status (including the error information of the parent task).
task_progress	**TaskProgress** object	Subtask execution statistics.
create_time	String	Time when the batch task was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-613 TaskPolicy

Parameter	Type	Description
schedule_time	String	**Parameter description**: time when the task is executed. **Value**: The task is executed within 7 days. If this parameter is not specified, the task is executed immediately. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
retry_count	Integer	**Parameter description**: number of automatic retries upon failure for subtasks. **Value**: This parameter is mandatory if **retry_interval** is specified. Up to five retries are supported. Minimum: **1** Maximum: **5**

Parameter	Type	Description
retry_interval	Integer	**Parameter description**: interval for automatic retries upon failure for subtasks, in minutes. **Value**: The maximum value is **1440** (24 hours). If this parameter is not passed, subtasks will not be retried. This parameter is mandatory if **retry_count** is specified. Minimum: **0** Maximum: **1440**

Table 1-614 TaskProgress

Parameter	Type	Description
total	Integer	Total number of subtasks.
processing	Integer	Number of subtasks that are being executed.
success	Integer	Number of subtasks that are executed.
fail	Integer	Number of subtasks that fail.
waitting	Integer	Number of subtasks to execute.
fail_wait_retry	Integer	Number of subtasks waiting for retry.
stopped	Integer	Number of stopped subtasks.
removed	Integer	Indicates the number of removed subtasks.

Table 1-615 TaskDetail

Parameter	Type	Description
target	String	Batch task targets.
status	String	Execution status of the subtask. Options: • **Waitting**: The task is waiting to be executed. • **Processing**: The task is being executed. • **Success**: The task is executed. • **Fail**: The task fails to be executed. • **FailWaitRetry**: The task is waiting to be retried after execution failure. • **Stopped**: The task is stopped.
output	String	Output information about subtask execution.

Parameter	Type	Description
error	**ErrorInfo** object	Information about the subtask execution failure.

Table 1-616 ErrorInfo

Parameter	Type	Description
error_code	String	Error Code
error_msg	String	Error description.

Table 1-617 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries the details of a specified batch task.

```
GET https://{endpoint}/v5/iot/{project_id}/batchtasks/{taskId}
```

Example Responses

Status code: 200

OK

```
{
  "batchtask" : {
    "task_id" : "5c8ba99030344005c02316ad",
    "task_name" : "testname",
    "task_type" : "softwareUpgrade",
    "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
    "targets_filter" : {
      "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
    },
    "document" : {
      "package_id" : "32822e5744a45ede319d2c50"
    },
    "task_policy" : {
      "schedule_time" : "20151212T121212Z",
      "retry_count" : 5,
      "retry_interval" : 60
    },
    "status" : "Success",
    "status_desc" : "string",
    "task_progress" : {
```



```
    "total" : 0,
    "processing" : 0,
    "success" : 0,
    "fail" : 0,
    "waitting" : 0,
    "fail_wait_retry" : 0,
    "stopped" : 0
  },
  "create_time" : "20151212T121212Z"
 },
 "task_details" : [ {
  "target" : "e495cf17-ff79-4294-8f64-4d367919d665",
  "status" : "Success",
  "output" : "success",
  "error" : {
    "error_code" : "IOTDA.000002",
    "error_msg" : "The request is unauthorized."
  }
 } ],
 "page" : {
  "count" : 10,
  "marker" : "5c90fa7d3c4e4405e8525079"
 }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.4 Delete a Batch Task

Function

This API is used by an application to delete a completed batch task (successful, failed, partially successful, or stopped) from the platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/batchtasks/{task_id}

Table 1-618 Path Parameters

Parameter	Mandatory	Type	Description
task_id	Yes	String	**Parameter description**: Indicates the batch task ID. It is allocated by the platform during batch task creation. **Value**: The value can contain a maximum of 24 characters. Only lowercase letters **a** to **f** and digits are allowed.
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-619 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Response Parameters

None

Example Requests

Deletes a batch task

```
DELETE https://{endpoint}/v5/iot/{project_id}/batchtasks/{taskId}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.5 Re-Execute a Batch Task

Function

This API is used by an application server to retry a batch task. Currently, **task_type** can only be set to **firmwareUpgrade** or **softwareUpgrade**. If the task specified by **task_id** is successful, stopped, being stopped, waiting, or being initialized, this API cannot be called. If the request body is **{}**, all subtasks in the **Fail**, **FailWaitRetry**, and **Stopped** statuses will be re-executed after this API is called.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/batchtasks/{task_id}/retry

Table 1-620 Path Parameters

Parameter	Mandatory	Type	Description
task_id	Yes	String	**Parameter description**: Indicates the batch task ID. It is allocated by the platform during batch task creation. **Value**: The value can contain a maximum of 24 characters. Only lowercase letters **a** to **f** and digits are allowed.
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-621 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Table 1-622 Request body parameters

Parameter	Mandatory	Type	Description
targets	No	Array of strings	Indicates the target set of the batch operation. Max: 100 targets. When **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**, set this parameter to **device_id**.

Response Parameters

Status code: 200

Table 1-623 Response body parameters

Parameter	Type	Description
targets	Array of **BatchTargetResult** objects	Indicates the batch operation result set.

Table 1-624 BatchTargetResult

Parameter	Type	Description
target	String	Indicates the batch operation targets.
status	String	Indicates the execution result. The value can be **success** or **failure**.
error_code	String	Indicates the error code.
error_msg	String	Indicates the error description.

Example Requests

- Re-execute subtasks in the **Fail**, **FailWaitRetry**, and **Stopped** statuses.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks/{task_id}/retry

{ }
```

- Re-executes a subtask of a specified target set.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks/{task_id}/retry

{
  "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
}
```

Example Responses

Status code: 200

OK

```
{
  "targets" : [ {
    "target" : "e495cf17-ff79-4294-8f64-4d367919d665",
    "status" : "failure",
    "error_code" : "IOTDA.014219",
    "error_msg" : "Invalid input. The target is not in the task"
  }, {
    "target" : "e495cf17-ff79-4294-8f64-4d367919d677",
    "status" : "success"
  } ]
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.6 Stop a Batch Task

Function

This API is used by an application server to stop a batch task. Currently, **task_type** can only be set to **firmwareUpgrade** or **softwareUpgrade**. If the task specified by **task_id** has been completed (successful, failed, partially successful, or stopped) or is being stopped, this API cannot be called. If the request body is **{}**, all subtasks in the **Processing**, **Waitting**, and **FailWaitRetry** statuses will be stopped after this API is called.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/batchtasks/{task_id}/stop

Table 1-625 Path Parameters

Parameter	Mandatory	Type	Description
task_id	Yes	String	**Parameter description**: Indicates the batch task ID. It is allocated by the platform during batch task creation. **Value**: The value can contain a maximum of 24 characters. Only lowercase letters **a** to **f** and digits are allowed.
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-626 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Table 1-627 Request body parameters

Parameter	Mandatory	Type	Description
targets	No	Array of strings	Indicates the target set of the batch operation. Max: 100 targets. When **task_type** is set to **firmwareUpgrade** or **softwareUpgrade**, set this parameter to **device_id**.

Response Parameters

Status code: 200

Table 1-628 Response body parameters

Parameter	Type	Description
targets	Array of **BatchTargetResult** objects	Indicates the batch operation result set.

Table 1-629 BatchTargetResult

Parameter	Type	Description
target	String	Indicates the batch operation targets.
status	String	Indicates the execution result. The value can be **success** or **failure**.
error_code	String	Indicates the error code.
error_msg	String	Indicates the error description.

Example Requests

- Stops all subtasks that are in the **Processing**, **Waitting**, and **FailWaitRetry** statuses.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks/{task_id}/stop

{ }
```

- Stops executing subtasks of a specified target set.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtasks/{task_id}/stop

{
  "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
}
```


Example Responses

Status code: 200

OK

```
{
  "targets" : [ {
    "target" : "e495cf17-ff79-4294-8f64-4d367919d665",
    "status" : "failure",
    "error_code" : "IOTDA.014219",
    "error_msg" : "Invalid input. The target is not in the task"
  }, {
    "target" : "e495cf17-ff79-4294-8f64-4d367919d677",
    "status" : "success"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.7 Batch Task File Management

You can call the APIs to manage task files in batches.

- You can manage files of batch device creation, deletion, freezing, and unfreezing tasks.

- You can manage a maximum of 10 files. If the number of files exceeds 10, new files cannot be uploaded.

- Files can be stored for a maximum duration of one hour. If the storage duration elapses, the files will be automatically aged by the platform.

1.4.16.7.1 Upload a Batch Task File

Function

This API is used by an application to upload a batch task file for task creation. You can upload files of batch device creation, deletion, freezing, and unfreezing tasks.

- **Template for batch device registration**
- **Template for batch device deletion**
- **Template for batch device freezing**
- **Template for batch device unfreezing**

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/batchtask-files

Table 1-630 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-631 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Table 1-632 FormData parameters

Parameter	Mandatory	Type	Description
file	Yes	File	**Parameter description**: file to be uploaded. **Value**: Currently, only XLSX and XLS files are supported, and a file can contain up to 100,000 lines.

Response Parameters

Status code: 201

Table 1-633 Response body parameters

Parameter	Type	Description
file_id	String	ID of a batch task file uploaded to the platform. The file ID is automatically generated by the platform.
file_name	String	Name of the uploaded batch task file. Minimum: **1** Maximum: **60**
upload_time	String	Time when the file was uploaded to the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Uploads a batch task file.

```
POST https://{endpoint}/v5/iot/{project_id}/batchtask-files
```

Example Responses

Status code: 201

Created

```
{
  "file_id" : "0c3c77dd-42a2-4309-9e10-da2e8bf64ac3",
  "file_name" : "BatchCreateDevices_test01.xlsx",
  "upload_time" : "20200617T081608Z"
}
```

Status Codes

Status Code	Description
201	Created
400	BAD REQUEST
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.7.2 Query the List of Batch Task Files

Function

This API is used by an application to query the batch task file list.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/batchtask-files

Table 1-634 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-635 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Response Parameters

Status code: 200

Table 1-636 Response body parameters

Parameter	Type	Description
files	Array of **BatchTaskFile** objects	Batch task file list. Array Length: **0 - 10**

Table 1-637 BatchTaskFile

Parameter	Type	Description
file_id	String	ID of a batch task file uploaded to the platform. The file ID is automatically generated by the platform.
file_name	String	Name of the uploaded batch task file. Minimum: **1** Maximum: **60**
upload_time	String	Time when the file was uploaded to the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Queries batch task files in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/batchtask-files
```

Example Responses

Status code: 200

OK

```
{
  "files" : [ {
    "file_id" : "0c3c77dd-42a2-4309-9e10-da2e8bf64ac3",
    "file_name" : "BatchCreateDevices_test01.xlsx",
    "upload_time" : "20200617T081608Z"
  }, {
    "file_id" : "9c338eba-b162-4005-98ea-ff34a13c70da",
    "file_name" : "BatchCreateDevices_test02.xlsx",
    "upload_time" : "20200617T081620Z"
  } ]
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.16.7.3 Delete a Batch Task File

Function

This API is used by an application to delete a batch task file.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/batchtask-files/{file_id}

Table 1-638 Path Parameters

Parameter	Mandatory	Type	Description
file_id	Yes	String	**Parameter description**: ID of the batch task file to be deleted. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Minimum: **1** Maximum: **128**
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-639 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**. Minimum: **1** Maximum: **36**

Response Parameters

None

Example Requests

Deletes a batch task file.

```
DELETE https://{endpoint}/v5/iot/{project_id}/batchtask-files/{file_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.17 Device CA Certificate Management

1.4.17.1 Upload a Device CA Certificate

Function

This API is used by an application to upload a device CA certificate to the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/certificates

Table 1-640 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-641 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	User token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	Instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-642 Request body parameters

Parameter	Mandatory	Type	Description
content	Yes	String	Certificate content. Minimum: **1** Maximum: **65535**
app_id	No	String	Resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the certificate to create will belong. If this parameter is not specified, the certificate to create will belong to the **default resource space**.

Response Parameters

Status code: 201

Table 1-643 Response body parameters

Parameter	Type	Description
certificate_id	String	Unique CA certificate ID, allocated by the platform when the certificate is uploaded.
cn_name	String	CN of the CA certificate.
owner	String	Owner of the CA certificate.
status	Boolean	Verification status of the CA certificate. **true** indicates that the certificate has been verified and can be used for device access authentication. **false** indicates that the certificate does not pass the verification.
verify_code	String	Verification code of the CA certificate.
provision_enable	Boolean	Whether to enable the self-registration capability. The options are **true** (yes) and **false** (no). If this parameter is set to **true**, this function must be used together with the pre-provisioning function.
template_id	String	ID of the bound pre-provisioning template.
create_date	String	Time when the certificate was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
effective_date	String	Time when the CA certificate starts to take effect. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
expiry_date	String	Time when CA certificate expires. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Uploads a device CA certificate.

```
POST https://{endpoint}/v5/iot/{project_id}/certificates

{
  "content" : "-----BEGINCERTIFICATE-----
\nMIID2TCCAsGgAwIBAgIJAOEDEgVdVMn9MA0GCSqGSIb3DQEBCwUAMIGCMQswCQYD
\nVQQGEwJDTjERMA8GA1UECAwIR3VhbmRvbmcxETAPBgNVBAcMCFNoZW56aGVuMQ8w
\nDQYDVQQKDAZIdWF3ZWkxDDAKBgNVBAsMA2lvdDESMBAGA1UEAwwJMTIzNDU2Nzg5\nMRowGAYJKoZ
IhvcNAQkBFgtkamthQHFxLmNvbTAeFw0xOTEyMTkxMzE1MjZaFw0y
```

\nMjEwMDgxMzE1MjZaMIGCMQswCQYDVQQGEwJDTjERMA8GA1UECAwIR3VhbmRvbmcx
\nETAPBgNVBAcMCFNoZW56aGVuMQ8wDQYDVQQKDAZIdWF3ZWkxDDAKBgNVBAsMA2lv
\ndDESMBAGA1UEAwwJMTIzNDU2Nzg5MRowGAYJKoZIhvcNAQkBFgtkamthQHFxLmNv
\nbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM72QUzoadvLfxGjt3UF
\noZ4MJbblqnRbouO4KpOVHBXyS2yQVl4CWWMhLh4pp2efNUSqKuXHjY3r68PquyNn
\nYk8zO59zVc7JHvjGkBvo7DgPRAhEKPLJIpRzkmlCBbxwTNCjc3FovGb/sHHNlpGn
\ncCKUzMfPGNZuBiuemskuEXL/eMHxDPbXYWn4Wq0wt+28PKUL5jybY7nsXSNnmAPF
\nTO0CAmq0meUukubT/jHDCQ78ihQ/iqw1RNq88aCqRleoHiGg5nWkjL+05GXqUrqV\nVnZNL
+YqcXzuVMs5XgyhNM2AsuH2g3D8ZuF6Dj9qY1n/v/Cp/DGpxP3A74SlplnF\nD/
0CAwEAAaNQME4wHQYDVR0OBBYEFAVPWVtpTdO6KQnmVrrNlMguWNR7MB8GA1Ud
\nIwQYMBaAFAVPWVtpTdO6KQnmVrrNlMguWNR7MAwGA1UdEwQFMAMBAf8wDQYJKoZI
\nhvcNAQELBQADggEBAE40ViqK+UaEn++Xq6f4Cmeg3JqYHu47v9RIAASNihYRBQ/r
\n3RE7Af3GqjIO5nMJJuCMzdcoAU8N9KwkgXD+GLR9fYLEoEmq5CrhgaGDsCi85vCs
\nmWhj5z8r5TG207xpmvH2KT447dnG+chMBE594ma85dCv+0mCDrqNToElipgT8+rY
\nAYVClnIt3kbsTg1vSRNHadd+TpgRVxJZBF0fHcCAyc/2f3UJgPYNWShIetHM6BdI\n3fZ4H
+eeHPjagm5kzmffli1cUv2/N+1hKUvcI4uFCqEwZRFtp90RyIbxUfQwi+Cs\nXVnwV
+BZS5qD9bTcfxZMXhuVRwO/5xWYMYPN1uY=\n-----END CERTIFICATE-----",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
}

Example Responses

Status code: 201

Created

{
  "certificate_id" : "string",
  "cn_name" : "string",
  "owner" : "string",
  "status" : true,
  "verify_code" : "string",
  "create_date" : "20191212T121212Z",
  "effective_date" : "20191212T121212Z",
  "expiry_date" : "20221212T121212Z",
  "provision_enable" : true,
  "template_id" : "61c970ce2d63eb6ee655dbf0"
}

Status Codes

Status Code	Description
201	Created
401	Unauthorized
403	Forbidden
400	Bad Request
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.17.2 Obtain the Device CA Certificate List

Function

This API is used by an application to obtain the device CA certificate list from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/certificates

Table 1-644 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-645 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	Resource space ID. This parameter is optional. If you have multiple resource spaces, you can specify this parameter to query certificates in the specified resource space. If this parameter is not carried, certificates in all resource spaces are queried.
limit	No	Integer	Number of records to be displayed on each page. The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. Default: **ffffffffffffffffffffffffffffffff**
offset	No	Integer	If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-646 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	User token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	Instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-647 Response body parameters

Parameter	Type	Description
certificates	Array of **CertificatesRspDTO** objects	Certificate list.
page	**Page** object	Pagination information of the query results.

Table 1-648 CertificatesRspDTO

Parameter	Type	Description
certificate_id	String	Unique CA certificate ID, allocated by the platform when the certificate is uploaded.
cn_name	String	CN of the CA certificate.
owner	String	Owner of the CA certificate.
status	Boolean	Verification status of the CA certificate. **true** indicates that the certificate has been verified and can be used for device access authentication. **false** indicates that the certificate does not pass the verification.
verify_code	String	Verification code of the CA certificate.
provision_enable	Boolean	Whether to enable the self-registration capability. The options are **true** (yes) and **false** (no). If this parameter is set to **true**, this function must be used together with the pre-provisioning function.
template_id	String	ID of the bound pre-provisioning template.
create_date	String	Time when the certificate was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
effective_date	String	Time when the CA certificate starts to take effect. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
expiry_date	String	Time when CA certificate expires. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-649 Page

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Obtains the device CA certificate list.


```
GET https://{endpoint}/v5/iot/{project_id}/certificates
```

Example Responses

Status code: 200

OK

```
{
  "certificates" : [ {
    "certificate_id" : "string",
    "cn_name" : "string",
    "owner" : "string",
    "status" : true,
    "verify_code" : "string",
    "create_date" : "20191212T121212Z",
    "effective_date" : "20191212T121212Z",
    "expiry_date" : "20221212T121212Z",
    "provision_enable" : true,
    "template_id" : "61c970ce2d63eb6ee655dbf0"
  } ],
  "page" : {
    "count" : 100,
    "marker" : "5c8f3d2d3df1f10d803adbda"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.17.3 Delete a Device CA Certificate

Function

This API is used by an application to delete a device CA certificate from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/certificates/{certificate_id}

Table 1-650 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
certificate_id	Yes	String	Unique CA certificate ID, allocated by the platform when the certificate is uploaded. Minimum: **1** Maximum: **36**

Request Parameters

Table 1-651 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	User token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	Instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a device CA certificate.

```
DELETE https://{endpoint}/v5/iot/{project_id}/certificates/{certificate_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.17.4 Update a CA Certificate

Function

This API is used by an application to update a CA certificate on the IoT platform. This API is supported only by standard and enterprise editions.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/certificates/{certificate_id}

Table 1-652 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
certificate_id	Yes	String	Unique CA certificate ID, allocated by the platform when the certificate is uploaded. Minimum: **1** Maximum: **36**

Request Parameters

Table 1-653 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	User token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	Instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-654 Request body parameters

Parameter	Mandatory	Type	Description
provision_enable	No	Boolean	Whether to enable the self-registration capability. The options are **true** (yes) and **false** (no). If this parameter is set to **true**, this function must be used together with the pre-provisioning function.
template_id	No	String	ID of the pre-provisioning template bound to the CA certificate. If this parameter is set to **null**, the binding relationship is canceled.

Response Parameters

Status code: 200

Table 1-655 Response body parameters

Parameter	Type	Description
certificate_id	String	Unique CA certificate ID, allocated by the platform when the certificate is uploaded.
cn_name	String	CN of the CA certificate.
owner	String	Owner of the CA certificate.
status	Boolean	Verification status of the CA certificate. **true** indicates that the certificate has been verified and can be used for device access authentication. **false** indicates that the certificate does not pass the verification.
verify_code	String	Verification code of the CA certificate.
provision_enable	Boolean	Whether to enable the self-registration capability. The options are **true** (yes) and **false** (no). If this parameter is set to **true**, this function must be used together with the pre-provisioning function.
template_id	String	ID of the bound pre-provisioning template.
create_date	String	Time when the certificate was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Type	Description
effective_date	String	Time when the CA certificate starts to take effect. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
expiry_date	String	Time when CA certificate expires. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Associates the certificate with the self-registration template and enables the self-registration function.

```
PUT https://{endpoint}/v5/iot/{project_id}/certificates/{certificate_id}

{
  "template_id" : "61c970ce2d63eb6ee655dbf0",
  "provision_enable" : true
}
```

Example Responses

Status code: 200

Successful response

```
{
  "certificate_id" : "string",
  "cn_name" : "string",
  "owner" : "string",
  "status" : true,
  "verify_code" : "string",
  "provision_enable" : true,
  "template_id" : "61c970ce2d63eb6ee655dbf0",
  "create_date" : "20191212T121212Z",
  "effective_date" : "20191212T121212Z",
  "expiry_date" : "20221212T121212Z"
}
```

Status Codes

Status Code	Description
200	Successful response
400	Bad Request
401	Unauthorized
404	Not Found
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.17.5 Verify a Device CA Certificate

Function

This API is used by an application to verify a device CA certificate on the IoT platform to check whether the user has the private key of the CA certificate.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/certificates/{certificate_id}/action

Table 1-656 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
certificate_id	Yes	String	Unique CA certificate ID, allocated by the platform when the certificate is uploaded. Minimum: **1** Maximum: **36**

Table 1-657 Query Parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Operation performed on the certificate. Currently, only **verify** (certificate verification) is supported.

Request Parameters

Table 1-658 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	User token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. **X-Subject-Token** in the response header returned by the API is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	Instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-659 Request body parameters

Parameter	Mandatory	Type	Description
verify_content	Yes	String	Certificate content to verify. Minimum: **1** Maximum: **65535**

Response Parameters

None

Example Requests

Verifies a device CA certificate.

```
POST https://{endpoint}/v5/iot/{project_id}/certificates/{certificate_id}/action

{
  "verify_content" : "-----BEGIN CERTIFICATE-----
\nMIIDnzCCAocCCQCs5+qyyItl5TANBgkqhkiG9w0BAQsFADCBgjELMAkGA1UEBhMC
```

\nQ04xETAPBgNVBAgMCEd1YW5kb25nMREwDwYDVQQHDAhTaGVuemhlbjEPMA0GA1UE
\nCgwGSHVhd2VpMQwwCgYDVQQLDANpb3QxEjAQBgNVBAMMCTEyMzQ1Njc4OTEaMBgG
\nCSqGSIb3DQEJARYLZGprYUBxcS5jb20wHhcNMTkxMjE5MTMyMTM3WhcNMjEwNTAy
\nMTMyMTM3WjCBnzELMAkGA1UEBhMCQ04xEjAQBgNVBAgMCUd1YW5nZG9uZzERMA8G
\nA1UEBwwIU2hlbnpoZW4xDzANBgNVBAoMBm9yaWdpbjENMAsGA1UECwwEdW5pdDEt
\nMCsGA1UEAwwkMmM4YjU5MDUtYjM0YS00YjY0LTgxMTItZjZjMDQ3YWUwNjVjMRow
\nGAYJKoZIhvcNAQkBFgtqbGtqQHFxLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEP
\nADCCAQoCggEBAM72QUzoadvLfxGjt3UFoZ4MJbblqnRbouO4KpOVHBXyS2yQVl4C
\nWWMhLh4pp2efNUSqKuXHjY3r68PquyNnYk8zO59zVc7JHvjGkBvo7DgPRAhEKPLJ
\nIpRzkmlCBbxwTNCjc3FovGb/sHHNlpGncCKUzMfPGNZuBiuemskuEXL/eMHxDPbX\nYWn4Wq0wt
+28PKUL5jybY7nsXSNnmAPFTO0CAmq0meUukubT/jHDCQ78ihQ/iqw1\nRNq88aCqRleoHiGg5nWkjL
+05GXqUrqVVnZNL+YqcXzuVMs5XgyhNM2AsuH2g3D8\nZuF6Dj9qY1n/v/Cp/DGpxP3A74SlplnFD/
0CAwEAATANBgkqhkiG9w0BAQsFAAOC\nAQEAh1SF1Z/
p8nT7k8868lLNBZrIcErMlkFdghn2HRYyw5iilDXL28lJEBax2X1M\nNl2fD/rov9gwxhyrBZD2YkevL8k
+DXcVpVEoozwpUR3p79YEyT0E3jI67G/EiB2h\n+o7+deDlH7d7Li/ZOSQC6JTSLshBhi
+B8CQmYYt6YCjN7Rswbf1Z8bsQNrcsxW36\nZM3uG3i9GrEktypTNXMRUbG5gngaFKbRGGUPWNYdNXQeXU
W9cpj8HAyndESEwAYz\ntLKHdnM874P8ZAmRkijZoToOCMcT0s8l8SoYUR7iWI0E08KYzAPgLX9Xvw42GCEF
\nb2TJfnOIwhu8gFf7cwlCGC+gRA==\n-----END CERTIFICATE-----"
}

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.18 OTA Upgrade Package Management

1.4.18.1 Create an OTA Upgrade Package

Function

This API is used to create an OBS object associated with an upgrade package. To use this API, grant IoTDA instances the permission to access OBS and the KMS Administrator permissions. On the **IAM console**, choose **Agencies**, locate the **iotda_admin_trust** agency, and grant **KMS Administrator** and **OBS OperateAccess** permissions to the agency.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/ota-upgrades/packages

Table 1-660 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-661 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-662 Request body parameters

Parameter	Mandatory	Type	Description
app_id	Yes	String	**Parameter description**: resource space ID. If you have multiple resource spaces, you can use this parameter to specify the resource space that the upgrade package to create will belong to. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
package_type	Yes	String	**Parameter description**: type of the upgrade package. **Options**: For a software package, set this parameter to **softwarePackage**. For a firmware package, set this parameter to **firmwarePackage**.
product_id	Yes	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
version	Yes	String	**Parameter description**: version number of the upgrade package. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.

Parameter	Mandatory	Type	Description
support_source_versions	No	Array of strings	**Parameter description**: list of source versions that support the upgrade of this version package. Up to 20 source versions are supported. **Value**: A source version number can contain only letters, digits, underscores (_), hyphens (-), and periods (.).
description	No	String	**Parameter description**: description of the upgrade package functions. **Value length**: a maximum of 1,024 characters Maximum: **1024**
custom_info	No	String	**Parameter description**: custom information pushed to the device. After the upgrade package is added and an upgrade task is created, the IoT platform delivers the custom information to the device when delivering an upgrade notification to the device. **Value length**: a maximum of 4,096 characters Maximum: **4096**
file_location	Yes	**FileLocation** object	Location of the upgrade package.

Table 1-663 FileLocation

Parameter	Mandatory	Type	Description
obs_location	No	**ObsLocation** object	Location of the OBS object with which the upgrade package is associated.

Table 1-664 ObsLocation

Parameter	Mandatory	Type	Description
region_name	Yes	String	**Parameter description**: OBS region. You can obtain the service endpoint from **Regions and Endpoints**. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and hyphens (-) are allowed.
bucket_name	Yes	String	**Parameter description**: OBS bucket name. **Value**: The value can contain 3 to 63 characters. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
object_key	Yes	String	**Parameter description**: OBS object name (including the folder path). **Value length**: a maximum of 1,024 characters Minimum: **1** Maximum: **1024**

Response Parameters

Status code: 201

Table 1-665 Response body parameters

Parameter	Type	Description
package_id	String	**Parameter description**: upgrade package ID, which uniquely identifies an upgrade package. The value is allocated by the IoT platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, and hyphens (-) are allowed.
app_id	String	**Parameter description**: resource space ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
package_type	String	**Parameter description**: type of the upgrade package. **Options**: For a software package, set this parameter to **softwarePackage**. For a firmware package, set this parameter to **firmwarePackage**.

Parameter	Type	Description
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
version	String	**Parameter description**: version number of the upgrade package. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
support_source_versions	Array of strings	**Parameter description**: list of source versions that support the upgrade of this version package. Up to 20 source versions are supported. **Value**: A source version number can contain only letters, digits, underscores (_), hyphens (-), and periods (.).
description	String	**Parameter description**: description of the upgrade package functions. **Value length**: a maximum of 1,024 characters
custom_info	String	**Parameter description**: custom information pushed to the device. After the upgrade package is added and an upgrade task is created, the IoT platform delivers the custom information to the device when delivering an upgrade notification to the device. **Value length**: a maximum of 4,096 characters
create_time	String	Time when the software/firmware package is uploaded to the IoT platform. The parameter value is in the format of yyyyMMdd'T'HHmmss'Z'.
file_location	**FileLocation** object	Location of the upgrade package.

Table 1-666 FileLocation

Parameter	Type	Description
obs_location	**ObsLocation** object	Location of the OBS object with which the upgrade package is associated.

Table 1-667 ObsLocation

Parameter	Type	Description
region_name	String	**Parameter description**: OBS region. You can obtain the service endpoint from **Regions and Endpoints**. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and hyphens (-) are allowed.
bucket_name	String	**Parameter description**: OBS bucket name. **Value**: The value can contain 3 to 63 characters. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
object_key	String	**Parameter description**: OBS object name (including the folder path). **Value length**: a maximum of 1,024 characters Minimum: **1** Maximum: **1024**

Example Requests

- Creates an OTA upgrade package and uploads the firmware package.

```
POST https://{endpoint}/v5/iot/{project_id}/ota-upgrades/packages

{
  "app_id" : "61f7e74d036aca5be29e1ed4",
  "package_type" : "firmwarePackage",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "version" : "V2.0",
  "description" : "package v2.0",
  "custom_info" : "Updated the XX function and fixed the XXXX issue.",
  "file_location" : {
    "obs_location" : {
      "region_name" : "cn-north-4",
      "bucket_name" : "abc",
      "object_key" : "bbb/upgrade.bin"
    }
  }
}
```

- Creates an OTA upgrade package and uploads the firmware package. In the differential package scenario, the upgrade from V1.0 and V1.1 is supported.

```
POST https://{endpoint}/v5/iot/{project_id}/ota-upgrades/packages

{
  "app_id" : "61f7e74d036aca5be29e1ed4",
  "package_type" : "firmwarePackage",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "version" : "V2.0",
  "support_source_versions" : [ "V1.0", "V1.1" ],
  "description" : "package for version V1.0 and V1.1",
  "custom_info" : "Updated the XX function and fixed the XXXX issue.",
  "file_location" : {
    "obs_location" : {
      "region_name" : "cn-north-4",
      "bucket_name" : "abc",
      "object_key" : "bbb/upgrade.bin"
    }
```



```
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "package_id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
  "app_id" : "61f7e74d036aca5be29e1ed4",
  "package_type" : "firmwarePackage",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "version" : "V2.0",
  "support_source_versions" : [ "V1.0", "V1.1" ],
  "description" : "package for version V1.0 and V1.1",
  "custom_info" : "Updated the XX function and fixed the XXXX issue.",
  "create_time" : "20230211T121212Z",
  "file_location" : {
    "obs_location" : {
      "region_name" : "cn-north-4",
      "bucket_name" : "abc",
      "object_key" : "bbb/upgrade.bin"
    }
  }
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.18.2 Query the OTA Upgrade Package List

Function

This API is used to query upgrade packages associated with OBS objects. To use this API, grant IoTDA instances the permission to access OBS and the KMS Administrator permissions. On the **IAM console**, choose **Agencies**, locate the **iotda_admin_trust** agency, and grant **KMS Administrator** and **OBS OperateAccess** permissions to the agency.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/ota-upgrades/packages

Table 1-668 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-669 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. If you have multiple resource spaces, you can use this parameter to specify the resource space that the upgrade packages to query belong to. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
package_type	Yes	String	**Parameter description**: type of the upgrade package. **Options**: For a software package, set this parameter to **softwarePackage**. For a firmware package, set this parameter to **firmwarePackage**.

Parameter	Mandatory	Type	Description
product_id	No	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
version	No	String	**Parameter description**: version number of the upgrade package. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-670 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-671 Response body parameters

Parameter	Type	Description
packages	Array of **OtaPackageInfo** objects	Upgrade package list.
page	**PageInfo** object	Response structure of the batch query result pagination. It defines page numbers, the number of records on each page, the total number of records, and ID of the last record on the current page.

Table 1-672 OtaPackageInfo

Parameter	Type	Description
package_id	String	**Parameter description**: upgrade package ID, which uniquely identifies an upgrade package. The value is allocated by the IoT platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, and hyphens (-) are allowed.
app_id	String	**Parameter description**: resource space ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
package_type	String	**Parameter description**: type of the upgrade package. **Options**: For a software package, set this parameter to **softwarePackage**. For a firmware package, set this parameter to **firmwarePackage**.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
version	String	**Parameter description**: version number of the upgrade package. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
support_source_versions	Array of strings	**Parameter description**: list of source versions that support the upgrade of this version package. Up to 20 source versions are supported. **Value**: A source version number can contain only letters, digits, underscores (_), hyphens (-), and periods (.).
description	String	**Parameter description**: description of the upgrade package functions. **Value length**: a maximum of 1,024 characters
custom_info	String	**Parameter description**: custom information pushed to the device. After the upgrade package is added and an upgrade task is created, the IoT platform delivers the custom information to the device when delivering an upgrade notification to the device. **Value length**: a maximum of 4,096 characters

Parameter	Type	Description
create_time	String	Time when the software/firmware package is uploaded to the IoT platform. The parameter value is in the format of yyyyMMdd'T'HHmmss'Z'.

Table 1-673 PageInfo

Parameter	Type	Description
count	Long	Total number of records that meet the query conditions.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries OTA upgrade packages in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/ota-upgrades/packages
```

Example Responses

Status code: 200

OK

```
{
  "packages" : [ {
    "package_id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "app_id" : "61f7e74d036aca5be29e1ed4",
    "package_type" : "firmwarePackage",
    "product_id" : "5ba24f5ebbe8f56f5a14f605",
    "version" : "V2.0",
    "support_source_versions" : [ "V1.0", "V1.1" ],
    "description" : "package for version V1.0 and V1.1",
    "custom_info" : "Updated the XX function and fixed the XXXX issue.",
    "create_time" : "20230211T121212Z"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	BAD REQUEST

Status Code	Description
401	Unauthorized
403	FORBIDDEN
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.18.3 Obtain OTA Upgrade Package Details

Function

This API is used to query details about an upgrade package associated with an OBS object. To use this API, grant IoTDA instances the permission to access OBS and the KMS Administrator permissions. On the **IAM console**, choose **Agencies**, locate the **iotda_admin_trust** agency, and grant **KMS Administrator** and **OBS OperateAccess** permissions to the agency.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/ota-upgrades/packages/{package_id}

Table 1-674 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
package_id	Yes	String	**Parameter description**: upgrade package ID, which uniquely identifies an upgrade package. The value is allocated by the IoT platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, and hyphens (-) are allowed.

Request Parameters

Table 1-675 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-676 Response body parameters

Parameter	Type	Description
package_id	String	**Parameter description**: upgrade package ID, which uniquely identifies an upgrade package. The value is allocated by the IoT platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, and hyphens (-) are allowed.
app_id	String	**Parameter description**: resource space ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
package_type	String	**Parameter description**: type of the upgrade package. **Options**: For a software package, set this parameter to **softwarePackage**. For a firmware package, set this parameter to **firmwarePackage**.
product_id	String	**Parameter description**: unique ID of the product associated with the device. The value is allocated by the platform after the product is created. For details, see **Creating a Product**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
version	String	**Parameter description**: version number of the upgrade package. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
support_source_versions	Array of strings	**Parameter description**: list of source versions that support the upgrade of this version package. Up to 20 source versions are supported. **Value**: A source version number can contain only letters, digits, underscores (_), hyphens (-), and periods (.).
description	String	**Parameter description**: description of the upgrade package functions. **Value length**: a maximum of 1,024 characters
custom_info	String	**Parameter description**: custom information pushed to the device. After the upgrade package is added and an upgrade task is created, the IoT platform delivers the custom information to the device when delivering an upgrade notification to the device. **Value length**: a maximum of 4,096 characters
create_time	String	Time when the software/firmware package is uploaded to the IoT platform. The parameter value is in the format of yyyyMMdd'T'HHmmss'Z'.
file_location	**FileLocation** object	Location of the upgrade package.

Table 1-677 FileLocation

Parameter	Type	Description
obs_location	**ObsLocation** object	Location of the OBS object with which the upgrade package is associated.

Table 1-678 ObsLocation

Parameter	Type	Description
region_name	String	**Parameter description**: OBS region. You can obtain the service endpoint from **Regions and Endpoints**. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and hyphens (-) are allowed.
bucket_name	String	**Parameter description**: OBS bucket name. **Value**: The value can contain 3 to 63 characters. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
object_key	String	**Parameter description**: OBS object name (including the folder path). **Value length**: a maximum of 1,024 characters Minimum: **1** Maximum: **1024**

Example Requests

Queries OTA upgrade package details.

```
GET https://{endpoint}/v5/iot/{project_id}/ota-upgrades/packages/{package_id}
```

Example Responses

Status code: 200

OK

```
{
  "package_id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
  "app_id" : "61f7e74d036aca5be29e1ed4",
  "package_type" : "firmwarePackage",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "version" : "V2.0",
  "support_source_versions" : [ "V1.0", "V1.1" ],
  "description" : "package for version V1.0 and V1.1",
  "custom_info" : "Updated the XX function and fixed the XXXX issue.",
  "create_time" : "20230211T121212Z",
  "file_location" : {
    "obs_location" : {
      "region_name" : "cn-north-4",
      "bucket_name" : "abc",
      "object_key" : "bbb/upgrade.bin"
```



```
  }
 }
}
```

Status Codes

Status Code	Description
200	OK
401	Unauthorized
403	FORBIDDEN
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.18.4 Delete an OTA Upgrade Package

Function

This API is used to delete the upgrade package information associated with an OBS object. The OBS object will not be deleted. To use this API, grant IoTDA instances the permission to access OBS and the KMS Administrator permissions. On the **IAM console**, choose **Agencies**, locate the **iotda_admin_trust** agency, and grant **KMS Administrator** and **OBS OperateAccess** permissions to the agency.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/ota-upgrades/packages/{package_id}

Table 1-679 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
package_id	Yes	String	**Parameter description**: upgrade package ID, which uniquely identifies an upgrade package. The value is allocated by the IoT platform. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, and hyphens (-) are allowed.

Request Parameters

Table 1-680 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

None

Example Requests

Deletes an OTA upgrade package.

```
DELETE https://{endpoint}/v5/iot/{project_id}/ota-upgrades/packages/{package_id}
```


Example Responses

None

Status Codes

Status Code	Description
204	No Content
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.19 Message Broadcasting

1.4.19.1 Broadcasting a Message

Function

This API is used by an application to broadcast a message to all online devices that subscribe to a specified topic. After an application delivers a broadcast message to the platform, the platform returns a response to the application and then broadcasts the message to devices. Note:

- This API applies only to devices that use the MQTT protocol for access.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/broadcast-messages

Table 1-681 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-682 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-683 Request body parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the broadcast message belongs. If this parameter is not specified, the broadcast message will be sent to devices that subscribe the specified topic in the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
topic_full_name	Yes	String	**Parameter description**: (mandatory) complete name of the topic for receiving broadcast messages. To broadcast messages to devices, you can use this parameter to specify a complete topic name. Then, the IoT platform sends messages to all online devices that subscribe to the topic in the specified resource space. Broadcast topics do not need to be created on the console, but the topic prefix must be $oc/broadcast/. Maximum: **128**
message	Yes	String	**Parameter description**: content of the broadcast message. You need to encode the original message using Base64. The maximum length of the request body is 256 KB. Maximum: **261848**

Parameter	Mandatory	Type	Description
ttl	No	Integer	**Parameter description**: aging time of the caches of broadcasted messages on the platform, in minutes. The default value is **0**, indicating that messages are not cached. The value of **ttl** must be a multiple of 5, that is, the granularity is 5 minutes. The maximum cache duration is 1440 minutes. For longer cache duration, you need to submit a request in advance, or the delivery will fail. If this parameter is set to a value greater than 0, no more than 10 devices are allowed to subscribe to the same topic at the same time, or the API returns an error. Minimum: **0** Default: **0**
message_id	No	String	**Parameter description**: message ID, which is user-defined (UUID is recommended). If the value of **ttl** is greater than 0, the platform caches messages. You need to specify a unique value for **message_id**, or the API returns an error. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Maximum: **128**

Response Parameters

Status code: 201

Table 1-684 Response body parameters

Parameter	Type	Description
app_id	String	**Parameter description**: resource space ID.

Parameter	Type	Description
topic_full_name	String	**Parameter description**: complete name of the topic for receiving broadcast messages. Maximum: **128**
message_id	String	Message ID, which is generated by the IoT platform to identify the message.
created_time	String	UTC time when the message was created. The value is in the format of yyyyMMdd'T'HHmmss'Z'.

Example Requests

Broadcasts a message.

```
POST https://{endpoint}/v5/iot/{project_id}/broadcast-messages

{
  "topic_full_name" : "$oc/broadcast/test",
  "message" : "SGVsbG9Xb3JsZA=="
}
```

Example Responses

Status code: 201

Created

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "topic_full_name" : "$oc/broadcast/test",
  "message_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "created_time" : "20151212T121212Z"
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.20 Device Tunnel Management

1.4.20.1 Create a Device Tunnel

Function

You can use this API to create a tunnel (WebSocket protocol) through which the application and device can transmit data.

- This API is not supported by Basic editions.
- After this API is called, the IoT platform delivers the tunnel address and secret to an MQTT/MQTTS device for setting up WebSocket connections and returns the tunnel address and secret to the application.
- Multiple tunnels cannot be created on a device.
- For details, see [Remote Login] (https://support.huaweicloud.com/intl/en-us/usermanual-iothub/iot_01_00301.html).

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/tunnels

Table 1-685 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-686 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-687 Request body parameters

Parameter	Mandatory	Type	Description
device_id	Yes	String	**Parameter description**: device ID.

Response Parameters

Status code: 201

Table 1-688 Response body parameters

Parameter	Type	Description
tunnel_id	String	Tunnel ID.
tunnel_access_token	String	Authentication information.

Parameter	Type	Description
expires_in	Integer	Validity period of authentication information, in seconds. Minimum: **0** Maximum: **86400000**
tunnel_uri	String	WebSocket access address. Minimum: **1** Maximum: **2048**

Example Requests

Creates a device tunnel whose device ID is **b64b7a625b84c1334befb648b_test**.

```
POST https://{endpoint}/v5/iot/{project_id}/tunnels

{
  "device_id" : "b64b7a625b84c1334befb648b_test"
}
```

Example Responses

Status code: 201

The device tunnel is successfully created.

```
{
  "tunnel_id" : "d144a524-1997-4b99-94bf-f27128da8a34",
  "tunnel_access_token" : "MIIDkgYJKoZIhvcNAQcCoIIDgzCCXXXXX",
  "expires_in" : 86400,
  "tunnel_uri" : "wss://tunnel.st1.iotda-app.cn-XXX.myhuaweicloud.com/v5/iot/tunnels/XXX/source-connect"
}
```

Status Codes

Status Code	Description
201	The device tunnel is successfully created.
400	Invalid value.
401	Unauthorized
403	Authentication failed.
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.20.2 Query All Tunnels of a Device

Function

You can use this API to query the data of all device tunnels in a project for device management. This API is used by an application to query the status of device tunnels from the IoT platform.

- This API is not supported by Basic editions.
- For details, see [Remote Login] (https://support.huaweicloud.com/intl/en-us/usermanual-iothub/iot_01_00301.html).

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/tunnels

Table 1-689 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-690 Query Parameters

Parameter	Mandatory	Type	Description
device_id	No	String	**Parameter description**: device ID.

Request Parameters

Table 1-691 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-692 Response body parameters

Parameter	Type	Description
tunnels	Array of **TunnelInfo** objects	Tunnel information list. Array Length: **0 - 100**

Table 1-693 TunnelInfo

Parameter	Type	Description
tunnel_id	String	Tunnel ID.
device_id	String	Device ID.

Parameter	Type	Description
create_time	String	Time when a tunnel was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
closed_time	String	Time when a tunnel was updated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
status	String	Tunnel status. Options: **CLOSED** and **OPEN**.
source_connect_state	**ConnectState** object	Access end (console) status.
device_connect_state	**ConnectState** object	Device status.

Table 1-694 ConnectState

Parameter	Type	Description
last_update_time	String	Time when the latest tunnel status was updated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
status	String	Client connection status. Options: **CONNECTED** and **DISCONNECTED**.

Example Requests

Queries device tunnels in a list.

```
GET https://{endpoint}/v5/iot/{project_id}/tunnels
```

Example Responses

Status code: 200

All tunnel information of the device is successfully queried.

```
{
  "tunnels" : [ {
    "tunnel_id" : "d144a524-1997-4b99-94bf-f27128da8a34",
    "device_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
    "create_time" : "20190303T081011Z",
    "closed_time" : "20190303T081011Z",
    "status" : "CLOSED",
    "source_connect_state" : {
      "last_update_time" : "20190303T081011Z",
      "status" : "CONNECTED"
    },
    "device_connect_state" : {
      "last_update_time" : "20190303T081011Z",
      "status" : "CONNECTED"
```



```
  }
 }]
}
```

Status Codes

Status Code	Description
200	All tunnel information of the device is successfully queried.
400	Invalid value.
401	Unauthorized
403	Authentication failed.
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.20.3 Query Device Tunnels

Function

You can use this API to query the information and connection status of a device tunnel in a project. This API is used by an application to query the status of device tunnels from the IoT platform.

- This API is not supported by Basic editions.
- For details, see [Remote Login] (https://support.huaweicloud.com/intl/en-us/usermanual-iothub/iot_01_00301.html).

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/tunnels/{tunnel_id}

Table 1-695 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
tunnel_id	Yes	String	Tunnel ID. Minimum: **1** Maximum: **128**

Request Parameters

Table 1-696 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-697 Response body parameters

Parameter	Type	Description
tunnel_id	String	Tunnel ID.
device_id	String	Device ID.

Parameter	Type	Description
create_time	String	Time when a tunnel was created. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
closed_time	String	Time when a tunnel was updated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
status	String	Tunnel status. Options: **CLOSED** and **OPEN**.
source_connect_state	**ConnectState** object	Access end (console) status.
device_connect_state	**ConnectState** object	Device status.

Table 1-698 ConnectState

Parameter	Type	Description
last_update_time	String	Time when the latest tunnel status was updated. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
status	String	Client connection status. Options: **CONNECTED** and **DISCONNECTED**.

Example Requests

Queries tunnel details of a device.

```
GET https://{endpoint}/v5/iot/{project_id}/tunnels/{tunnel_id}
```

Example Responses

Status code: 200

OK

```
{
  "tunnel_id" : "d144a524-1997-4b99-94bf-f27128da8a34",
  "device_id" : "1a7ffc5c-d89c-44dd-8265-b1653d951ce0",
  "create_time" : "20190303T081011Z",
  "closed_time" : "20190303T081011Z",
  "status" : "CLOSED",
  "source_connect_state" : {
    "last_update_time" : "20190303T081011Z",
    "status" : "CONNECTED"
  },
  "device_connect_state" : {
    "last_update_time" : "20190303T081011Z",
    "status" : "CONNECTED"
```



```
 }
}
```

Status Codes

Status Code	Description
200	OK
400	Invalid value.
401	Unauthorized
403	Authentication failed.
404	The tunnel does not exist.
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.20.4 Disable a Device Tunnel

Function

This API is used by an application to disable a device tunnel. After the disabling, the tunnel can be connected again.

- This API is not supported by Basic editions.
- For details, see [Remote Login] (https://support.huaweicloud.com/intl/en-us/usermanual-iothub/iot_01_00301.html).

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/tunnels/{tunnel_id}

Table 1-699 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
tunnel_id	Yes	String	Tunnel ID. Minimum: **1** Maximum: **128**

Request Parameters

Table 1-700 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Disables a device tunnel.

```
PUT https://{endpoint}/v5/iot/{project_id}/tunnels/{tunnel_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	no content
400	Invalid value.
401	Unauthorized
403	Authentication failed.
404	The tunnel does not exist.
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.20.5 Delete a Device Tunnel

Function

You can use this API to delete a device tunnel. After the deletion, the channel does not exist and cannot be connected again.

- This API is not supported by Basic editions.
- For details, see [Remote Login] (https://support.huaweicloud.com/intl/en-us/usermanual-iothub/iot_01_00301.html).

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/tunnels/{tunnel_id}

Table 1-701 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
tunnel_id	Yes	String	Tunnel ID. Minimum: **1** Maximum: **128**

Request Parameters

Table 1-702 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a device tunnel.

```
DELELE https://{endpoint}/v5/iot/{project_id}/tunnels/{tunnel_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	no content
400	Invalid value.
401	Unauthorized
403	Authentication failed.
404	The tunnel does not exist.
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.21 Stack policy management

1.4.21.1 Create a Data Transfer Stacking Policy

Function

This API is used by an application to create a data transfer and stacking policy in the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/routing-rule/backlog-policy

Table 1-703 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-704 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-705 Request body parameters

Parameter	Mandatory	Type	Description
policy_name	No	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: Description of a custom data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
backlog_size	No	Integer	**Parameter description**: Size of stacked data. The value is an integer ranging from 0 to 1073741823, in bytes. The default value is **1073741823** (1 GB). If the value of **backlog_size** is 0, logs are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **1073741823** Default: **1073741823**
backlog_time	No	Integer	**Parameter description**: Data backlog time. The value is an integer ranging from 0 to 86399, in seconds. The default value is **86399** (one day). If the value of **backlog_time** is 0, messages are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **86399** Default: **86399**

Response Parameters

Status code: 201

Table 1-706 Response body parameters

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.
policy_name	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	String	**Parameter description**: Description of a custom data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
backlog_size	Integer	**Parameter description**: Size of stacked data. The value is an integer ranging from 0 to 1073741823, in bytes. The default value is **1073741823** (1 GB). If the value of **backlog_size** is 0, logs are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **1073741823** Default: **1073741823**
backlog_time	Integer	**Parameter description**: Data backlog time. The value is an integer ranging from 0 to 86399, in seconds. The default value is **86399** (one day). If the value of **backlog_time** is 0, messages are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **86399** Default: **86399**

Example Requests

Creates a stack policy.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/backlog-policy

{
  "policy_name" : "rulename",
  "description" : "description",
  "backlog_size" : 100,
  "backlog_time" : 100
}
```

Example Responses

Status code: 201

Created

```
{
  "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
  "policy_name" : "policyName",
  "description" : "description",
  "backlog_size" : 1073741823,
  "backlog_time" : 86399
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.21.2 Query the Data Transfer Stacking Policy List

Function

This API is used by an application to query the list of data transfer and stacking policies configured on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/backlog-policy

Table 1-707 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-708 Query Parameters

Parameter	Mandatory	Type	Description
policy_name	No	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
limit	No	Integer	**Parameter description**: Number of records to display on each page. By default, 10 records are displayed on each page. A maximum of 50 records can be displayed on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. - To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-709 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-710 Response body parameters

Parameter	Type	Description
backlog_policies	Array of **BacklogPolicyInfo** objects	List of data transfer stacking policies. Array Length: **0 - 50**
count	Integer	Total number of records that meet the filter criteria. Minimum: **0**
marker	String	ID of the last record in this query, which can be used in the next query.

Table 1-711 BacklogPolicyInfo

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.
policy_name	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	String	**Parameter description**: Description of a custom data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
backlog_size	Integer	**Parameter description**: Size of stacked data. The value is an integer ranging from 0 to 1073741823, in bytes. The default value is **1073741823** (1 GB). If the value of **backlog_size** is 0, logs are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **1073741823** Default: **1073741823**
backlog_time	Integer	**Parameter description**: Data backlog time. The value is an integer ranging from 0 to 86399, in seconds. The default value is **86399** (one day). If the value of **backlog_time** is 0, messages are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **86399** Default: **86399**

Example Requests

Queries the stack policy list.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/backlog-policy
```

Example Responses

Status code: 200

Ok

```
{
  "backlog_policies" : [ {
    "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
    "policy_name" : "policyName",
    "description" : "description",
    "backlog_size" : 1073741823,
    "backlog_time" : 86399
  } ],
  "count" : 10,
  "marker" : "5c90fa7d3c4e4405e8525079"
}
```

Status Codes

Status Code	Description
200	Ok
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.21.3 Modify a Data Transfer Stacking Policy

Function

This API is used by an application to modify a specified data transfer and stacking policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/routing-rule/backlog-policy/{policy_id}

Table 1-712 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.

Request Parameters

Table 1-713 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-714 Request body parameters

Parameter	Mandatory	Type	Description
policy_name	No	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	No	String	**Parameter description**: Description of a custom data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
backlog_size	No	Integer	**Parameter description**: Size of stacked data. The value is an integer ranging from 0 to 1073741823, in bytes. The default value is **1073741823** (1 GB). If the value of **backlog_size** is 0, logs are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **1073741823** Default: **1073741823**

Parameter	Mandatory	Type	Description
backlog_time	No	Integer	**Parameter description**: Data backlog time. The value is an integer ranging from 0 to 86399, in seconds. The default value is **86399** (one day). If the value of **backlog_time** is 0, messages are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **86399** Default: **86399**

Response Parameters

Status code: 200

Table 1-715 Response body parameters

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.
policy_name	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	String	**Parameter description**: Description of a custom data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
backlog_size	Integer	**Parameter description**: Size of stacked data. The value is an integer ranging from 0 to 1073741823, in bytes. The default value is **1073741823** (1 GB). If the value of **backlog_size** is 0, logs are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **1073741823** Default: **1073741823**
backlog_time	Integer	**Parameter description**: Data backlog time. The value is an integer ranging from 0 to 86399, in seconds. The default value is **86399** (one day). If the value of **backlog_time** is 0, messages are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **86399** Default: **86399**

Example Requests

Update a stack policy.

```
PUT https://{endpoint}/v5/iot/{project_id}/routing-rule/backlog-policy/{policy_id}

{
  "backlog_size" : 100,
  "backlog_time" : 100
}
```

Example Responses

Status code: 200

Ok

```
{
  "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
  "policy_name" : "policyName",
  "description" : "description",
  "backlog_size" : 1073741823,
  "backlog_time" : 86399
}
```


Status Codes

Status Code	Description
200	Ok
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.21.4 Query the Data Transfer Stacking Policy

Function

This API is used by an application to query a specified data transfer and stacking policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/backlog-policy/{policy_id}

Table 1-716 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.

Request Parameters

Table 1-717 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-718 Response body parameters

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.
policy_name	String	**Parameter description**: Name of a data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
description	String	**Parameter description**: Description of a custom data transfer stacking policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
backlog_size	Integer	**Parameter description**: Size of stacked data. The value is an integer ranging from 0 to 1073741823, in bytes. The default value is **1073741823** (1 GB). If the value of **backlog_size** is 0, logs are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **1073741823** Default: **1073741823**
backlog_time	Integer	**Parameter description**: Data backlog time. The value is an integer ranging from 0 to 86399, in seconds. The default value is **86399** (one day). If the value of **backlog_time** is 0, messages are not stacked. If both **backlog_size** and **backlog_time** are configured, the dimension that first reaches the threshold is used. Minimum: **0** Maximum: **86399** Default: **86399**

Example Requests

Queries a specified stack policy.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/backlog-policy/{policy_id}
```

Example Responses

Status code: 200

Ok

```
{
  "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
  "policy_name" : "policyName",
  "description" : "description",
  "backlog_size" : 1073741823,
```



```
  "backlog_time" : 86399
}
```

Status Codes

Status Code	Description
200	Ok
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.21.5 Delete a Data Transfer Stacking Policy

Function

This API is used by an application to delete a specified data transfer and stacking policy from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/routing-rule/backlog-policy/{policy_id}

Table 1-719 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
policy_id	Yes	String	**Parameter description**: ID of the data stacking policy. The value of this parameter is unique and is allocated by the platform during data stacking policy creation.

Request Parameters

Table 1-720 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

None

Example Requests

Deletes a specified stack policy.

```
DELETE https://{endpoint}/v5/iot/{project_id}/routing-rule/backlog-policy/{policy_id}
```


Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.22 Flow control policy management

1.4.22.1 Create a Data Forwarding Flow Control Policy

Function

This API is used by an application to create a data forwarding flow control policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/routing-rule/flowcontrol-policy

Table 1-721 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-722 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-723 Request body parameters

Parameter	Mandatory	Type	Description
policy_name	No	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: Description of a custom data transfer flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
scope	No	String	**Parameter description**: Scope of a flow control policy. **Options**: • **USER**: tenant-level flow control policy • **CHANNEL**: forwarding channel-level flow control policy • **RULE**: forwarding rule-level flow control policy • **ACTION**: forwarding action-level flow control policy
scope_value	No	String	**Parameter description**: Additional value of the flow control policy scope. If **scope** is set to **USER**, this parameter is optional, indicating tenant-level flow control. If **scope** is set to **CHANNEL**, this parameter can be set to **HTTP_FORWARDING**, **DIS_FORWARDING**, **OBS_FORWARDING**, **AMQP_FORWARDING**, or **DMS_KAFKA_FORWARDING**. If **scope** is set to **RULE**, set this parameter to the corresponding rule ID. If **scope** is set to **ACTION**, set this parameter to the corresponding action ID.

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: Flow control size for data forwarding. The value is an integer ranging from 1 to 1000, in TPS. The default value is **1000**. Minimum: **1** Maximum: **1000** Default: **1000**

Response Parameters

Status code: 201

Table 1-724 Response body parameters

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.
policy_name	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	String	**Parameter description**: Description of a custom data transfer flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
scope	String	**Parameter description**: Scope of a flow control policy. **Options**: • **USER**: tenant-level flow control policy • **CHANNEL**: forwarding channel-level flow control policy • **RULE**: forwarding rule-level flow control policy • **ACTION**: forwarding action-level flow control policy
scope_value	String	**Parameter description**: Additional value of the flow control policy scope. If **scope** is set to **USER**, this parameter is optional, indicating tenant-level flow control. If **scope** is set to **CHANNEL**, this parameter can be set to **HTTP_FORWARDING**, **DIS_FORWARDING**, **OBS_FORWARDING**, **AMQP_FORWARDING**, or **DMS_KAFKA_FORWARDING**. If **scope** is set to **RULE**, set this parameter to the corresponding rule ID. If **scope** is set to **ACTION**, set this parameter to the corresponding action ID.
limit	Integer	**Parameter description**: Flow control size for data forwarding. The value is an integer ranging from 1 to 1000, in TPS. The default value is **1000**. Minimum: **1** Maximum: **1000** Default: **1000**

Example Requests

- Creates an instance-level flow control policy.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy

{
  "policy_name" : "policy_name",
  "description" : "description",
  "scope" : "USER",
  "limit" : 100
}
```

- Creates a channel-level flow control policy.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy

{
  "policy_name" : "policy_name",
  "description" : "description",
  "scope" : "CHANNEL",
  "scope_value" : "HTTP_FORWARDING",
```



```
  "limit" : 100
}
```

- Creates a rule-level flow control policy.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy

{
  "policy_name" : "policy_name",
  "description" : "description",
  "scope" : "RULE",
  "scope_value" : "b0443335-2627-4ebe-bdef-276113646520",
  "limit" : 100
}
```

- Creates an action-level flow control policy.

```
POST https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy

{
  "policy_name" : "policy_name",
  "description" : "description",
  "scope" : "ACTION",
  "scope_value" : "b0443335-2627-4ebe-bdef-276113646520",
  "limit" : 100
}
```

Example Responses

Status code: 201

Created

```
{
  "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
  "policy_name" : "policyName",
  "description" : "description",
  "scope" : "CHANNEL",
  "scope_value" : "HTTP_FORWARDING",
  "limit" : 10
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.22.2 Query a List of Data Forwarding Flow Control Policies

Function

This API is used by an application to query a list of data forwarding flow control policies configured on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/flowcontrol-policy

Table 1-725 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-726 Query Parameters

Parameter	Mandatory	Type	Description
scope	No	String	**Parameter description**: Scope of a flow control policy. If this parameter is not carried, flow control policies in all scopes are queried. If this parameter is set to **USER**, tenant-level flow control policies are queried. If this parameter is set to **CHANNEL**, forwarding channel-level flow control policies are queried. If this parameter is set to **RULE**, forwarding rule-level flow control policies are queried. If this parameter is set to **ACTION**, the forwarding action-level flow control policy is queried.

Parameter	Mandatory	Type	Description
scope_value	No	String	**Parameter description**: Additional value of the flow control policy scope. If **scope** is not specified or is set to **USER**, this parameter can be left empty to query tenant-level flow control policies. If **scope** is set to **CHANNEL** and this parameter is not carried, the policies applied to all forwarding channels are queried. If this parameter is carried, the policy applied to the specified channel is queried. ** Value range**: **HTTP_FORWARDING**, **DIS_FORWARDING**, **OBS_FORWARDING**, **AMQP_FORWARDING**, and **DMS_KAFKA_FORWARDING**. If **scope** is set to **RULE** and this parameter is not carried, the policies applied to all forwarding rules are queried. If this parameter is carried, the policy applied to the specified rule is queried. If **scope** is set to **ACTION** and this parameter is not carried, the policies applied to all forwarding actions are queried. If this parameter is carried, the policy applied to the specified action is queried.
policy_name	No	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: Number of records to display on each page. By default, 10 records are displayed on each page. A maximum of 50 records can be displayed on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: The value is a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. - To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-727 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-728 Response body parameters

Parameter	Type	Description
flowcontrol_policies	Array of **FlowControlPolicyInfo** objects	List of data forwarding flow control policies. Array Length: **0 - 50**
count	Integer	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Table 1-729 FlowControlPolicyInfo

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.
policy_name	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	String	**Parameter description**: Description of a custom data transfer flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
scope	String	**Parameter description**: Scope of a flow control policy. **Options**: • **USER**: tenant-level flow control policy • **CHANNEL**: forwarding channel-level flow control policy • **RULE**: forwarding rule-level flow control policy • **ACTION**: forwarding action-level flow control policy
scope_value	String	**Parameter description**: Additional value of the flow control policy scope. If **scope** is set to **USER**, this parameter is optional, indicating tenant-level flow control. If **scope** is set to **CHANNEL**, this parameter can be set to **HTTP_FORWARDING**, **DIS_FORWARDING**, **OBS_FORWARDING**, **AMQP_FORWARDING**, or **DMS_KAFKA_FORWARDING**. If **scope** is set to **RULE**, set this parameter to the corresponding rule ID. If **scope** is set to **ACTION**, set this parameter to the corresponding action ID.

Parameter	Type	Description
limit	Integer	**Parameter description**: Flow control size for data forwarding. The value is an integer ranging from 1 to 1000, in TPS. The default value is **1000**. Minimum: **1** Maximum: **1000** Default: **1000**

Example Requests

Queries the flow control policy list.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy
```

Example Responses

Status code: 200

Ok

```
{
  "flowcontrol_policies" : [ {
    "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
    "policy_name" : "policyName",
    "description" : "description",
    "scope" : "CHANNEL",
    "scope_value" : "HTTP_FORWARDING",
    "limit" : 10
  } ],
  "count" : 10,
  "marker" : "5c90fa7d3c4e4405e8525079"
}
```

Status Codes

Status Code	Description
200	Ok
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.22.3 Modify a Data Forwarding Flow Control Policy

Function

This API is used by an application to modify a specified data forwarding flow control policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/routing-rule/flowcontrol-policy/{policy_id}

Table 1-730 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.

Request Parameters

Table 1-731 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Table 1-732 Request body parameters

Parameter	Mandatory	Type	Description
policy_name	No	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	No	String	**Parameter description**: Description of a custom data transfer flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
limit	No	Integer	**Parameter description**: Flow control size for data forwarding. The value is an integer ranging from 1 to 1000, in TPS. The default value is **1000**. Minimum: **1** Maximum: **1000** Default: **1000**

Response Parameters

Status code: 200

Table 1-733 Response body parameters

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.
policy_name	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
description	String	**Parameter description**: Description of a custom data transfer flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
scope	String	**Parameter description**: Scope of a flow control policy. **Options**: • **USER**: tenant-level flow control policy • **CHANNEL**: forwarding channel-level flow control policy • **RULE**: forwarding rule-level flow control policy • **ACTION**: forwarding action-level flow control policy
scope_value	String	**Parameter description**: Additional value of the flow control policy scope. If **scope** is set to **USER**, this parameter is optional, indicating tenant-level flow control. If **scope** is set to **CHANNEL**, this parameter can be set to **HTTP_FORWARDING**, **DIS_FORWARDING**, **OBS_FORWARDING**, **AMQP_FORWARDING**, or **DMS_KAFKA_FORWARDING**. If **scope** is set to **RULE**, set this parameter to the corresponding rule ID. If **scope** is set to **ACTION**, set this parameter to the corresponding action ID.

Parameter	Type	Description
limit	Integer	**Parameter description**: Flow control size for data forwarding. The value is an integer ranging from 1 to 1000, in TPS. The default value is **1000**. Minimum: **1** Maximum: **1000** Default: **1000**

Example Requests

Modifies a flow control policy.

```
PUT https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy/{policy_id}

{
  "policy_name" : "policyName",
  "description" : "description",
  "limit" : 1000
}
```

Example Responses

Status code: 200

Ok

```
{
  "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
  "policy_name" : "policyName",
  "description" : "description",
  "scope" : "CHANNEL",
  "scope_value" : "HTTP_FORWARDING",
  "limit" : 10
}
```

Status Codes

Status Code	Description
200	Ok
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.22.4 Query a Specified Data Forwarding Flow Control Policy

Function

This API is used by an application to query a specified data forwarding flow control policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/routing-rule/flowcontrol-policy/{policy_id}

Table 1-734 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.

Request Parameters

Table 1-735 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

Status code: 200

Table 1-736 Response body parameters

Parameter	Type	Description
policy_id	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.
policy_name	String	**Parameter description**: Name of a data flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**

Parameter	Type	Description
description	String	**Parameter description**: Description of a custom data transfer flow control policy. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **256**
scope	String	**Parameter description**: Scope of a flow control policy. **Options**: • **USER**: tenant-level flow control policy • **CHANNEL**: forwarding channel-level flow control policy • **RULE**: forwarding rule-level flow control policy • **ACTION**: forwarding action-level flow control policy
scope_value	String	**Parameter description**: Additional value of the flow control policy scope. If **scope** is set to **USER**, this parameter is optional, indicating tenant-level flow control. If **scope** is set to **CHANNEL**, this parameter can be set to **HTTP_FORWARDING**, **DIS_FORWARDING**, **OBS_FORWARDING**, **AMQP_FORWARDING**, or **DMS_KAFKA_FORWARDING**. If **scope** is set to **RULE**, set this parameter to the corresponding rule ID. If **scope** is set to **ACTION**, set this parameter to the corresponding action ID.
limit	Integer	**Parameter description**: Flow control size for data forwarding. The value is an integer ranging from 1 to 1000, in TPS. The default value is **1000**. Minimum: **1** Maximum: **1000** Default: **1000**

Example Requests

Queries a specified flow control policy.

```
GET https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy/{policy_id}
```

Example Responses

Status code: 200

Ok

```
{
  "policy_id" : "adadd5cb-6383-4b5b-a65c-f8c92fdf3c34",
  "policy_name" : "policyName",
  "description" : "description",
  "scope" : "CHANNEL",
  "scope_value" : "HTTP_FORWARDING",
  "limit" : 10
}
```

Status Codes

Status Code	Description
200	Ok
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.22.5 Delete a Specified Data Forwarding Flow Control Policy

Function

This API is used by an application to delete a specified data forwarding flow control policy from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/routing-rule/flowcontrol-policy/{policy_id}

Table 1-737 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	**Parameter description**: ID of the flow control policy. The value of this parameter is unique and is allocated by the platform during policy creation.

Request Parameters

Table 1-738 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: Instance ID. This parameter is required only when the API is called from the management plane in the physical multi-tenant scenario. You can log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID.

Response Parameters

None

Example Requests

Deletes a specified flow control policy.

```
DELETE https://{endpoint}/v5/iot/{project_id}/routing-rule/flowcontrol-policy/{policy_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.23 Device Proxy

1.4.23.1 Create a Device Proxy

Function

This API is used by an application to create a dynamic device proxy rule on the IoT platform for a child device to select a gateway to go online and report messages. That is, a child device under any gateway in the proxy group can go online through other devices in the proxy group. For details, see **Gateway Updating Child Device Status**) and **Gateway Reporting Device Properties in Batches**).

- Max. device proxies for an instance: 10.
- Max. TPS for an account to call this API: 1 (one request per second).

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-proxies

Table 1-739 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-740 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-741 Request body parameters

Parameter	Mandatory	Type	Description
proxy_name	Yes	String	**Parameter description**: device proxy name. Minimum: **1** Maximum: **64**
proxy_devices	Yes	Array of strings	**Parameter description**: device proxy list. All devices in the list share the gateway permission. That is, child devices under any gateway in the list can go online through any gateway in the group and report data. **Value**: Enter two to ten device IDs in the list. A device ID can contain up to 128 characters, including letters, digits, underscores (_), and hyphens (-). Recommended min. ID length: 4 characters.
effective_time_range	Yes	**EffectiveTimeRange** object	**Parameter Description**: validity period of a device proxy rule.
app_id	Yes	String	**Parameter description**: resource space ID, which specifies the resource space to which the created device belongs. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Table 1-742 EffectiveTimeRange

Parameter	Mandatory	Type	Description
start_time	No	String	Time when the device proxy starts to take effect. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Parameter	Mandatory	Type	Description
end_time	No	String	Time when the device proxy expires. The value must be later than the value of **start_time**. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Response Parameters

Status code: 201

Table 1-743 Response body parameters

Parameter	Type	Description
proxy_id	String	**Parameter description**: device proxy ID, which uniquely identifies a proxy rule.
proxy_name	String	**Parameter description**: device proxy name.
proxy_devices	Array of strings	**Parameter description**: device proxy group. All devices in the group share the gateway permission. That is, child devices under any gateway in the group can go online through any gateway in the group and report data.
effective_time_range	**EffectiveTimeRangeResponseDTO** object	**Parameter Description**: validity period of a device proxy rule.
app_id	String	**Parameter description**: resource space ID.

Table 1-744 EffectiveTimeRangeResponseDTO

Parameter	Type	Description
start_time	String	Time when the device proxy starts to take effect. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.
end_time	String	Time when the device proxy expires. The value must be later than the value of **start_time**. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Example Requests

Creates a device proxy.


```
POST https://{endpoint}/v5/iot/{project_id}/device-proxies

{
  "proxy_name" : "testAPP01",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "proxy_devices" : [ "d4922d8a-6c8e-4396-852c-164aefa6638f",
"d4922d8a-6c8e-4396-852c-164aefa6638g" ],
  "effective_time_range" : {
    "start_time" : "20200812T121212Z",
    "end_time" : "20210812T121212Z"
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "proxy_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
  "proxy_name" : "testAPP01",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "proxy_devices" : [ "d4922d8a-6c8e-4396-852c-164aefa6638f",
"d4922d8a-6c8e-4396-852c-164aefa6638g" ],
  "effective_time_range" : {
    "start_time" : "20200812T121212Z",
    "end_time" : "20210812T121212Z"
  }
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.23.2 Query Device Proxy List

Function

This API is used by an application to query the device proxy list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-proxies

Table 1-745 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-746 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can specify this parameter to query the device proxy list in the specified resource space. If this parameter is not carried, device proxies in all resource spaces are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
proxy_name	No	String	**Parameter description**: device proxy name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-747 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-748 Response body parameters

Parameter	Type	Description
device_proxies	Array of **QueryDeviceProxySimplify** objects	Proxy device list.
page	**Page** object	Pagination information of the query results.

Table 1-749 QueryDeviceProxySimplify

Parameter	Type	Description
proxy_id	String	**Parameter description**: device proxy ID, which uniquely identifies a proxy rule.
proxy_name	String	**Parameter description**: device proxy name.
effective_time_range	**EffectiveTimeRangeResponseDTO** object	**Parameter Description**: rule validity period.
app_id	String	**Parameter description**: resource space ID.

Table 1-750 EffectiveTimeRangeResponseDTO

Parameter	Type	Description
start_time	String	Time when the device proxy starts to take effect. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.
end_time	String	Time when the device proxy expires. The value must be later than the value of **start_time**. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Table 1-751 Page

Parameter	Type	Description
count	Long	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

```
GET https://{endpoint}/v5/iot/{project_id}/device-proxies
```

Example Responses

Status code: 200

OK

```
{
  "device_proxies" : [ {
    "proxy_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
    "proxy_name" : "testProxyName",
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
```



```
    "effective_time_range" : {
      "start_time" : "20200812T121212Z",
      "end_time" : "20210812T121212Z"
    }
  } ],
  "page" : {
    "count" : 1,
    "marker" : "66178add3b98944277731d0a"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.23.3 Query Device Proxy Details

Function

This API is used by an application to query the details of a specific device proxy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-proxies/{proxy_id}

Table 1-752 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
proxy_id	Yes	String	**Parameter description**: device proxy ID, which uniquely identifies a device proxy. The value of this parameter is allocated by the IoT platform during device proxy registration.

Request Parameters

Table 1-753 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-754 Response body parameters

Parameter	Type	Description
proxy_id	String	**Parameter description**: device proxy ID, which uniquely identifies a proxy rule.
proxy_name	String	**Parameter description**: device proxy name.
proxy_devices	Array of strings	**Parameter description**: device proxy group. All devices in the group share the gateway permission. That is, child devices under any gateway in the group can go online through any gateway in the group and report data.
effective_time_range	**EffectiveTimeRangeResponseDTO** object	**Parameter Description**: validity period of a device proxy rule.
app_id	String	**Parameter description**: resource space ID.

Table 1-755 EffectiveTimeRangeResponseDTO

Parameter	Type	Description
start_time	String	Time when the device proxy starts to take effect. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.
end_time	String	Time when the device proxy expires. The value must be later than the value of **start_time**. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Example Requests

```
GET https://{endpoint}/v5/iot/{project_id}/device-proxies/{proxy_id}
```

Example Responses

Status code: 200

OK

```
{
  "proxy_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
  "proxy_name" : "testAPP01",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "proxy_devices" : [ "d4922d8a-6c8e-4396-852c-164aefa6638f",
"d4922d8a-6c8e-4396-852c-164aefa6638g" ],
  "effective_time_range" : {
    "start_time" : "20200812T121212Z",
    "end_time" : "20210812T121212Z"
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.23.4 Modify a Device Proxy

Function

This API is used by an application to modify the basic information of a specific device proxy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/device-proxies/{proxy_id}

Table 1-756 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
proxy_id	Yes	String	**Parameter description**: device proxy ID, which uniquely identifies a device proxy. The value of this parameter is allocated by the IoT platform during device proxy registration.

Request Parameters

Table 1-757 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-758 Request body parameters

Parameter	Mandatory	Type	Description
proxy_name	No	String	**Parameter description**: device proxy name. Minimum: **1** Maximum: **64**
proxy_devices	No	Array of strings	**Parameter description**: device proxy list. All devices in the list share the gateway permission. That is, child devices under any gateway in the list can go online through any gateway in the group and report data. **Value**: Enter two to ten device IDs in the list. A device ID can contain up to 128 characters, including letters, digits, underscores (_), and hyphens (-). Recommended min. ID length: 4 characters.
effective_time_range	No	**EffectiveTimeRange** object	**Parameter Description**: rule validity period.

Table 1-759 EffectiveTimeRange

Parameter	Mandatory	Type	Description
start_time	No	String	Time when the device proxy starts to take effect. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.
end_time	No	String	Time when the device proxy expires. The value must be later than the value of **start_time**. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Response Parameters

Status code: 200

Table 1-760 Response body parameters

Parameter	Type	Description
proxy_id	String	**Parameter description**: device proxy ID, which uniquely identifies a proxy rule.
proxy_name	String	**Parameter description**: device proxy name.
proxy_devices	Array of strings	**Parameter description**: device proxy group. All devices in the group share the gateway permission. That is, child devices under any gateway in the group can go online through any gateway in the group and report data.
effective_time_range	**EffectiveTimeRangeResponseDTO** object	**Parameter Description**: validity period of a device proxy rule.
app_id	String	**Parameter description**: resource space ID.

Table 1-761 EffectiveTimeRangeResponseDTO

Parameter	Type	Description
start_time	String	Time when the device proxy starts to take effect. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.
end_time	String	Time when the device proxy expires. The value must be later than the value of **start_time**. Format (UTC time): yyyyMMdd'T'HHmmss'Z'.

Example Requests

Modifies a device proxy.

```
PUT https://{endpoint}/v5/iot/{project_id}/device-proxies/{proxy_id}

{
  "proxy_name" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "proxy_devices" : [ "d4922d8a-6c8e-4396-852c-164aefa6638f",
"d4922d8a-6c8e-4396-852c-164aefa6638g" ],
  "effective_time_range" : {
    "start_time" : "20200812T121212Z",
    "end_time" : "20200812T121212Z"
  }
}
```

Example Responses

Status code: 200

OK


```
{
  "proxy_id" : "04ed32dc1b0025b52fe3c01a27c2babc",
  "proxy_name" : "testAPP01",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "proxy_devices" : [ "d4922d8a-6c8e-4396-852c-164aefa6638f",
"d4922d8a-6c8e-4396-852c-164aefa6638g" ],
  "effective_time_range" : {
    "start_time" : "20200812T121212Z",
    "end_time" : "20210812T121212Z"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.23.5 Delete a Device Proxy

Function

This API is used by an application to delete a specific device proxy from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/device-proxies/{proxy_id}

Table 1-762 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
proxy_id	Yes	String	**Parameter description**: device proxy ID, which uniquely identifies a device proxy. The value of this parameter is allocated by the IoT platform during device proxy registration.

Request Parameters

Table 1-763 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. You can obtain the token by calling the IAM API **Obtaining a User Token Through Password Authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

```
DELETE https://{endpoint}/v5/iot/{project_id}/device-proxies/{proxy_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.24 Bridge Management

1.4.24.1 Create a Bridge

Function

This API is used by an application to create a bridge on the IoT platform. Only the created bridge can connect to the platform.

- An instance supports a maximum of 20 bridges.
- This API is supported only by **standard and enterprise editions**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/bridges

Table 1-764 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-765 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-766 Request body parameters

Parameter	Mandatory	Type	Description
bridge_name	Yes	String	Bridge name. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Minimum: **1** Maximum: **64**
bridge_id	No	String	Bridge ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Response Parameters

Status code: 201

Table 1-767 Response body parameters

Parameter	Type	Description
bridge_id	String	Bridge ID, which uniquely identifies a bridge. The value of this parameter is specified during bridge registration or allocated by the platform.
bridge_name	String	Bridge name.
auth_info	**BridgeAuthInfo** object	Bridge authentication information.
create_time	String	Time when the bridge was registered on the platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-768 BridgeAuthInfo

Parameter	Type	Description
auth_type	String	Authentication mode. Currently, secret authentication (SECRET) is supported. If you use secret authentication, specify the **secret** parameter. If **auth_type** is not set, the secret authentication is used.

Parameter	Type	Description
secret	String	Bridge secret. This parameter is mandatory when **auth_type** is **SECRET**.

Example Requests

```
POST https://{endpoint}/v5/iot/{project_id}/bridges

{
  "bridge_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "bridge_name" : "dianabridge"
}
```

Example Responses

Status code: 201

Created

```
{
  "bridge_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "bridge_name" : "dianabridge",
  "auth_info" : {
    "auth_type" : "SECRET",
    "secret" : "3b935******dc3c"
  }
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.24.2 Query the Bridge List

Function

This API is used by an application to query the bridge list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/bridges

Table 1-769 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-770 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-771 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-772 Response body parameters

Parameter	Type	Description
bridges	Array of **BridgeResponse** objects	Bridge list.
page	**Page** object	Pagination information of the query results.

Table 1-773 BridgeResponse

Parameter	Type	Description
bridge_id	String	Bridge ID.

Parameter	Type	Description
bridge_name	String	Bridge name.
status	String	Bridge status. Options: • **ONLINE** • **OFFLINE**

Table 1-774 Page

Parameter	Type	Description
count	Long	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

```
GET https://{endpoint}/v5/iot/{project_id}/bridges
```

Example Responses

Status code: 200

OK

```
{
  "bridges" : [ {
    "bridge_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "bridge_name" : "dianabridge",
    "status" : "ONLINE"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.24.3 Delete a Bridge

Function

This API is used by an application to delete a specific bridge from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/bridges/{bridge_id}

Table 1-775 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
bridge_id	Yes	String	Bridge ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-776 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

```
DELETE https://{endpoint}/v5/iot/{project_id}/bridges/{bridge_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.24.4 Reset the Bridge Secret

Function

This API is used by an application to reset the bridge secret on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/bridges/{bridge_id}/reset-secret

Table 1-777 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
bridge_id	Yes	String	Bridge ID. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Request Parameters

Table 1-778 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.

Parameter	Mandatory	Type	Description
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-779 Request body parameters

Parameter	Mandatory	Type	Description
force_disconnect	No	Boolean	Whether to forcibly disconnect a bridge. Only persistent connections are supported. Default: **false**

Response Parameters

Status code: 200

Table 1-780 Response body parameters

Parameter	Type	Description
bridge_id	String	Bridge ID.
secret	String	Bridge secret.

Example Requests

```
POST https://{endpoint}/v5/iot/{project_id}/bridges/{bridge_id}/reset-secret

{
  "force_disconnect" : false
}
```

Example Responses

Status code: 200

OK


```
{
  "bridge_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "secret" : "3b935a250c50dc2c6d481d048cefdc3c"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25 Device Policy Management

1.4.25.1 Create a Device Policy

Function

This API is used by an application to create a policy on the IoT platform. The policy takes effect only after being bound to a device or product.

- A maximum of 50 device policies can be created for an instance.
- This API is supported only by **standard and enterprise editions**.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-policies

Table 1-781 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-782 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-783 Request body parameters

Parameter	Mandatory	Type	Description
policy_name	Yes	String	**Parameter description**: policy name. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can use this parameter to specify the resource space to which the device to create will belong. If this parameter is not specified, the device to create will belong to the **default resource space**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
statement	Yes	Array of **Statement** objects	**Parameter description**: policy document.

Table 1-784 Statement

Parameter	Mandatory	Type	Description
effect	Yes	String	Specifies whether to allow or reject the operation. If there are both **ALLOW** and **DENY** statements, the **DENY** statement takes precedence. • **ALLOW** • **DENY**
actions	Yes	Array of strings	Specifies the operation allowed or denied by the policy. Format: *Service name:Resource:Operation*. Options: • **iotda:devices:publish**: The device uses MQTT to publish messages. • **iotda:devices:subscribe**: The device uses MQTT to subscribe to messages.

Parameter	Mandatory	Type	Description
resources	Yes	Array of strings	Specifies the resource on which the operation is allowed or rejected. Format: *Resource type:Resource name*. For example, the resource subscribed by the device is **topic:/v1/${devices.deviceId}/test/hello**. **Value**: Length of the resource list: 1 to 10. Only letters, digits, and special characters (/{}$=+#?*:._-) are allowed.

Response Parameters

Status code: 201

Table 1-785 Response body parameters

Parameter	Type	Description
app_id	String	**Parameter description**: resource space ID.
policy_id	String	Policy ID.
policy_name	String	Policy name.
statement	Array of **Statement** objects	Policy documents.
create_time	String	Time when the policy was created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the policy was updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-786 Statement

Parameter	Type	Description
effect	String	Specifies whether to allow or reject the operation. If there are both **ALLOW** and **DENY** statements, the **DENY** statement takes precedence. • **ALLOW** • **DENY**
actions	Array of strings	Specifies the operation allowed or denied by the policy. Format: *Service name:Resource:Operation*. Options: • **iotda:devices:publish**: The device uses MQTT to publish messages. • **iotda:devices:subscribe**: The device uses MQTT to subscribe to messages.
resources	Array of strings	Specifies the resource on which the operation is allowed or rejected. Format: *Resource type:Resource name*. For example, the resource subscribed by the device is **topic:/v1/${devices.deviceId}/test/hello**. **Value**: Length of the resource list: 1 to 10. Only letters, digits, and special characters (/{}$=+#?*:._-) are allowed.

Example Requests

- Creates a device policy, allowing devices to subscribe to and publish messages through specified topics.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies

{
  "policy_name" : "myPolicyAllow",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "statement" : [ {
    "effect" : "ALLOW",
    "actions" : [ "iotda:devices:publish\riotda:devices:subscribe" ],
    "resources" : [ "topic:/v1/${devices.deviceId}/test/allow" ]
  } ]
}
```

- Creates a device policy, forbidding devices to subscribe to and publish messages through specified topics.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies

{
  "policy_name" : "myPolicyDeny",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "statement" : [ {
    "effect" : "DENY",
    "actions" : [ "iotda:devices:publish\riotda:devices:subscribe" ],
    "resources" : [ "topic:/v1/${devices.deviceId}/test/deny" ]
  } ]
}
```


Example Responses

Status code: 201

Created

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "policy_id" : "5c90fa7d3c4e4405e8525079",
  "policy_name" : "testPolicy",
  "statement" : [ {
    "effect" : "ALLOW",
    "actions" : [ "iotda:devices:publish", "iotda:devices:subscribe" ],
    "resources" : [ "topic:/v1/${devices.deviceId}/test/hello", "topic:/v1/${devices.productId}/test/hello" ]
  } ],
  "create_time" : "20230810T070547Z",
  "update_time" : "20230810T070547Z"
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.2 Query the Device Policy List

Function

This API is used by an application to query the policy list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-policies

Table 1-787 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-788 Query Parameters

Parameter	Mandatory	Type	Description
app_id	No	String	**Parameter description**: resource space ID. This parameter is optional. If you have multiple resource spaces, you can specify this parameter to query device policies in the specified resource space. If this parameter is not carried, device policies in all resource spaces are queried. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
policy_name	No	String	**Parameter description**: policy name. **Value**: The value can contain a maximum of 256 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**

Parameter	Mandatory	Type	Description
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-789 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-790 Response body parameters

Parameter	Type	Description
policies	Array of **ListDevicePolicyBase** objects	Policy information list.
page	**Page** object	Pagination information of the query results.

Table 1-791 ListDevicePolicyBase

Parameter	Type	Description
app_id	String	**Parameter description**: resource space ID.
policy_id	String	Policy ID.
policy_name	String	Policy name.
create_time	String	Time when the policy was created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the policy was updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-792 Page

Parameter	Type	Description
count	Long	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries the device policy list of a specified application.

```
GET https://{endpoint}/v5/iot/{project_id}/device-policies?app_id={app_id}
```

Example Responses

Status code: 200

OK

```
{
  "policies" : [ {
    "policy_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
    "policy_name" : "myPolicy",
    "create_time" : "20190303T081011Z",
    "update_time" : "20190303T081011Z"
  } ],
  "page" : {
    "count" : 10,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.3 Delete a Device Policy

Function

This API is used by an application to delete a specified policy from the IoT platform. Note that deleting a policy will unbind all objects bound to the policy.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/device-policies/{policy_id}

Table 1-793 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	Policy ID. **Value**: A string of 24 characters consisting of letters and digits.

Request Parameters

Table 1-794 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a device policy with a specified ID.

```
DELETE https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content

Status Code	Description
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.4 Query Device Policy Details

Function

This API is used by an application to query details about a specified policy ID on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-policies/{policy_id}

Table 1-795 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	Policy ID.

Request Parameters

Table 1-796 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-797 Response body parameters

Parameter	Type	Description
app_id	String	**Parameter description**: resource space ID.
policy_id	String	Policy ID.
policy_name	String	Policy name.
statement	Array of **Statement** objects	Policy documents.

Parameter	Type	Description
create_time	String	Time when the policy was created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the policy was updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-798 Statement

Parameter	Type	Description
effect	String	Specifies whether to allow or reject the operation. If there are both **ALLOW** and **DENY** statements, the **DENY** statement takes precedence. • **ALLOW** • **DENY**
actions	Array of strings	Specifies the operation allowed or denied by the policy. Format: *Service name:Resource:Operation*. Options: • **iotda:devices:publish**: The device uses MQTT to publish messages. • **iotda:devices:subscribe**: The device uses MQTT to subscribe to messages.
resources	Array of strings	Specifies the resource on which the operation is allowed or rejected. Format: *Resource type:Resource name*. For example, the resource subscribed by the device is **topic:/v1/${devices.deviceId}/test/hello**. **Value**: Length of the resource list: 1 to 10. Only letters, digits, and special characters (/{}$=+#?*:._-) are allowed.

Example Requests

Queries details about a device policy with a specified ID.

```
GET https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}
```

Example Responses

Status code: 200

OK

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "policy_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "policy_name" : "myPolicy",
  "statement" : [ {
    "effect" : "ALLOW",
    "actions" : [ "iotda:devices:publish", "iotda:devices:subscribe" ],
    "resources" : [ "topic:/v1/${devices.deviceId}/test/hello", "topic:/v1/${devices.productId}/test/hello" ]
  } ],
  "create_time" : "20230810T070547Z",
  "update_time" : "20230810T070547Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.5 Update Device Policy Information

Function

This API is used by an application to update a policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/device-policies/{policy_id}

Table 1-799 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	Policy ID.

Request Parameters

Table 1-800 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-801 Request body parameters

Parameter	Mandatory	Type	Description
policy_name	No	String	**Parameter description**: policy name. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
statement	No	Array of **Statement** objects	**Parameter description**: policy document.

Table 1-802 Statement

Parameter	Mandatory	Type	Description
effect	Yes	String	Specifies whether to allow or reject the operation. If there are both **ALLOW** and **DENY** statements, the **DENY** statement takes precedence. • **ALLOW** • **DENY**
actions	Yes	Array of strings	Specifies the operation allowed or denied by the policy. Format: *Service name:Resource:Operation*. Options: • **iotda:devices:publish**: The device uses MQTT to publish messages. • **iotda:devices:subscribe**: The device uses MQTT to subscribe to messages.
resources	Yes	Array of strings	Specifies the resource on which the operation is allowed or rejected. Format: *Resource type:Resource name*. For example, the resource subscribed by the device is **topic:/v1/${devices.deviceId}/test/hello**. **Value**: Length of the resource list: 1 to 10. Only letters, digits, and special characters (/{}$=+#?*:._-) are allowed.

Response Parameters

Status code: 200

Table 1-803 Response body parameters

Parameter	Type	Description
app_id	String	**Parameter description**: resource space ID.
policy_id	String	Policy ID.
policy_name	String	Policy name.
statement	Array of **Statement** objects	Policy documents.
create_time	String	Time when the policy was created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the policy was updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-804 Statement

Parameter	Type	Description
effect	String	Specifies whether to allow or reject the operation. If there are both **ALLOW** and **DENY** statements, the **DENY** statement takes precedence. • **ALLOW** • **DENY**
actions	Array of strings	Specifies the operation allowed or denied by the policy. Format: *Service name:Resource:Operation*. Options: • **iotda:devices:publish**: The device uses MQTT to publish messages. • **iotda:devices:subscribe**: The device uses MQTT to subscribe to messages.

Parameter	Type	Description
resources	Array of strings	Specifies the resource on which the operation is allowed or rejected. Format: *Resource type:Resource name*. For example, the resource subscribed by the device is **topic:/v1/${devices.deviceId}/test/hello**. **Value**: Length of the resource list: 1 to 10. Only letters, digits, and special characters (/{}$=+#?*:._-) are allowed.

Example Requests

Updates a device policy.

```
PUT https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}

{
  "policy_name" : "testPolicy",
  "statement" : [ {
    "effect" : "ALLOW",
    "actions" : [ "iotda:devices:publish", "iotda:devices:subscribe" ],
    "resources" : [ "topic:/v1/${devices.deviceId}/test/hello", "topic:/v1/${devices.productId}/test/hello" ]
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "policy_id" : "5c90fa7d3c4e4405e8525079",
  "policy_name" : "testPolicy",
  "statement" : [ {
    "effect" : "ALLOW",
    "actions" : [ "iotda:devices:publish", "iotda:devices:subscribe" ],
    "resources" : [ "topic:/v1/${devices.deviceId}/test/hello", "topic:/v1/${devices.productId}/test/hello" ]
  } ],
  "create_time" : "20230810T070547Z",
  "update_time" : "20230810T070547Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden

Status Code	Description
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.6 Bind a Device Policy

Function

This API is used by an application to bind a policy to devices in batches on the IoT platform. Currently, the target type can be device or product. When the target type is product, the policy takes effect on all devices under the product.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-policies/{policy_id}/bind

Table 1-805 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	Policy ID.

Request Parameters

Table 1-806 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-807 Request body parameters

Parameter	Mandatory	Type	Description
target_type	Yes	String	**Parameter description**: type of the target bound to the policy. **Value**: **device**, **product**, and **app** (the entire resource space).
target_ids	Yes	Array of strings	ID list of targets bound to the policy.

Response Parameters

Status code: 200

Table 1-808 Response body parameters

Parameter	Type	Description
policy_id	String	Policy ID.
target_type	String	**Parameter description**: target type of a policy. **Value**: **device**, **product**, and **app** (the entire resource space).
success_targets	Array of strings	ID list of successful targets.
failure_targets	Array of **DevicePolicyBindOrUnbindFailureDetail** objects	ID list of failed targets.

Table 1-809 DevicePolicyBindOrUnbindFailureDetail

Parameter	Type	Description
target_id	String	ID of the failed target.
error_code	String	Error code.
error_msg	String	Error details.

Example Requests

- Binds a policy to a device.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}/bind

{
  "target_type" : "device",
  "target_ids" : [ "64a7ba7ef9cb063d27e16b97_123456", "64a7ba7ef9cb063d27e16b97_123457" ]
}
```

- Binds a policy to a product.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}/bind

{
  "target_type" : "product",
  "target_ids" : [ "64a7ba7ef9cb063d27e16b97" ]
}
```

Example Responses

Status code: 200

OK

```
{
  "policy_id" : "5c90fa7d3c4e4405e8525079",
  "target_type" : "device",
  "success_targets" : [ "64a7ba7ef9cb063d27e16b97_123456", "64a7ba7ef9cb063d27e16b97_123457" ],
```



```
 "failure_targets" : [ {
   "target_id" : "64a7ba7ef9cb063d27e16b97_123458",
   "error_code" : "IOTDA.015204",
   "error_msg" : "Operation not allowed. The number of policies bound to the target reaches the upper
limit (5)."
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.7 Unbind a Device Policy

Function

This API is used by an application to unbind a target object from a specified policy on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-policies/{policy_id}/unbind

Table 1-810 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Parameter	Mandatory	Type	Description
policy_id	Yes	String	Policy ID.

Request Parameters

Table 1-811 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-812 Request body parameters

Parameter	Mandatory	Type	Description
target_type	Yes	String	**Parameter description**: type of the target unbound from the policy. **Value**: **device**, **product**, and **app** (the entire resource space).
target_ids	Yes	Array of strings	ID list of targets unbound from the policy.

Response Parameters

Status code: 200

Table 1-813 Response body parameters

Parameter	Type	Description
policy_id	String	Policy ID.
target_type	String	**Parameter description**: target type of a policy. **Value**: **device**, **product**, and **app** (the entire resource space).
success_targets	Array of strings	ID list of successful targets.
failure_targets	Array of **DevicePolicyBindOrUnbindFailureDetail** objects	ID list of failed targets.

Table 1-814 DevicePolicyBindOrUnbindFailureDetail

Parameter	Type	Description
target_id	String	ID of the failed target.
error_code	String	Error code.
error_msg	String	Error details.

Example Requests

- Unbinds a policy from a device.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}/unbind

{
  "target_type" : "device",
  "target_ids" : [ "64a7ba7ef9cb063d27e16b97_123456" ]
}
```

- Unbinds a policy from a product.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}/unbind

{
  "target_type" : "product",
  "target_ids" : [ "64a7ba7ef9cb063d27e16b97" ]
}
```

Example Responses

Status code: 200

OK


```
{
  "policy_id" : "5c90fa7d3c4e4405e8525079",
  "target_type" : "device",
  "success_targets" : [ "64a7ba7ef9cb063d27e16b97_123456", "64a7ba7ef9cb063d27e16b97_123457" ],
  "failure_targets" : [ {
    "target_id" : "64a7ba7ef9cb063d27e16b97_123458",
    "error_code" : "IOTDA.015207",
    "error_msg" : "Invalid input. The target does not bind the device-policy."
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.25.8 Query the List of the Targets Bound to a Device Policy

Function

This API is used by an application to query the list of the targets bound to a specified policy ID on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-policies/{policy_id}/list-targets

Table 1-815 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
policy_id	Yes	String	Policy ID.

Request Parameters

Table 1-816 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-817 Request body parameters

Parameter	Mandatory	Type	Description
target_type	No	String	**Parameter description**: type of the target bound to the policy. **Value**: **device**, **product**, and **app** (the entire resource space).
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Response Parameters

Status code: 200

Table 1-818 Response body parameters

Parameter	Type	Description
targets	Array of **PolicyTargetBase** objects	List of policy binding information.
page	**Page** object	Pagination information of the query results.

Table 1-819 PolicyTargetBase

Parameter	Type	Description
target_type	String	**Parameter description**: type of the target bound to the policy. **Value**: **device**, **product**, and **app** (the entire resource space).
target_id	String	ID of the target bound to the policy.

Table 1-820 Page

Parameter	Type	Description
count	Long	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries the list of the targets bound to the policy with a specified ID.

```
POST https://{endpoint}/v5/iot/{project_id}/device-policies/{policy_id}/list-targets

{
  "target_type" : "device"
}
```

Example Responses

Status code: 200

OK

```
{
  "targets" : [ {
    "target_type" : "device",
    "target_id" : "64a7ba7ef9cb063d27e16b97_123456"
  } ],
  "page" : {
    "marker" : "5c90fa7d3c4e4405e8525079",
    "count" : 1
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request

Status Code	Description
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.26 Pre-provisioning Template Management

1.4.26.1 Create a Pre-provisioning Template

Function

This API is used by an application to create a pre-provisioning template on the IoT platform. For a device not registered, you can use a pre-provisioning template to automatically register the device information on the IoT platform when the device connects to the platform for the first time.

- A pre-provisioning template takes effect only after it is bound to at least one device CA certificate.
- An instance can have a maximum of 10 pre-provisioning templates.
- This API is supported only by standard and enterprise editions.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/provisioning-templates

Table 1-821 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-822 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-823 Request body parameters

Parameter	Mandatory	Type	Description
template_name	Yes	String	**Parameter description**: name of the pre-provisioning template. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: description of the pre-provisioning template. **Value**: The value can contain a maximum of 2,048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**
template_body	Yes	**ProvisioningTemplateBody** object	**Parameter description**: detailed content of the pre-provisioning template in JSON format.

Table 1-824 ProvisioningTemplateBody

Parameter	Mandatory	Type	Description
parameters	Yes	Object	**Parameter description**: pre-provisioning template parameters, which can be extracted from the user fields of pre-provisioning device certificates. The configuration format is {"*parameter*": {"type":"String"}}. Huawei Cloud IoT platform defines the parameters that can be declared and referenced in pre-provisioning templates. A device certificate must contain the parameters referenced in the template. • **iotda::certificate::country**: country/region, C • **iotda::certificate::organization**: organization, O • **iotda::certificate::organizational_unit**: organizational unit, OU • **iotda::certificate::distinguished_name_qualifier**: distinguishable name qualifier, dnQualifier • **iotda::certificate::state_name**: province, ST • **iotda::certificate::common_name**: common name, CN • **iotda::certificate::serial_number**: serial number, serialNumber **type** indicates the parameter type. Currently, only **String** is supported. Configuration example: '{"iotda::certificate::country": {"type":"String"}, "iotda::certificate::organization":{"type":"String"}, "iotda::certificate::organizational_unit":{"type":"String"},

Parameter	Mandatory	Type	Description
			"iotda::certificate::distinguished_name_qualifier": {"type":"String"}, "iotda::certificate::state_name":{"type":"String"}, "iotda::certificate::common_name":{"type":"String"}, "iotda::certificate::serial_number":{"type":"String"}}'
resources	Yes	**TemplateResource** object	Structure of device resources of a pre-provisioning template.

Table 1-825 TemplateResource

Parameter	Mandatory	Type	Description
device	Yes	**DeviceResource** object	Structure of device resource details of a pre-provisioning template.
policy	No	**PolicyResource** object	Structure of device policy resource details of a pre-provisioning template.

Table 1-826 DeviceResource

Parameter	Mandatory	Type	Description
device_name	No	**ParameterRef** object	Device name.
node_id	Yes	**ParameterRef** object	Node ID.

Parameter	Mandatory	Type	Description
product_id	Yes	Object	**Parameter description**: ID of the product to which the device belongs. The value can be a static string ID or dynamic parameters referenced from the template. • Static string: **642bf260f2f9030e44210d8d**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. " • Parameter reference: {"ref": "iotda::certificate::country"}
tags	No	Array of **TagRef** objects	**Parameter description**: list of tags bound to a device.

Table 1-827 ParameterRef

Parameter	Mandatory	Type	Description
ref	Yes	String	Parameter reference name.

Table 1-828 TagRef

Parameter	Mandatory	Type	Description
tag_key	No	Object	**Parameter description**: tag key name. The value can be a static string or dynamic parameters referenced from the template. • Static string: **myTagKey**. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}

Parameter	Mandatory	Type	Description
tag_value	No	Object	**Parameter description**: tag value. The value can be a static string or dynamic parameters referenced from the template. • Static string: "myTagValue". **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}

Table 1-829 PolicyResource

Parameter	Mandatory	Type	Description
policy_ids	No	Array of strings	**Parameter description**: ID list of policies to be bound to a device.

Response Parameters

Status code: 201

Table 1-830 Response body parameters

Parameter	Type	Description
template_id	String	**Parameter description**: ID of the pre-provisioning template.
template_name	String	**Parameter description**: name of the pre-provisioning template. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
description	String	**Parameter description**: description of the pre-provisioning template. **Value**: The value can contain a maximum of 2,048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**

Parameter	Type	Description
template_body	ProvisioningTemplateBody object	**Parameter description**: detailed content of the pre-provisioning template in JSON format.
create_time	String	Time when a pre-provisioning template is created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the pre-provisioning template is updated in the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-831 ProvisioningTemplateBody

Parameter	Type	Description
parameters	Object	**Parameter description**: pre-provisioning template parameters, which can be extracted from the user fields of pre-provisioning device certificates. The configuration format is {"*parameter*":{"type":"String"}}. Huawei Cloud IoT platform defines the parameters that can be declared and referenced in pre-provisioning templates. A device certificate must contain the parameters referenced in the template. • **iotda::certificate::country**: country/region, C • **iotda::certificate::organization**: organization, O • **iotda::certificate::organizational_unit**: organizational unit, OU • **iotda::certificate::distinguished_name_qualifier**: distinguishable name qualifier, dnQualifier • **iotda::certificate::state_name**: province, ST • **iotda::certificate::common_name**: common name, CN • **iotda::certificate::serial_number**: serial number, serialNumber **type** indicates the parameter type. Currently, only **String** is supported. Configuration example: '{"iotda::certificate::country":{"type":"String"}, "iotda::certificate::organization": {"type":"String"}, "iotda::certificate::organizational_unit": {"type":"String"}, "iotda::certificate::distinguished_name_qualifier":{"type":"String"}, "iotda::certificate::state_name": {"type":"String"}, "iotda::certificate::common_name": {"type":"String"}, "iotda::certificate::serial_number": {"type":"String"}}'
resources	**TemplateResource** object	Structure of device resources of a pre-provisioning template.

Table 1-832 TemplateResource

Parameter	Type	Description
device	**DeviceResource** object	Structure of device resource details of a pre-provisioning template.
policy	**PolicyResource** object	Structure of device policy resource details of a pre-provisioning template.

Table 1-833 DeviceResource

Parameter	Type	Description
device_name	**ParameterRef** object	Device name.
node_id	**ParameterRef** object	Node ID.
product_id	Object	**Parameter description**: ID of the product to which the device belongs. The value can be a static string ID or dynamic parameters referenced from the template. • Static string: **642bf260f2f9030e44210d8d**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. " • Parameter reference: {"ref": "iotda::certificate::country"}
tags	Array of **TagRef** objects	**Parameter description**: list of tags bound to a device.

Table 1-834 ParameterRef

Parameter	Type	Description
ref	String	Parameter reference name.

Table 1-835 TagRef

Parameter	Type	Description
tag_key	Object	**Parameter description**: tag key name. The value can be a static string or dynamic parameters referenced from the template. ● Static string: **myTagKey**. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. ● Parameter reference: {"ref": "iotda::certificate::country"}
tag_value	Object	**Parameter description**: tag value. The value can be a static string or dynamic parameters referenced from the template. ● Static string: "myTagValue". **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. ● Parameter reference: {"ref": "iotda::certificate::country"}

Table 1-836 PolicyResource

Parameter	Type	Description
policy_ids	Array of strings	**Parameter description**: ID list of policies to be bound to a device.

Example Requests

- Creates a pre-provisioning template with parameter reference.

```
POST https://{endpoint}/v5/iot/{project_id}/provisioning-templates

{
  "template_name" : "myTemplate",
  "description" : "myTemplate",
  "template_body" : {
    "parameters" : {
      "iotda::certificate::country" : {
        "type" : "String"
      },
      "iotda::certificate::organization" : {
        "type" : "String"
      },
      "iotda::certificate::organizational_unit" : {
        "type" : "String"
      },
      "iotda::certificate::distinguished_name_qualifier" : {
        "type" : "String"
      },
      "iotda::certificate::state_name" : {
```



```
      "type" : "String"
    },
    "iotda::certificate::common_name" : {
      "type" : "String"
    },
    "iotda::certificate::serial_number" : {
      "type" : "String"
    }
  },
  "resources" : {
    "device" : {
      "device_name" : {
        "ref" : "iotda::certificate::organization"
      },
      "node_id" : {
        "ref" : "iotda::certificate::common_name"
      },
      "product_id" : {
        "ref" : "iotda::certificate::organization"
      },
      "tags" : [ {
        "tag_key" : {
          "ref" : "iotda::certificate::organization"
        },
        "tag_value" : {
          "ref" : "iotda::certificate::organizational_unit"
        }
      } ]
    },
    "policy" : {
      "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
    }
  }
 }
}
```

- Creates a pre-provisioning template some custom parameters.

```
POST https://{endpoint}/v5/iot/{project_id}/provisioning-templates

{
  "template_name" : "myTemplate2",
  "description" : "myTemplate2",
  "template_body" : {
    "parameters" : {
      "iotda::certificate::country" : {
        "type" : "String"
      },
      "iotda::certificate::organization" : {
        "type" : "String"
      },
      "iotda::certificate::organizational_unit" : {
        "type" : "String"
      },
      "iotda::certificate::distinguished_name_qualifier" : {
        "type" : "String"
      },
      "iotda::certificate::state_name" : {
        "type" : "String"
      },
      "iotda::certificate::common_name" : {
        "type" : "String"
      },
      "iotda::certificate::serial_number" : {
        "type" : "String"
      }
    },
    "resources" : {
      "device" : {
        "device_name" : {
          "ref" : "iotda::certificate::organization"
```

```
      },
      "node_id" : {
        "ref" : "iotda::certificate::common_name"
      },
      "product_id" : "642bf260f2f9030e44210d8d",
      "tags" : [ {
        "tag_key" : "myTagKey",
        "tag_value" : "myTagValue"
      } ]
    },
    "policy" : {
      "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
    }
  }
 }
}
```

Example Responses

Status code: 201

Created

```
{
 "template_id" : "5c90fa7d3c4e4405e8525079",
 "template_name" : "myTemplate",
 "description" : "myTemplate",
 "template_body" : {
   "parameters" : {
     "iotda::certificate::country" : {
       "type" : "String"
     },
     "iotda::certificate::organization" : {
       "type" : "String"
     },
     "iotda::certificate::organizational_unit" : {
       "type" : "String"
     },
     "iotda::certificate::distinguished_name_qualifier" : {
       "type" : "String"
     },
     "iotda::certificate::state_name" : {
       "type" : "String"
     },
     "iotda::certificate::common_name" : {
       "type" : "String"
     },
     "iotda::certificate::serial_number" : {
       "type" : "String"
     }
   },
   "resources" : {
     "device" : {
       "device_name" : {
         "ref" : "iotda::certificate::organization"
       },
       "node_id" : {
         "ref" : "iotda::certificate::common_name"
       },
       "product_id" : {
         "ref" : "iotda::certificate::organization"
       }
     },
     "policy" : {
       "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
     }
   }
 },
 "create_time" : "20230810T070547Z",
```



```
  "update_time" : "20230810T070547Z"
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.26.2 Query the Pre-provisioning Template List

Function

This API is used by an application to query a pre-provisioning template list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/provisioning-templates

Table 1-837 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-838 Query Parameters

Parameter	Mandatory	Type	Description
template_name	No	String	**Parameter description**: name of the pre-provisioning template. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-839 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-840 Response body parameters

Parameter	Type	Description
templates	Array of **ProvisioningTemplateSimple** objects	**Parameter description**: pre-provisioning template details.
page	**Page** object	Pagination information of the query results.

Table 1-841 ProvisioningTemplateSimple

Parameter	Type	Description
template_id	String	**Parameter description**: ID of the pre-provisioning template.
template_name	String	**Parameter description**: name of the pre-provisioning template. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
description	String	**Parameter description**: description of the pre-provisioning template. **Value**: The value can contain a maximum of 2,048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**
create_time	String	Time when a pre-provisioning template is created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the pre-provisioning template is updated in the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-842 Page

Parameter	Type	Description
count	Long	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries the pre-provisioning template list of a tenant.

```
GET https://{endpoint}/v5/iot/{project_id}/provisioning-templates?template_name={template_name}
```

Example Responses

Status code: 200

OK

```
{
  "templates" : [ {
```



```
  "template_id" : "5c90fa7d3c4e4405e8525079",
  "template_name" : "myTemplate",
  "description" : "myTemplate",
  "create_time" : "20230810T070547Z",
  "update_time" : "20230810T070547Z"
 } ],
 "page" : {
  "count" : 1,
  "marker" : "5c90fa7d3c4e4405e8525079"
 }
}
```

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.26.3 Delete a Pre-provisioning Template

Function

This API is used by an application to delete a specified pre-provisioning template from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/provisioning-templates/{template_id}

Table 1-843 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
template_id	Yes	String	Pre-provisioning template ID.

Request Parameters

Table 1-844 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a pre-provisioning template with a specified ID.

```
DELETE https://{endpoint}/v5/iot/{project_id}/provisioning-templates/{template_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content

Status Code	Description
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.26.4 Query Pre-provisioning Template Details

Function

This API is used by an application to query details about a pre-provisioning template with a specified ID on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/provisioning-templates/{template_id}

Table 1-845 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
template_id	Yes	String	Pre-provisioning template ID.

Request Parameters

Table 1-846 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-847 Response body parameters

Parameter	Type	Description
template_id	String	**Parameter description**: ID of the pre-provisioning template.
template_name	String	**Parameter description**: name of the pre-provisioning template. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
description	String	**Parameter description**: description of the pre-provisioning template. **Value**: The value can contain a maximum of 2,048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**
template_body	**ProvisioningTemplateBody** object	**Parameter description**: detailed content of the pre-provisioning template in JSON format.
create_time	String	Time when a pre-provisioning template is created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the pre-provisioning template is updated in the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-848 ProvisioningTemplateBody

Parameter	Type	Description
parameters	Object	**Parameter description**: pre-provisioning template parameters, which can be extracted from the user fields of pre-provisioning device certificates. The configuration format is {"*parameter*":{"type":"String"}}. Huawei Cloud IoT platform defines the parameters that can be declared and referenced in pre-provisioning templates. A device certificate must contain the parameters referenced in the template. • **iotda::certificate::country**: country/region, C • **iotda::certificate::organization**: organization, O • **iotda::certificate::organizational_unit**: organizational unit, OU • **iotda::certificate::distinguished_name_qualifier**: distinguishable name qualifier, dnQualifier • **iotda::certificate::state_name**: province, ST • **iotda::certificate::common_name**: common name, CN • **iotda::certificate::serial_number**: serial number, serialNumber **type** indicates the parameter type. Currently, only **String** is supported. Configuration example: '{"iotda::certificate::country":{"type":"String"}, "iotda::certificate::organization": {"type":"String"}, "iotda::certificate::organizational_unit": {"type":"String"}, "iotda::certificate::distinguished_name_qualifier":{"type":"String"}, "iotda::certificate::state_name": {"type":"String"}, "iotda::certificate::common_name": {"type":"String"}, "iotda::certificate::serial_number": {"type":"String"}}'
resources	**TemplateResource** object	Structure of device resources of a pre-provisioning template.

Table 1-849 TemplateResource

Parameter	Type	Description
device	**DeviceResource** object	Structure of device resource details of a pre-provisioning template.
policy	**PolicyResource** object	Structure of device policy resource details of a pre-provisioning template.

Table 1-850 DeviceResource

Parameter	Type	Description
device_name	**ParameterRef** object	Device name.
node_id	**ParameterRef** object	Node ID.
product_id	Object	**Parameter description**: ID of the product to which the device belongs. The value can be a static string ID or dynamic parameters referenced from the template. • Static string: **642bf260f2f9030e44210d8d**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. " • Parameter reference: {"ref": "iotda::certificate::country"}
tags	Array of **TagRef** objects	**Parameter description**: list of tags bound to a device.

Table 1-851 ParameterRef

Parameter	Type	Description
ref	String	Parameter reference name.

Table 1-852 TagRef

Parameter	Type	Description
tag_key	Object	**Parameter description**: tag key name. The value can be a static string or dynamic parameters referenced from the template. • Static string: **myTagKey**. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}
tag_value	Object	**Parameter description**: tag value. The value can be a static string or dynamic parameters referenced from the template. • Static string: "myTagValue". **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}

Table 1-853 PolicyResource

Parameter	Type	Description
policy_ids	Array of strings	**Parameter description**: ID list of policies to be bound to a device.

Example Requests

Queries details about a pre-provisioning template with a specified ID.

```
GET https://{endpoint}/v5/iot/{project_id}/provisioning-templates/{template_id}
```

Example Responses

Status code: 200

OK

```
{
  "template_id" : "5c90fa7d3c4e4405e8525079",
  "template_name" : "myTemplate",
  "description" : "myTemplate",
  "template_body" : {
    "parameters" : {
      "iotda::certificate::country" : {
        "type" : "String"
      },
      "iotda::certificate::organization" : {
```



```
      "type" : "String"
    },
    "iotda::certificate::organizational_unit" : {
      "type" : "String"
    },
    "iotda::certificate::distinguished_name_qualifier" : {
      "type" : "String"
    },
    "iotda::certificate::state_name" : {
      "type" : "String"
    },
    "iotda::certificate::common_name" : {
      "type" : "String"
    },
    "iotda::certificate::serial_number" : {
      "type" : "String"
    }
  },
  "resources" : {
    "device" : {
      "device_name" : {
        "ref" : "iotda::certificate::organization"
      },
      "node_id" : {
        "ref" : "iotda::certificate::common_name"
      },
      "product_id" : {
        "ref" : "iotda::certificate::organization"
      }
    },
    "policy" : {
      "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
    }
  }
},
"create_time" : "20230810T070547Z",
"update_time" : "20230810T070547Z"
}
```


Status Codes

Status Code	Description
200	OK
404	Not Found

Error Codes

See **Error Codes**.

1.4.26.5 Update Information About a Pre-provisioning Template with a Specified ID

Function

This API is used by an application to update a pre-provisioning template with a specified ID on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/provisioning-templates/{template_id}

Table 1-854 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
template_id	Yes	String	Pre-provisioning Template ID.

Request Parameters

Table 1-855 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-856 Request body parameters

Parameter	Mandatory	Type	Description
description	No	String	**Parameter description**: description of the pre-provisioning template. **Value**: The value can contain a maximum of 2,048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**
template_body	No	**ProvisioningTemplateBody** object	**Parameter description**: detailed content of the pre-provisioning template in JSON format.

Table 1-857 ProvisioningTemplateBody

Parameter	Mandatory	Type	Description
parameters	Yes	Object	**Parameter description**: pre-provisioning template parameters, which can be extracted from the user fields of pre-provisioning device certificates. The configuration format is {"*parameter*": {"type":"String"}}. Huawei Cloud IoT platform defines the parameters that can be declared and referenced in pre-provisioning templates. A device certificate must contain the parameters referenced in the template. • **iotda::certificate::country**: country/region, C • **iotda::certificate::organization**: organization, O • **iotda::certificate::organizational_unit**: organizational unit, OU • **iotda::certificate::distinguished_name_qualifier**: distinguishable name qualifier, dnQualifier • **iotda::certificate::state_name**: province, ST • **iotda::certificate::common_name**: common name, CN • **iotda::certificate::serial_number**: serial number, serialNumber **type** indicates the parameter type. Currently, only **String** is supported. Configuration example: '{"iotda::certificate::country": {"type":"String"}, "iotda::certificate::organization":{"type":"String"}, "iotda::certificate::organizational_unit":{"type":"String"},

Parameter	Mandatory	Type	Description
			"iotda::certificate::distinguished_name_qualifier": {"type":"String"}, "iotda::certificate::state_name":{"type":"String"}, "iotda::certificate::common_name":{"type":"String"}, "iotda::certificate::serial_number":{"type":"String"}}'
resources	Yes	**TemplateResource** object	Structure of device resources of a pre-provisioning template.

Table 1-858 TemplateResource

Parameter	Mandatory	Type	Description
device	Yes	**DeviceResource** object	Structure of device resource details of a pre-provisioning template.
policy	No	**PolicyResource** object	Structure of device policy resource details of a pre-provisioning template.

Table 1-859 DeviceResource

Parameter	Mandatory	Type	Description
device_name	No	**ParameterRef** object	Device name.
node_id	Yes	**ParameterRef** object	Node ID.

Parameter	Mandatory	Type	Description
product_id	Yes	Object	**Parameter description**: ID of the product to which the device belongs. The value can be a static string ID or dynamic parameters referenced from the template. • Static string: **642bf260f2f9030e44210d8d**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. " • Parameter reference: {"ref": "iotda::certificate::country"}
tags	No	Array of **TagRef** objects	**Parameter description**: list of tags bound to a device.

Table 1-860 ParameterRef

Parameter	Mandatory	Type	Description
ref	Yes	String	Parameter reference name.

Table 1-861 TagRef

Parameter	Mandatory	Type	Description
tag_key	No	Object	**Parameter description**: tag key name. The value can be a static string or dynamic parameters referenced from the template. • Static string: **myTagKey**. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}

Parameter	Mandatory	Type	Description
tag_value	No	Object	**Parameter description**: tag value. The value can be a static string or dynamic parameters referenced from the template. • Static string: "myTagValue". **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}

Table 1-862 PolicyResource

Parameter	Mandatory	Type	Description
policy_ids	No	Array of strings	**Parameter description**: ID list of policies to be bound to a device.

Response Parameters

Status code: 200

Table 1-863 Response body parameters

Parameter	Type	Description
template_id	String	**Parameter description**: ID of the pre-provisioning template.
template_name	String	**Parameter description**: name of the pre-provisioning template. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
description	String	**Parameter description**: description of the pre-provisioning template. **Value**: The value can contain a maximum of 2,048 characters. Only letters, digits, and special characters (_?'#().,&%@!-) are allowed. Maximum: **2048**

Parameter	Type	Description
template_body	ProvisioningTemplateBody object	**Parameter description**: detailed content of the pre-provisioning template in JSON format.
create_time	String	Time when a pre-provisioning template is created on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the pre-provisioning template is updated in the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-864 ProvisioningTemplateBody

Parameter	Type	Description
parameters	Object	**Parameter description**: pre-provisioning template parameters, which can be extracted from the user fields of pre-provisioning device certificates. The configuration format is {"*parameter*":{"type":"String"}}. Huawei Cloud IoT platform defines the parameters that can be declared and referenced in pre-provisioning templates. A device certificate must contain the parameters referenced in the template. • **iotda::certificate::country**: country/region, C • **iotda::certificate::organization**: organization, O • **iotda::certificate::organizational_unit**: organizational unit, OU • **iotda::certificate::distinguished_name_qualifier**: distinguishable name qualifier, dnQualifier • **iotda::certificate::state_name**: province, ST • **iotda::certificate::common_name**: common name, CN • **iotda::certificate::serial_number**: serial number, serialNumber **type** indicates the parameter type. Currently, only **String** is supported. Configuration example: '{"iotda::certificate::country":{"type":"String"}, "iotda::certificate::organization": {"type":"String"}, "iotda::certificate::organizational_unit": {"type":"String"}, "iotda::certificate::distinguished_name_qualifier":{"type":"String"}, "iotda::certificate::state_name": {"type":"String"}, "iotda::certificate::common_name": {"type":"String"}, "iotda::certificate::serial_number": {"type":"String"}}'
resources	**TemplateResource** object	Structure of device resources of a pre-provisioning template.

Table 1-865 TemplateResource

Parameter	Type	Description
device	**DeviceResource** object	Structure of device resource details of a pre-provisioning template.
policy	**PolicyResource** object	Structure of device policy resource details of a pre-provisioning template.

Table 1-866 DeviceResource

Parameter	Type	Description
device_name	**ParameterRef** object	Device name.
node_id	**ParameterRef** object	Node ID.
product_id	Object	**Parameter description**: ID of the product to which the device belongs. The value can be a static string ID or dynamic parameters referenced from the template. • Static string: **642bf260f2f9030e44210d8d**. **Value**: The value can contain a maximum of 36 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. " • Parameter reference: {"ref": "iotda::certificate::country"}
tags	Array of **TagRef** objects	**Parameter description**: list of tags bound to a device.

Table 1-867 ParameterRef

Parameter	Type	Description
ref	String	Parameter reference name.

Table 1-868 TagRef

Parameter	Type	Description
tag_key	Object	**Parameter description**: tag key name. The value can be a static string or dynamic parameters referenced from the template. • Static string: **myTagKey**. **Value**: The value can contain a maximum of 64 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}
tag_value	Object	**Parameter description**: tag value. The value can be a static string or dynamic parameters referenced from the template. • Static string: "myTagValue". **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), periods (.), and hyphens (-) are allowed. • Parameter reference: {"ref": "iotda::certificate::country"}

Table 1-869 PolicyResource

Parameter	Type	Description
policy_ids	Array of strings	**Parameter description**: ID list of policies to be bound to a device.

Example Requests

- Updates a pre-provisioning template.

```
PUT https://{endpoint}/v5/iot/{project_id}/provisioning-templates/{template_id}

{
  "description" : "myTemplate",
  "template_body" : {
    "parameters" : {
      "iotda::certificate::country" : {
        "type" : "String"
      },
      "iotda::certificate::organization" : {
        "type" : "String"
      },
      "iotda::certificate::organizational_unit" : {
        "type" : "String"
      },
      "iotda::certificate::distinguished_name_qualifier" : {
        "type" : "String"
      },
      "iotda::certificate::state_name" : {
        "type" : "String"
```



```
    },
    "iotda::certificate::common_name" : {
      "type" : "String"
    },
    "iotda::certificate::serial_number" : {
      "type" : "String"
    }
  },
  "resources" : {
    "device" : {
      "device_name" : {
        "ref" : "iotda::certificate::organization"
      },
      "node_id" : {
        "ref" : "iotda::certificate::common_name"
      },
      "product_id" : {
        "ref" : "iotda::certificate::organization"
      },
      "tags" : [ {
        "tag_key" : {
          "ref" : "iotda::certificate::organization"
        },
        "tag_value" : {
          "ref" : "iotda::certificate::organizational_unit"
        }
      } ]
    },
    "policy" : {
      "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
    }
  }
 }
}
```

- Updates a pre-provisioning template with some custom parameters.

```
PUT https://{endpoint}/v5/iot/{project_id}/provisioning-templates/{template_id}

{
 "description" : "myTemplate2",
 "template_body" : {
  "parameters" : {
    "iotda::certificate::country" : {
      "type" : "String"
    },
    "iotda::certificate::organization" : {
      "type" : "String"
    },
    "iotda::certificate::organizational_unit" : {
      "type" : "String"
    },
    "iotda::certificate::distinguished_name_qualifier" : {
      "type" : "String"
    },
    "iotda::certificate::state_name" : {
      "type" : "String"
    },
    "iotda::certificate::common_name" : {
      "type" : "String"
    },
    "iotda::certificate::serial_number" : {
      "type" : "String"
    }
  },
  "resources" : {
    "device" : {
      "device_name" : {
        "ref" : "iotda::certificate::organization"
      },
      "node_id" : {
```

```
        "ref" : "iotda::certificate::common_name"
       },
       "product_id" : "642bf260f2f9030e44210d8d",
       "tags" : [ {
        "tag_key" : "myTagKey",
        "tag_value" : "myTagValue"
       } ]
      },
      "policy" : {
       "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
      }
     }
    }
}
```

Example Responses

Status code: 200

OK

```
{
 "template_id" : "5c90fa7d3c4e4405e8525079",
 "template_name" : "myTemplate",
 "description" : "myTemplate",
 "template_body" : {
  "parameters" : {
   "iotda::certificate::country" : {
    "type" : "String"
   },
   "iotda::certificate::organization" : {
    "type" : "String"
   },
   "iotda::certificate::organizational_unit" : {
    "type" : "String"
   },
   "iotda::certificate::distinguished_name_qualifier" : {
    "type" : "String"
   },
   "iotda::certificate::state_name" : {
    "type" : "String"
   },
   "iotda::certificate::common_name" : {
    "type" : "String"
   },
   "iotda::certificate::serial_number" : {
    "type" : "String"
   }
  },
  "resources" : {
   "device" : {
    "device_name" : {
     "ref" : "iotda::certificate::organization"
    },
    "node_id" : {
     "ref" : "iotda::certificate::common_name"
    },
    "product_id" : {
     "ref" : "iotda::certificate::organization"
    }
   },
   "policy" : {
    "policy_ids" : [ "5c90fa7d3c4e4405e8525079" ]
   }
  }
 },
 "create_time" : "20230810T070547Z",
 "update_time" : "20230810T070547Z"
}
```


Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.27 Custom Authentication

1.4.27.1 Create a Custom Authenticator

Function

This API is used by an application to create a custom authenticator on the IoT platform. You can use function services to customize the logic to authenticate devices connected to the platform.

- A maximum of 10 custom authenticators can be configured for a single instance.
- This API is supported only by standard and enterprise editions.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

POST /v5/iot/{project_id}/device-authorizers

Table 1-870 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Parameters

Table 1-871 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-872 Request body parameters

Parameter	Mandatory	Type	Description
authorizer_name	Yes	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
func_urn	Yes	String	**Parameter description**: function uniform resource name (URN), which uniquely identifies the function. It is the address of the processing function corresponding to the custom authenticator.

Parameter	Mandatory	Type	Description
signing_enable	No	Boolean	**Parameter description**: whether to enable signature authentication (enabled by default). You are advised to enable this function. If this function is enabled, authentication information that does not meet signature requirements will be rejected to reduce invalid function calls, and **signing_token** and **signing_public_key** are mandatory. Default: **true**
signing_token	No	String	**Parameter description**: key value for signature authentication. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
signing_public_key	No	String	**Parameter description**: public secret for signature authentication. Used to check whether the signature information carried by the device is correct.
default_authorizer	No	Boolean	**Parameter description**: whether the current custom authenticator is the default one. The default value is **false**. If this parameter is set to**true**, the current authenticator policy is used for authentication on all devices that support SNI unless otherwise specified. Default: **false**
status	No	String	**Parameter description**: whether to enable the authentication mode. • **ACTIVE**: The authentication is enabled. • **INACTIVE**: The authentication is disabled. Default: **INACTIVE**

Parameter	Mandatory	Type	Description
cache_enable	No	Boolean	**Parameter description**: whether to enable the cache function. The default value is **false**. If this parameter is set to **true** and the device input parameters (username, client ID, password, certificate information, and function URN) remain unchanged, the cache result is directly used when the cache result exists. You are advised to set this parameter to **false** during debugging, set this parameter to **true** during production to avoid frequent function invoking. Default: **false**

Response Parameters

Status code: 201

Table 1-873 Response body parameters

Parameter	Type	Description
authorizer_id	String	**Parameter description**: custom authenticator ID.
authorizer_name	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**
func_urn	String	**Parameter description**: function uniform resource name (URN), which uniquely identifies the function. It is the address of the processing function corresponding to the custom authenticator. Minimum: **0** Maximum: **65535**

Parameter	Type	Description
signing_enable	Boolean	**Parameter description**: whether to enable signature authentication (enabled by default). You are advised to enable this function. If this function is enabled, authentication information that does not meet signature requirements will be rejected to reduce invalid function calls. Default: **true**
signing_token	String	**Parameter description**: key value for signature authentication. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
signing_public_key	String	**Parameter description**: public secret for signature authentication. Used to check whether the signature information carried by the device is correct. Minimum: **0** Maximum: **65535**
default_authorizer	Boolean	**Parameter description**: whether the authentication mode is used by default. The default value is **false**. Default: **false**
status	String	**Parameter description**: whether to enable the authentication mode. ● **ACTIVE**: The authentication is enabled. ● **INACTIVE**: The authentication is disabled. Default: **INACTIVE**
cache_enable	Boolean	**Parameter description**: whether to enable the cache function. The default value is **false**. If this parameter is set to **true** and the device input parameters (username, client ID, password, certificate information, and function URN) remain unchanged, the cache result is directly used when the cache result exists. You are advised to set this parameter to **false** during debugging, set this parameter to **true** during production to avoid frequent function invoking. Default: **false**
create_time	String	Time when operations on custom authenticator are performed on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Type	Description
update_time	String	Time when the custom authenticator is updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Creates a custom authenticator.

```
POST https://{endpoint}/v5/iot/{project_id}/device-authorizers

{
  "authorizer_name" : "myTest",
  "func_urn" : "urn:fss:cn-north-5:d92d9c5eb8e347b5bb31ecfe5bc0c4e1:function:default:mqtt_auth:latest",
  "signing_enable" : true,
  "signing_token" : "string",
  "signing_public_key" : "string",
  "default_authorizer" : false,
  "status" : "ACTIVE",
  "cache_enable" : true
}
```

Example Responses

Status code: 201

Created

```
{
  "authorizer_id" : "5c90fa7d3c4e4405e8525079",
  "authorizer_name" : "myTest",
  "func_name" : "mqtt_auth",
  "func_urn" : "urn:fss:cn-north-5:d92d9c5eb8e347b5bb31ecfe5bc0c4e1:function:default:mqtt_auth:latest",
  "signing_enable" : true,
  "signing_token" : "string",
  "signing_public_key" : "string",
  "default_authorizer" : false,
  "status" : "ACTIVE",
  "cache_enable" : false,
  "create_time" : "20231031T070547Z",
  "update_time" : "20231031T070547Z"
}
```

Status Codes

Status Code	Description
201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.27.2 Query the Custom Authenticator List

Function

This API is used by an application to query the custom authenticator list on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-authorizers

Table 1-874 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 1-875 Query Parameters

Parameter	Mandatory	Type	Description
authorizer_name	No	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Mandatory	Type	Description
limit	No	Integer	**Parameter description**: number of records to display on each page. **Value**: The value is an integer ranging from 1 to 50. The default value is **10**. Minimum: **1** Maximum: **50** Default: **10**
marker	No	String	**Parameter description**: ID of the last record in the previous query. The value is returned by the platform during the previous query. Records are queried in descending order of record IDs (the **marker** value). A newer record will have a larger ID. If **marker** is specified, only the records whose IDs are smaller than **marker** are queried. If **marker** is not specified, the query starts from the record with the largest ID, that is, the latest record. If all data needs to be queried in sequence, this parameter must be filled with the value of **marker** returned in the last query response each time. **Value**: a string of 24 hexadecimal characters. The default value is **ffffffffffffffffffffffff**. Default: **ffffffffffffffffffffffff**

Parameter	Mandatory	Type	Description
offset	No	Integer	**Parameter description**: If **offset** is set to *N*, the query starts from the *N+1* record after the last record in the previous query. The value is an integer ranging from 0 to 500. The default value is **0**. If **offset** is set to **0**, the output starts from the first record after the last record in the previous query. To ensure API performance, you can use this parameter together with **marker** to turn pages. For example, if there are 50 records on each page, you can directly specify **offset** to jump to the specified page within page 1 and 11. If you want to view records displayed on pages 12 to 22, you need to use the **marker** value returned on page 11 as the **marker** value for the next query. **Value**: The value is an integer ranging from 0 to 500. The default value is **0**. Minimum: **0** Maximum: **500** Default: **0**

Request Parameters

Table 1-876 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-877 Response body parameters

Parameter	Type	Description
authorizers	Array of **DeviceAuthorizerSimple** objects	**Parameter description**: custom authenticator list.
page	**Page** object	Pagination information of the query results.

Table 1-878 DeviceAuthorizerSimple

Parameter	Type	Description
authorizer_id	String	**Parameter description**: custom authenticator ID.
authorizer_name	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
func_name	String	**Parameter description**: name of the function corresponding to the custom authenticator.
func_urn	String	**Parameter description**: function uniform resource name (URN), which uniquely identifies the function. It is the address of the processing function corresponding to the custom authenticator.
signing_enable	Boolean	**Parameter description**: whether to enable signature authentication. After signature authentication is enabled, authentication information that does not meet signature requirements will be rejected to reduce invalid function calls. You are advised to perform signature authentication for security, which is enabled by default. Default: **true**
default_authorizer	Boolean	**Parameter description**: whether the current custom authenticator is the default one. The default value is **false**. If this parameter is set to**true**, the current authenticator policy is used for authentication on all devices that support SNI unless otherwise specified. Default: **false**
status	String	**Parameter description**: whether to enable the authentication mode. • **ACTIVE**: The authentication is enabled. • **INACTIVE**: The authentication is disabled. Default: **INACTIVE**

Parameter	Type	Description
cache_enable	Boolean	**Parameter description**: whether to enable the cache function. The default value is **false**. If this parameter is set to **true** and the device input parameters (username, client ID, password, certificate information, and function URN) remain unchanged, the cache result is directly used when the cache result exists. You are advised to set this parameter to **false** during debugging, set this parameter to **true** during production to avoid frequent function invoking. Default: **false**
create_time	String	Time when the custom authenticator is queried on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the custom authenticator is updated and queried on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Table 1-879 Page

Parameter	Type	Description
count	Long	Total number of records that meet the filter criteria.
marker	String	ID of the last record in this query, which can be used in the next query.

Example Requests

Queries the custom authenticator list of a tenant.

```
GET https://{endpoint}/v5/iot/{project_id}/device-authorizers
```

Example Responses

Status code: 200

OK

```
{
  "authorizers" : [ {
    "authorizer_id" : "5c90fa7d3c4e4405e8525079",
    "authorizer_name" : "myTest",
    "func_name" : "mqtt_auth",
```



```
    "func_urn" : "urn:fss:cn-north-5:d92d9c5eb8e347b5bb31ecfe5bc0c4e1:function:default:mqtt_auth:lates",
    "signing_enable" : true,
    "default_authorizer" : false,
    "status" : "ACTIVE",
    "cache_enable" : false,
    "create_time" : "20231031T070547Z",
    "update_time" : "20231031T070547Z"
  } ],
  "page" : {
    "count" : 1,
    "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

1.4.27.3 Delete a Custom Authenticator

Function

This API is used by an application to delete a specified custom authenticator from the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

DELETE /v5/iot/{project_id}/device-authorizers/{authorizer_id}

Table 1-880 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
authorizer_id	Yes	String	Custom authenticator ID.

Request Parameters

Table 1-881 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

None

Example Requests

Deletes a custom authenticator with a specified ID.

```
DELETE https://{endpoint}/v5/iot/{project_id}/device-authorizers/{authorizer_id}
```

Example Responses

None

Status Codes

Status Code	Description
204	No Content

Status Code	Description
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.4.27.4 Query Custom Authenticator Details

Function

This API is used by an application to query details about a custom authenticator with a specified ID on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

GET /v5/iot/{project_id}/device-authorizers/{authorizer_id}

Table 1-882 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
authorizer_id	Yes	String	Custom authenticator ID.

Request Parameters

Table 1-883 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Response Parameters

Status code: 200

Table 1-884 Response body parameters

Parameter	Type	Description
authorizer_id	String	**Parameter description**: custom authenticator ID.
authorizer_name	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Parameter	Type	Description
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**
func_urn	String	**Parameter description**: function uniform resource name (URN), which uniquely identifies the function. It is the address of the processing function corresponding to the custom authenticator. Minimum: **0** Maximum: **65535**
signing_enable	Boolean	**Parameter description**: whether to enable signature authentication (enabled by default). You are advised to enable this function. If this function is enabled, authentication information that does not meet signature requirements will be rejected to reduce invalid function calls. Default: **true**
signing_token	String	**Parameter description**: key value for signature authentication. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
signing_public_key	String	**Parameter description**: public secret for signature authentication. Used to check whether the signature information carried by the device is correct. Minimum: **0** Maximum: **65535**
default_authorizer	Boolean	**Parameter description**: whether the authentication mode is used by default. The default value is **false**. Default: **false**
status	String	**Parameter description**: whether to enable the authentication mode. • **ACTIVE**: The authentication is enabled. • **INACTIVE**: The authentication is disabled. Default: **INACTIVE**

Parameter	Type	Description
cache_enable	Boolean	**Parameter description**: whether to enable the cache function. The default value is **false**. If this parameter is set to **true** and the device input parameters (username, client ID, password, certificate information, and function URN) remain unchanged, the cache result is directly used when the cache result exists. You are advised to set this parameter to **false** during debugging, set this parameter to **true** during production to avoid frequent function invoking. Default: **false**
create_time	String	Time when operations on custom authenticator are performed on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.
update_time	String	Time when the custom authenticator is updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Queries details about a custom authenticator with a specified ID.

```
GET https://{endpoint}/v5/iot/{project_id}/device-authorizers/{authorizer_id}
```

Example Responses

Status code: 200

OK

```
{
  "authorizer_id" : "5c90fa7d3c4e4405e8525079",
  "authorizer_name" : "myTest",
  "func_name" : "mqtt_auth",
  "func_urn" : "urn:fss:cn-north-5:d92d9c5eb8e347b5bb31ecfe5bc0c4e1:function:default:mqtt_auth:latest",
  "signing_enable" : true,
  "signing_token" : "string",
  "signing_public_key" : "string",
  "default_authorizer" : false,
  "status" : "ACTIVE",
  "cache_enable" : false,
  "create_time" : "20231031T070547Z",
  "update_time" : "20231031T070547Z"
}
```


Status Codes

Status Code	Description
200	OK
404	Not Found

Error Codes

See **Error Codes**.

1.4.27.5 Update a Custom Authenticator with a Specified ID

Function

This API is used by an application to update a custom authenticator with a specified ID on the IoT platform.

Debugging

You can debug this API through automatic authentication in **API Explorer** or use the SDK sample code generated by API Explorer.

URI

PUT /v5/iot/{project_id}/device-authorizers/{authorizer_id}

Table 1-885 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	**Parameter description**: project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
authorizer_id	Yes	String	Custom authenticator ID.

Request Parameters

Table 1-886 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	No	String	**Parameter description**: user token. Obtain the token by calling the IAM API for **obtaining a user token through password authentication**. In the returned response header, **X-Subject-Token** is the desired user token. For details about how to obtain the token, see **Token Authentication**.
Instance-Id	No	String	**Parameter description**: instance ID. Unique identifier of each instance in the physical multi-tenant scenario. Mandatory for professional editions and recommended in other cases. Log in to the IoTDA console and choose **Overview** in the navigation pane to view the instance ID. For details, see **Viewing Instance Details**.

Table 1-887 Request body parameters

Parameter	Mandatory	Type	Description
authorizer_name	No	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
func_urn	No	String	**Parameter description**: function uniform resource name (URN), which uniquely identifies the function. It is the address of the processing function corresponding to the custom authenticator.

Parameter	Mandatory	Type	Description
signing_enable	No	Boolean	**Parameter description**: whether to enable signature authentication (enabled by default). You are advised to enable this function. If this function is enabled, authentication information that does not meet signature requirements will be rejected to reduce invalid function calls, and **signing_token** and **signing_public_key** are mandatory.
signing_token	No	String	**Parameter description**: key value for signature authentication. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
signing_public_key	No	String	**Parameter description**: public secret for signature authentication. Used to check whether the signature information carried by the device is correct.
default_authorizer	No	Boolean	**Parameter description**: whether the current custom authenticator is the default one. The default value is **false**. If this parameter is set to**true**, the current authenticator policy is used for authentication on all devices that support SNI unless otherwise specified.
status	No	String	**Parameter description**: whether to enable the authentication mode (enabled by default). • **ACTIVE**: The authentication is enabled. • **INACTIVE**: The authentication is disabled.

Parameter	Mandatory	Type	Description
cache_enable	No	Boolean	**Parameter description**: whether to enable the cache function. The default value is **false**. If this parameter is set to **true** and the device input parameters (username, client ID, password, certificate information, and function URN) remain unchanged, the cache result is directly used when the cache result exists. You are advised to set this parameter to **false** during debugging, set this parameter to **true** during production to avoid frequent function invoking.

Response Parameters

Status code: 200

Table 1-888 Response body parameters

Parameter	Type	Description
authorizer_id	String	**Parameter description**: custom authenticator ID.
authorizer_name	String	**Parameter description**: name of a custom authenticator, which must be unique under a tenant. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
func_name	String	**Parameter description**: function name. Minimum: **0** Maximum: **65535**
func_urn	String	**Parameter description**: function uniform resource name (URN), which uniquely identifies the function. It is the address of the processing function corresponding to the custom authenticator. Minimum: **0** Maximum: **65535**

Parameter	Type	Description
signing_enable	Boolean	**Parameter description**: whether to enable signature authentication (enabled by default). You are advised to enable this function. If this function is enabled, authentication information that does not meet signature requirements will be rejected to reduce invalid function calls. Default: **true**
signing_token	String	**Parameter description**: key value for signature authentication. **Value**: The value can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
signing_public_key	String	**Parameter description**: public secret for signature authentication. Used to check whether the signature information carried by the device is correct. Minimum: **0** Maximum: **65535**
default_authorizer	Boolean	**Parameter description**: whether the authentication mode is used by default. The default value is **false**. Default: **false**
status	String	**Parameter description**: whether to enable the authentication mode. ● **ACTIVE**: The authentication is enabled. ● **INACTIVE**: The authentication is disabled. Default: **INACTIVE**
cache_enable	Boolean	**Parameter description**: whether to enable the cache function. The default value is **false**. If this parameter is set to **true** and the device input parameters (username, client ID, password, certificate information, and function URN) remain unchanged, the cache result is directly used when the cache result exists. You are advised to set this parameter to **false** during debugging, set this parameter to **true** during production to avoid frequent function invoking. Default: **false**
create_time	String	Time when operations on custom authenticator are performed on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Parameter	Type	Description
update_time	String	Time when the custom authenticator is updated on the IoT platform. The value is in the format of yyyyMMdd'T'HHmmss'Z', for example, **20151212T121212Z**.

Example Requests

Updates a custom authenticator.

```
PUT https://{endpoint}/v5/iot/{project_id}/device-authorizers/{authorizer_id}

{
  "authorizer_name" : "myTest",
  "func_urn" : "urn:fss:cn-north-5:d92d9c5eb8e347b5bb31ecfe5bc0c4e1:function:default:mqtt_auth:latest",
  "signing_enable" : true,
  "signing_token" : "string",
  "signing_public_key" : "string",
  "default_authorizer" : false,
  "status" : "ACTIVE",
  "cache_enable" : false
}
```

Example Responses

Status code: 200

OK

```
{
  "authorizer_id" : "5c90fa7d3c4e4405e8525079",
  "authorizer_name" : "myTest",
  "func_name" : "mqtt_auth",
  "func_urn" : "urn:fss:cn-north-5:d92d9c5eb8e347b5bb31ecfe5bc0c4e1:function:default:mqtt_auth:latest",
  "signing_enable" : true,
  "signing_token" : "string",
  "signing_public_key" : "string",
  "default_authorizer" : false,
  "status" : "ACTIVE",
  "cache_enable" : false,
  "create_time" : "20231031T070547Z",
  "update_time" : "20231031T070947Z"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Forbidden
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

1.5 Permissions and Supported Actions

1.5.1 Permissions and Supported Actions

This chapter describes fine-grained permissions management for your IoTDA. If your Huawei Cloud account does not need individual IAM users, then you may skip over this chapter.

By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on cloud services based on the permissions.

You can grant users permissions by using **roles** and **policies**. Roles are provided by IAM to define service-based permissions that match user's job responsibilities. Policies are more fine-grained, API-based permissions required to perform operations on specific cloud resources under certain conditions, meeting requirements for secure access control.

NOTE

Use policy-based authorization if you want to allow or deny the access to an API.

Your account has all the permissions required to call all APIs, but IAM users under your account must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users that have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user creates a device using an API, the user must have been granted permissions that allow the **iotda:devices:register** action.

Supported Actions

IoTDA provides system-defined policies that can be directly used in IAM. You can also create custom policies to supplement system-defined policies for more refined access control. Actions supported by policies are specific to APIs. Common concepts related to policies include:

- API Group: group to which the API belongs.
- Permissions: statements in a policy that allow or deny certain operations
- APIs: APIs that will be called for performing certain operations.
- Actions: specific operations that are allowed or denied in a custom policy

1.5.2 List of Supported Actions

API Group	Permission	API	Action
Product management	Creating a product	POST /v5/iot/{project_id}/products	iotda:products:create
	Querying the product list	GET /v5/iot/{project_id}/products	iotda:products:queryList
	Querying a product	GET /v5/iot/{project_id}/products/{product_id}	iotda:products:query
	Modifying a product	PUT /v5/iot/{project_id}/products/{product_id}	iotda:products:modify
	Deleting a product	DELETE /v5/iot/{project_id}/products/{product_id}	iotda:products:delete
Device management	Creating a device	POST /v5/iot/{project_id}/devices	iotda:devices:register
	Querying the device list	GET /v5/iot/{project_id}/devices	iotda:devices:queryList
	Querying a device	GET /v5/iot/{project_id}/devices/{device_id}	iotda:devices:query
	Modifying a device	PUT /v5/iot/{project_id}/devices/{device_id}	iotda:devices:modify
	Deleting a device	DELETE /v5/iot/{project_id}/devices/{device_id}	iotda:devices:delete
	Resetting a device secret	POST /v5/iot/{project_id}/devices/{device_id}/action	iotda:devices:resetSecret
	Freezing a device	POST /v5/iot/{project_id}/devices/{device_id}/freeze	iotda:devices:freeze
	Unfreezing a device	POST /v5/iot/{project_id}/devices/{device_id}/unfreeze	iotda:devices:unfreeze
	Resetting a device fingerprint	POST /v5/iot/{project_id}/devices/{device_id}/reset-fingerprint	iotda:devices:resetFingerprint
	Querying device list flexibly	POST /v5/iot/{project_id}/search/query-devices	iotda:devices:queryList

Device message management	Delivering a device message	POST /v5/iot/{project_id}/devices/{device_id}/messages	iotda:messages:send
	Querying device messages	GET /v5/iot/{project_id}/devices/{device_id}/messages	iotda:messages:queryList
	Querying a message by message ID	GET /v5/iot/{project_id}/devices/{device_id}/messages/{message_id}	iotda:messages:query
	Broadcasting a message	POST /v5/iot/{project_id}/broadcast-messages	iotda:message:broadcast
Device command management	Delivering a device command	POST /v5/iot/{project_id}/devices/{device_id}/commands	iotda:commands:send
	Delivering an asynchronous command	POST /v5/iot/{project_id}/devices/{device_id}/async-commands	iotda:asynccommands:send
	Querying a command with a specific ID	GET /v5/iot/{project_id}/devices/{device_id}/async-commands/{command_id}	iotda:asynccommands:query
Device property management	Modifying device properties	PUT /v5/iot/{project_id}/devices/{device_id}/properties	iotda:properties:modify
	Querying device properties	GET /v5/iot/{project_id}/devices/{device_id}/properties	iotda:properties:query
Device shadow	Querying device shadow data	GET /v5/iot/{project_id}/devices/{device_id}/shadow	iotda:shadow:query
	Configuring desired device shadow data	PUT /v5/iot/{project_id}/devices/{device_id}/shadow	iotda:shadow:config
AMQP queue management	Creating an AMQP queue	POST /v5/iot/{project_id}/amqp-queues	iotda:amqpqueue:create
	Querying the AMQP list	GET /v5/iot/{project_id}/amqp-queues	iotda:amqpqueue:queryList
	Querying an AMQP queue	GET /v5/iot/{project_id}/amqp-queues/{queue_id}	iotda:amqpqueue:query
	Deleting an AMQP queue	DELETE /v5/iot/{project_id}/amqp-queues/{queue_id}	iotda:amqpqueue:delete

Access code management	Generating an access credential	POST /v5/iot/{project_id}/auth/accesscode	iotda:accesscode:create
Forwarding rule management	Creating a rule triggering condition	POST /v5/iot/{project_id}/routing-rule/rules	iotda:routingrules:create
	Querying the rule triggering condition list	GET /v5/iot/{project_id}/routing-rule/rules	iotda:routingrules:queryList
	Querying a rule triggering condition	GET /v5/iot/{project_id}/routing-rule/rules/{rule_id}	iotda:routingrules:query
	Modifying a rule triggering condition	PUT /v5/iot/{project_id}/routing-rule/rules/{rule_id}	iotda:routingrules:modify
	Deleting a rule triggering condition	DELETE /v5/iot/{project_id}/routing-rule/rules/{rule_id}	iotda:routingrules:delete
Forwarding rule action management	Creating a rule action	POST /v5/iot/{project_id}/routing-rule/actions	iotda:routingactions:create
	Querying the rule action list	GET /v5/iot/{project_id}/routing-rule/actions	iotda:routingactions:queryList
	Querying a rule action	GET /v5/iot/{project_id}/routing-rule/actions/{action_id}	iotda:routingactions:query
	Modifying a rule action	PUT /v5/iot/{project_id}/routing-rule/actions/{action_id}	iotda:routingactions:modify
	Deleting a rule action	DELETE /v5/iot/{project_id}/routing-rule/actions/{action_id}	iotda:routingactions:delete
Linkage rule management	Creating a rule	POST /v5/iot/{project_id}/rules	iotda:rules:create
	Querying the rule list	GET /v5/iot/{project_id}/rules	iotda:rules:queryList
	Modifying a rule	PUT /v5/iot/{project_id}/rules/{rule_id}	iotda:rules:modify

	Querying a rule	GET /v5/iot/{project_id}/rules/{rule_id}	iotda:rules:query
	Deleting a rule	DELETE /v5/iot/{project_id}/rules/{rule_id}	iotda:rules:delete
	Modifying the rule status	PUT /v5/iot/{project_id}/rules/{rule_id}/status	iotda:rules:modifyStatus
Group management	Adding a device group	POST /v5/iot/{project_id}/device-group	iotda:group:create
	Querying the device group list	GET /v5/iot/{project_id}/device-group	iotda:group:queryList
	Querying a device group	GET /v5/iot/{project_id}/device-group/{group_id}	iotda:group:query
	Modifying a device group	PUT /v5/iot/{project_id}/device-group/{group_id}	iotda:group:modify
	Deleting a device group	DELETE /v5/iot/{project_id}/device-group/{group_id}	iotda:group:delete
	Managing devices in a device group	POST /v5/iot/{project_id}/device-group/{group_id}/action	iotda:group:addDevice
	Querying devices in a device group	GET /v5/iot/{project_id}/device-group/{group_id}/devices	iotda:group:queryDeviceList
Device tag management	Binding a tag	POST /v5/iot/{project_id}/tags/bind-resource	iotda:tags:bind
	Unbinding a tag	POST /v5/iot/{project_id}/tags/unbind-resource	iotda:tags:unbind
	Querying resources by tag	POST /v5/iot/{project_id}/tags/query-resources	iotda:tags:query
Resource space management	Querying the resource space list	GET /v5/iot/{project_id}/apps	iotda:apps:queryList
	Creating a resource space	POST /v5/iot/{project_id}/apps	iotda:app:create
	Querying a resource space	GET /v5/iot/{project_id}/apps/{app_id}	iotda:apps:query

	Deleting a resource space	DELETE /v5/iot/{project_id}/apps/{app_id}	iotda:apps:delete
Batch task management	Creating a batch task	POST /v5/iot/{project_id}/batchtasks	iotda:batchtasks:create
	Querying the batch task list	GET /v5/iot/{project_id}/batchtasks	iotda:batchtasks:queryList
	Querying a batch task	GET /v5/iot/{project_id}/batchtasks/{task_id}	iotda:batchtasks:query
	Retrying a batch task	POST /v5/iot/{project_id}/batchtasks/{task_id}/retry	iotda:batchtasks:retry
	Stopping a batch task	POST /v5/iot/{project_id}/batchtasks/{task_id}/stop	iotda:batchtasks:stop
	Deleting a batch task	DELETE /v5/iot/{project_id}/batchtasks/{task_id}	iotda:batchtasks:delete
Batch task file management	Uploading a batch task file	POST /v5/iot/{project_id}/batchtask-files	iotda:batchtasks:stop
	Querying the list of batch task files	GET /v5/iot/{project_id}/batchtask-files	iotda:batchtaskfiles:queryList
	Deleting a batch task file	DELETE /v5/iot/{project_id}/batchtask-files/{file_id}	iotda:batchtaskfiles:delete
Certificate management	Uploading a device CA certificate	POST /v5/iot/{project_id}/certificates	iotda:certificates:upload
	Obtaining the device CA certificate list	GET /v5/iot/{project_id}/certificates	iotda:certificates:queryList
	Deleting a device CA certificate	DELETE /v5/iot/{project_id}/certificates/{certificate_id}	iotda:certificates:delete
	Verifying a device CA certificate	POST /v5/iot/{project_id}/certificates/{certificate_id}/action	iotda:certificates:check

Software / Firmware upgrade package management	Creating an OTA upgrade package	POST /v5/iot/{project_id}/ota-upgrades/packages	iotda:otapackages:create
	Querying the OTA upgrade package list	GET /v5/iot/{project_id}/ota-upgrades/packages	iotda:otapackages:queryList
	Obtaining OTA upgrade package details	GET /v5/iot/{project_id}/ota-upgrades/packages/{package_id}	iotda:otapackages:query
	Deleting an OTA upgrade package	DELETE /v5/iot/{project_id}/ota-upgrades/packages/{package_id}	iotda:otapackages:delete
Tunnel management	Querying the tunnel list	GET /v5/iot/{project_id}/tunnels	iotda:tunnel:queryList
	Creating a device tunnel	POST /v5/iot/{project_id}/tunnels	iotda:tunnel:create
	Deleting a device tunnel	DELETE /v5/iot/{project_id}/tunnels/{id}	iotda:tunnel:delete
	Querying tunnel details	GET /v5/iot/{project_id}/tunnels/{id}	iotda:tunnel:query
	Modifying a device tunnel	PUT /v5/iot/{project_id}/tunnels/{id}	iotda:tunnel:update

1.6 Examples

1.6.1 Example 1: Creating Devices in Batches Using a Template

Scenarios

This topic describes how to create devices in batches using APIs. For details about how to call APIs, see **1.2 Calling APIs**.

The platform allows you to create devices in batches using request parameters or a template. This topic describes how to create devices in batches using a template.

Involved APIs

- **1.4.1.2 Query the Product List**: Determine the product to which devices to create belong.
- **1.4.16.7.1 Upload a Batch Task File**: Fill in a batch task file and upload the file to determine the devices to create in batches.

- **1.4.16.1 Create a Batch Task**: Create devices in batches by using a template.
- **1.4.16.3 Query a Batch Task**: Confirm the batch device creation result.

Procedure

Step 1 Determine the product to which devices to create belong.

1. Query the product list
   - API

     URL: GET /v5/iot/{project_id}/products

     For details, see **1.4.1.2 Query the Product List**.
   - Example request

     ```
     GET https://{Endpoint}/v5/iot/{project_id}/products?
     limit={limit}&marker={marker}&app_id={app_id}&offset={offset}
     Content-Type: application/json
     X-Auth-Token: ********
     Instance-Id: ********
     ```
   - Example response

     Status Code: 200 OK

     ```
     Content-Type: application/json

     {
       "products" : [ {
         "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
         "app_name" : "testAPP01",
         "product_id" : "5ba24f5ebbe8f56f5a14f605",
         "name" : "Thermometer",
         "device_type" : "Thermometer",
         "protocol_type" : "CoAP",
         "data_format" : "binary",
         "manufacturer_name" : "ABC",
         "industry" : "smartCity",
         "description" : "this is a thermometer produced by Huawei",
         "create_time" : "20190303T081011Z"
       } ],
       "page" : {
         "count" : 10,
         "marker" : "5c90fa7d3c4e4405e8525079"
       }
     }
     ```
2. Select a product based on project requirements and record the value of **product_id**.

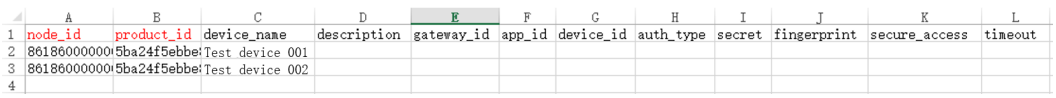
Step 2 Download the batch task template.

Click **here** to download the template.

Step 3 Set parameters of devices to create in the template.

Edit the template downloaded in **2** by referring to **1.4.2.1 Create a Device**.

Example:



	A	B	C	D	E	F	G	H	I	J	K	L
1	node_id	product_id	device_name	description	gateway_id	app_id	device_id	auth_type	secret	fingerprint	secure_access	timeout
2	861860000006	5ba24f5ebbe8	Test device 001									
3	861860000006	5ba24f5ebbe8	Test device 002									
4												

Step 4 Update a batch task file.

1. Update a batch task file.

- API

  URL: POST /v5/iot/{project_id}/batchtask-files

  For details, see **1.4.16.7.1 Upload a Batch Task File**.

- Example request

```
POST https://{Endpoint}/v5/iot/{project_id}/batchtask-files
Content-Type: multipart/form-data
X-Auth-Token: ********
Instance-Id: ********
```

- Example response

  Status Code: 201 Created

```
Content-Type: application/json

{
  "file_id" : "0c3c77dd-42a2-4309-9e10-da2e8bf64ac3",
  "file_name" : "BatchCreateDevices_test01.xlsx",
  "upload_time" : "20200617T081608Z"
}
```

2. Record the value of **file_id**.

Step 5 Create a batch task.

1. Create a batch task.

- API

  URL: POST /v5/iot/{project_id}/batchtasks

  For details, see **1.4.16.1 Create a Batch Task**.

- Example request

```
POST https://{Endpoint}/v5/iot/{project_id}/batchtasks
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

{
  "app_id" : "Ev8FVvCfOdQDzrFrxSOemiw_aMca",
  "task_name" : "BatchCommandTask",
  "task_type" : "softwareUpgrade",
  "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
  "targets_filter" : {
    "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
  },
  "document" : {
    "package_id" : "32822e5744a45ede319d2c50"
  },
  "task_policy" : {
    "schedule_time" : "20151212T121212Z",
    "retry_count" : 5,
    "retry_interval" : 60
  },
  "document_source" : "jeQDJQZltU8iKgFFoW060F5SGZka"
}
```

- Example response

  Status Code: 201 Created

```
Content-Type: application/json

{
  "task_id" : "5c8ba99030344005c02316ad",
  "task_name" : "testname",
  "task_type" : "softwareUpgrade",
  "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
  "targets_filter" : {
    "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
```



```
  },
  "document" : {
    "package_id" : "32822e5744a45ede319d2c50"
  },
  "task_policy" : {
    "schedule_time" : "20151212T121212Z",
    "retry_count" : 5,
    "retry_interval" : 60
  },
  "status" : "Success",
  "status_desc" : "string",
  "task_progress" : {
    "total" : 0,
    "processing" : 0,
    "success" : 0,
    "fail" : 0,
    "waitting" : 0,
    "fail_wait_retry" : 0,
    "stopped" : 0
  },
  "create_time" : "20151212T121212Z"
}
```

2. Record the value of **task_id**.

Step 6 Query the batch task.

1. Query the batch task.
   - API

     URL: GET /v5/iot/{project_id}/batchtasks/{task_id}

     For details, see **1.4.16.3 Query a Batch Task**.
   - Example request

     ```
     GET https://{Endpoint}/v5/iot/{project_id}/batchtasks/{task_id}?
     limit={limit}&marker={marker}&offset={offset}
     Content-Type: application/json
     X-Auth-Token: ********
     Instance-Id: ********
     ```
   - Example response

     Status Code: 200 OK

     ```
     Content-Type: application/json

     {
       "batchtasks" : {
         "task_id" : "5c8ba99030344005c02316ad",
         "task_name" : "testname",
         "task_type" : "softwareUpgrade",
         "targets" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ],
         "targets_filter" : {
           "group_ids" : [ "e495cf17-ff79-4294-8f64-4d367919d665" ]
         },
         "document" : {
           "package_id" : "32822e5744a45ede319d2c50"
         },
         "task_policy" : {
           "schedule_time" : "20151212T121212Z",
           "retry_count" : 5,
           "retry_interval" : 60
         },
         "status" : "Success",
         "status_desc" : "string",
         "task_progress" : {
           "total" : 0,
           "processing" : 0,
           "success" : 0,
           "fail" : 0,
     ```



```
    "waitting" : 0,
    "fail_wait_retry" : 0,
    "stopped" : 0
   },
   "create_time" : "20151212T121212Z"
  },
  "task_details" : [ {
   "target" : "e495cf17-ff79-4294-8f64-4d367919d665",
   "status" : "Success",
   "output" : "success",
   "error" : {
    "error_code" : "IOTDA.000002",
    "error_msg" : "The request is unauthorized."
   }
  } ],
  "page" : {
   "count" : 10,
   "marker" : "5c90fa7d3c4e4405e8525079"
  }
}
```


2. Check the task execution result.

   Check whether the batch device creation task is complete based on the response.

----End

1.6.2 Example 2: Delivering a Message to a Specific Device

Scenarios

This topic describes how to deliver messages to a specific device using APIs. For details about how to call APIs, see **Calling APIs**.

Involved APIs

- **1.4.2.2 Query the Device List**: Determine the device to which a message is to be delivered.
- **1.4.3.1 Deliver a Message to a Device**: Deliver the message to the specified device.
- **1.4.3.3 Query a Message by Message ID**: Confirm the message delivery result.

Procedure

Step 1 Determine the device to which a message is to be delivered.

1. Query the device list
   - API

     URL: GET /v5/iot/{project_id}/devices

     For details, see **1.4.2.2 Query the Device List**.
   - Example request

```
GET https://{Endpoint}/v5/iot/{project_id}/devices?
product_id={product_id}&gateway_id={gateway_id}&is_cascade_query={is_cascade_query}&node_
id={node_id}&device_name={device_name}&limit={limit}&marker={marker}&offset={offset}&star
t_time={start_time}&end_time={end_time}&app_id={app_id}
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********
```


- Example response

  Status Code: 200 OK

```
Content-Type: application/json

{
  "devices" : [ {
    "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "description" : "watermeter device",
    "product_name" : "Thermometer",
    "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
    "sw_version" : "1.1.0",
    "tags" : [ {
      "tag_value" : "testTagValue",
      "tag_key" : "testTagName"
    } ],
    "app_name" : "testAPP01",
    "device_name" : "dianadevice",
    "node_type" : "ENDPOINT",
    "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
    "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
    "fw_version" : "1.1.0",
    "node_id" : "ABC123456789",
    "status" : "INACTIVE"
  } ],
  "page" : {
    "marker" : "5c8f3d2d3df1f10d803adbda",
    "count" : 100
  }
}
```

2. Select a device based on project requirements and record the value of **device_id**.

Step 2 Deliver the message to the device specified in **Step 1**.

1. Deliver a message.
   - API

     URL: POST /v5/iot/{project_id}/devices/{device_id}/messages

     For details, see **1.4.3.1 Deliver a Message to a Device**.

   - Example request

```
POST https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}/messages
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

{
  "message_id" : "99b32da9-cd17-4cdf-a286-f6e849cbc364",
  "name" : "messageName",
  "message" : "HelloWorld",
  "topic" : "messageDown"
}
```

   - Example response

     Status Code: 201 Created

```
Content-Type: application/json

{
  "message_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
  "result" : {
    "status" : "PENDING",
    "created_time" : "20151212T121212Z",
    "finished_time" : "20151212T121213Z"
  }
}
```


2. Record the value of **message_id** in the response.

Step 3 Confirm the message delivery result.

1. Query the message with **message_id** obtained in **Step 2.2**.
   - API

     URL: GET /v5/iot/{project_id}/devices/{device_id}/messages/{message_id}

     For details, see **1.4.3.3 Query a Message by Message ID**.
   - Example request

     ```
     GET https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}/messages/{message_id}
     Content-Type: application/json
     X-Auth-Token: ********
     Instance-Id: ********
     ```
   - Example response

     Status Code: 200 OK

     ```
     Content-Type: application/json

     {
       "message_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",
       "name" : "message_name",
       "message" : "string",
       "topic" : "string",
       "status" : "PENDING",
       "created_time" : "20151212T121212Z",
       "finished_time" : "20151212T121212Z"
     }
     ```
2. Check the message delivery result based on the value of **status** in the response.

----End

1.6.3 Example 3: Creating a Device in a Specific Resource Space

Scenarios

This topic describes how to create a device in a specific resource space using APIs. For details about how to call APIs, see **Calling APIs**.

By default, the platform creates a default resource space for you. Unless otherwise specified, a device created is allocated to the default resource space. To manage devices by resource space, you can specify a resource space when creating a device.

Involved APIs

- **1.4.15.2 Create a Resource Space**: Create a resource space.
- **1.4.1.1 Create a Product**: Create a product in the specified resource space.
- **1.4.2.1 Create a Device**: Create a device in the specified resource space.

Procedure

Step 1 Create a resource space.

1. Create a resource space.
   - API

     URL: POST /v5/iot/{project_id}/apps

     For details, see **1.4.15.2 Create a Resource Space**.
   - Example request

```
POST https://{Endpoint}/v5/iot/{project_id}/apps
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

{
  "app_name" : "testApp"
}
```

   - Example response

     Status Code: 201 Created

```
Content-Type: application/json

{
  "applications" : [ {
    "app_id" : "0ab87ceecbfc49acbcc8d5acdef3c68c",
    "app_name" : "testApp",
    "create_time" : "20151212T121212Z",
    "default_app" : true
  } ]
}
```

2. Record the value of **app_id** in the response.

Step 2 Create a product in the resource space created in **Step 1**.

1. Create a product by carrying the **app_id** recorded in **Step 1.2**.
   - API

     URL: POST /v5/iot/{project_id}/products

     For details, see **1.4.1.1 Create a Product**.
   - Example request

```
POST https://{Endpoint}/v5/iot/{project_id}/products
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

{
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "name" : "Thermometer",
  "device_type" : "Thermometer",
  "protocol_type" : "CoAP",
  "data_format" : "binary",
  "manufacturer_name" : "ABC",
  "industry" : "smartCity",
  "description" : "this is a thermometer produced by Huawei",
  "service_capabilities" : [ {
    "service_type" : "temperature",
    "service_id" : "temperature",
    "description" : "temperature",
    "properties" : [ {
      "unit" : "centigrade",
      "min" : "1",
      "method" : "R",
      "max" : "100",
      "data_type" : "decimal",
      "description" : "force",
      "step" : 0.1,
      "default_value" : {
```



```
      "color" : "red",
      "size" : 1
    },
    "enum_list" : [ "string" ],
    "required" : true,
    "property_name" : "temperature",
    "max_length" : 100
  } ],
  "commands" : [ {
    "command_name" : "reboot",
    "responses" : [ {
      "response_name" : "ACK",
      "paras" : [ {
        "unit" : "km/h",
        "min" : "1",
        "max" : "100",
        "para_name" : "force",
        "data_type" : "string",
        "description" : "force",
        "step" : 0.1,
        "enum_list" : [ "string" ],
        "required" : false,
        "max_length" : 100
      } ]
    } ],
    "paras" : [ {
      "unit" : "km/h",
      "min" : "1",
      "max" : "100",
      "para_name" : "force",
      "data_type" : "string",
      "description" : "force",
      "step" : 0.1,
      "enum_list" : [ "string" ],
      "required" : false,
      "max_length" : 100
    } ]
  } ],
  "events" : [ {
    "event_type" : "reboot",
    "paras" : [ {
      "unit" : "km/h",
      "min" : "1",
      "max" : "100",
      "para_name" : "force",
      "data_type" : "string",
      "description" : "force",
      "step" : 0.1,
      "enum_list" : [ "string" ],
      "required" : false,
      "max_length" : 100
    } ]
  } ],
  "option" : "Mandatory"
} ],
"app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka"
}
```

– Example response

Status Code: 201 Created

```
Content-Type: application/json

{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name" : "testAPP01",
  "product_id" : "5ba24f5ebbe8f56f5a14f605",
  "name" : "Thermometer",
  "device_type" : "Thermometer",
  "protocol_type" : "CoAP",
```



```
"data_format" : "binary",
"manufacturer_name" : "ABC",
"industry" : "smartCity",
"description" : "this is a thermometer produced by Huawei",
"service_capabilities" : [ {
  "service_id" : "temperature",
  "service_type" : "temperature",
  "properties" : [ {
    "property_name" : "temperature",
    "required" : true,
    "data_type" : "decimal",
    "min" : 1,
    "max" : 100,
    "max_length" : 100,
    "step" : 0.1,
    "unit" : "centigrade",
    "method" : "R",
    "description" : "force",
    "default_value" : {
      "color" : "red",
      "size" : 1
    }
  } ],
  "commands" : [ {
    "command_name" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "required" : false,
      "data_type" : "string",
      "min" : 1,
      "max" : 100,
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ],
    "responses" : [ {
      "paras" : [ {
        "para_name" : "force",
        "required" : false,
        "data_type" : "string",
        "min" : 1,
        "max" : 100,
        "max_length" : 100,
        "step" : 0.1,
        "unit" : "km/h",
        "description" : "force"
      } ],
      "response_name" : "ACK"
    } ]
  } ],
  "events" : [ {
    "event_type" : "reboot",
    "paras" : [ {
      "para_name" : "force",
      "required" : false,
      "data_type" : "string",
      "min" : 1,
      "max" : 100,
      "max_length" : 100,
      "step" : 0.1,
      "unit" : "km/h",
      "description" : "force"
    } ]
  } ],
  "description" : "temperature",
  "option" : "Mandatory"
} ],
```



```
  "create_time" : "20190303T081011Z"
}
```

2. Record the value of **product_id** in the response.

Step 3 Create a device in the resource space created in **Step 1.1**.

1. Create a product by carrying the **app_id** and **product_id** recorded in **Step 1.2** and **Step 2.2**, respectively.
   - API

     URL: POST /v5/iot/{project_id}/devices

     For details, see **1.4.2.1 Create a Device**.
   - Example request

```
POST https://{Endpoint}/v5/iot/{project_id}/devices
Content-Type: application/json
X-Auth-Token: ********
Instance-Id: ********

{
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "device_name" : "dianadevice",
  "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
  "auth_info" : {
    "auth_type" : "SECRET",
    "secure_access" : true,
    "fingerprint" : "dc0f1016f495157344ac5f1296335cff725ef22f",
    "secret" : "3b935a250c50dc2c6d481d048cefdc3c",
    "timeout" : 300
  },
  "description" : "watermeter device",
  "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "extension_info" : {
    "aaa" : "xxx",
    "bbb" : 0
  },
  "shadow" : [ {
    "desired" : {
      "temperature" : "60"
    },
    "service_id" : "WaterMeter"
  } ]
}
```

   - Example response

     Status Code: 201 Created

```
Content-Type: application/json

{
  "app_id" : "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name" : "testAPP01",
  "device_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id" : "ABC123456789",
  "gateway_id" : "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "device_name" : "dianadevice",
  "node_type" : "ENDPOINT",
  "description" : "watermeter device",
  "fw_version" : "1.1.0",
  "sw_version" : "1.1.0",
  "auth_info" : {
    "auth_type" : "SECRET",
    "secure_access" : true,
    "fingerprint" : "dc0f1016f495157344ac5f1296335cff725ef22f",
    "secret" : "3b935a250c50dc2c6d481d048cefdc3c",
    "timeout" : 300
```



```
  },
  "product_id" : "b640f4c203b7910fc3cbd446ed437cbd",
  "product_name" : "Thermometer",
  "status" : "INACTIVE",
  "create_time" : "20190303T081011Z",
  "tags" : [ {
    "tag_value" : "testTagValue",
    "tag_key" : "testTagName"
  } ],
  "extension_info" : {
    "aaa" : "xxx",
    "bbb" : 0
  }
}
```


2. Confirm the device creation result based on the response.

----End

1.7 Appendix

1.7.1 Status Code

Table 1-889 describes status codes.

Table 1-889 Status codes

Status Code	Code	Error Code Description
100	Continue	The client should continue sending the request. This interim response is used to inform the client that the initial part of the requests has been received and not rejected by the server.
101	Switching Protocols	The requester has asked the server to switch protocols and the server has agreed to do so. The protocol should be switched only when it is advantageous to do so. For example, the protocol in use is switched to a later version of HTTP.
201	Created	The request has been fulfilled, resulting in the creation of a new resource.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	The request has been fulfilled.
204	NoContent	The server has successfully processed the request, but does not return any response. The status code is returned in response to an HTTP OPTIONS request.

Status Code	Code	Error Code Description
205	Reset Content	The server successfully processed the request, but is not returning any content.
206	Partial Content	The server has fulfilled the partial GET request for the resource.
300	Multiple Choices	There are multiple options for the resource from which the client may choose. For example, this code could be used to present a list of resource characteristics and addresses from which the client, such as a browser, may choose.
301	Moved Permanently	This and all future requests should be permanently directed to the given URI indicated in this response.
302	Found	The requested resource was temporarily moved.
303	See Other	The response to the request can be found under another URI using a GET or POST method.
304	Not Modified	The requested resource has not been modified. In such case, there is no need to retransmit the resource since the client still has a previously downloaded copy.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	This HTTP status code is no longer used.
400	BadRequest	The request could not be understood by the server due to malformed syntax. The client should not repeat the request without modifications.
401	Unauthorized	The authorization information provided by the client is incorrect or invalid.
402	Payment Required	Reserved for future use.
403	Forbidden	The request is rejected. The client should not repeat the request without modifications.
404	NotFound	The requested resource cannot be found. The client should not repeat the request without modifications.

Status Code	Code	Error Code Description
405	MethodNotAllowed	A request method is not supported for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server cannot fulfill the request based on the content characteristics of the request.
407	Proxy Authentication Required	This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The client does not produce a request within the time that the server was prepared to wait. The client may re-initiate the request without any modification at any time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. The resource that the client attempts to create already exists, or the request fails to be processed because of the update of the conflict request.
410	Gone	The requested resource is no longer available. The requested resource has been deleted permanently.
411	Length Required	The server refused to process the request because the request does not specify the length of its content.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The server is refusing to process a request because the request entity is larger than the server is willing or able to process. The server may close the connection to prevent the client from continuing the request. If the server temporarily cannot process the request, the response will contain a **Retry-After** header field.
414	Request-URI Too Large	The server is refusing to service the request because the request URI is longer than the server is willing to interpret.

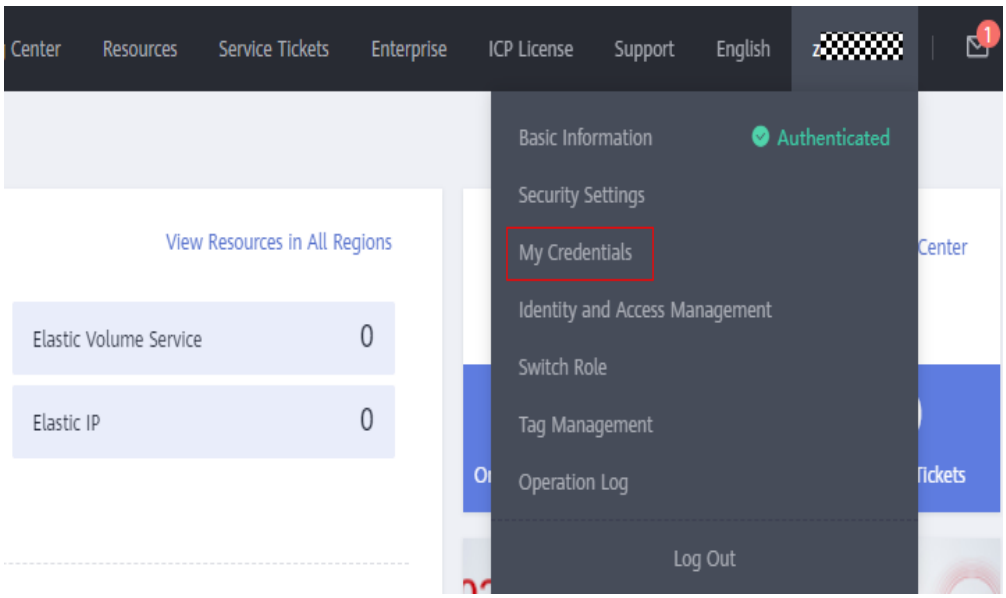
Status Code	Code	Error Code Description
415	Unsupported Media Type	The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the **Expect** request-header field.
422	UnprocessableEntity	The request was well-formed but was unable to be followed due to semantic errors.
429	TooManyRequests	The client sends excessive requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server receives excessive requests within a given time (beyond its processing capability). In this case, the client should repeat requests after the time specified in the **Retry-After** header of the response expires.
500	InternalServerError	The server encountered an unexpected condition which prevented it from fulfilling the request.
501	Not Implemented	The server does not support the functionality required to fulfill the request.
502	Bad Gateway	The server acting as a gateway or proxy receives an invalid response from a remote server.
503	ServiceUnavailable	The requested service is unavailable. The client should not repeat the request without modifications.
504	ServerTimeout	The request cannot be fulfilled within a given amount of time. This status code is returned to the client only when the timeout parameter is specified in the request.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

1.7.2 Obtaining a Project ID

Obtaining a Project ID on the Console

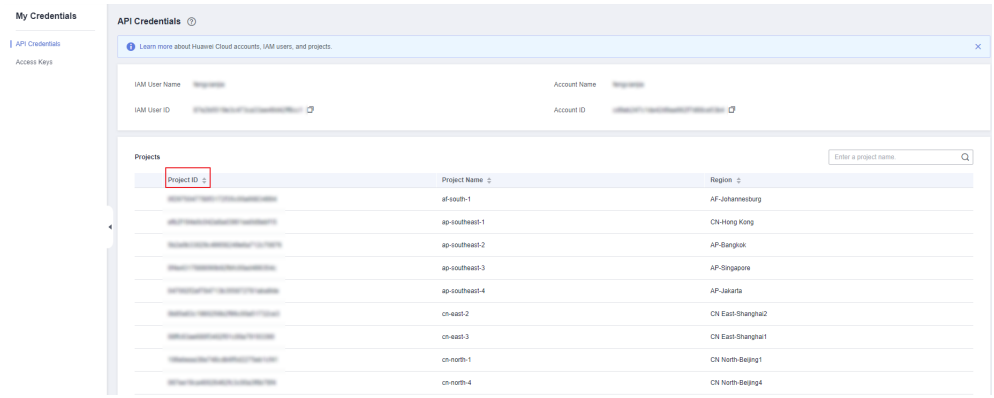
A project ID needs to be specified in the URI or request body for calling certain API. To obtain a project ID, perform the following procedure:

1. On the Huawei Cloud homepage, click **Console** in the upper right corner.
2. Hover over the username in the upper right corner and choose **My Credentials**.



3. On the **My Credentials** page, choose **API Credentials** to view the project ID.

Figure 1-3 API credentials - obtaining a project ID



Obtaining a Project ID by Calling an API

You can obtain a project ID by calling the API **Querying Project Information**.

The API for obtaining a project ID is **GET https://{Endpoint}/v3/projects/**. **{Endpoint}** indicates the endpoint of IAM, which can be obtained from **Platform Connection Information**. For details about API authentication, see **Authentication**.

In the following example, **id** indicates a project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d...",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d1...",
            "name": "cn-north-4",
```



```
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd0..."
            },
            "id": "a4a5d4098fb4474fa22cd05f897xxxx",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```


1.7.3 Error Codes

If an error code starting with APIGW is returned after you call an API, rectify the fault by referring to the instructions provided in **API Gateway Error Codes**.

Status Code	Error Codes	Error Message	Description	Solution
200	IOTDA.014111	Command request timed out. Check whether the device returns a response within the specified time after receiving the request.	Command request timed out. Check whether the device returns a response within the specified time after receiving the request.	When this synchronous API is called, the device needs to respond after receiving the command. Check whether the device returns a response to the platform within the specified time after receiving the request.
400	IOTDA.000006	Invalid input data.	Invalid input data.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.000008	Invalid input. The request format is invalid. For details, see the error message.	Invalid input. The request format is invalid. For details, see the error message.	Check whether the request format is correct.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.000009	Invalid input. Invalid time format.	Invalid input. Invalid time format.	Check whether the time format in request parameters is the same as that specified in the documentation.
400	IOTDA.000010	Invalid input. The start time must be earlier than the end time.	Invalid input. The start time must be earlier than the end time.	The start time in the request must be earlier than the end time.
400	IOTDA.000011	Invalid input. The specified parameter 'pageNo' is out of range.	Invalid input. The specified parameter 'pageNo' is out of range.	Check whether the value of pageNo in the request is within the range specified in the documentation.
400	IOTDA.000012	Invalid input. The specified parameter 'pageSize' is out of range.	Invalid input. The specified parameter 'pageSize' is out of range.	Check whether the value of pageSize in the request is within the range specified in the documentation.
400	IOTDA.000013	Invalid input. The value of 'pageSize' multiplying 'pageNo' exceeds the upper limit.	Invalid input. The value of 'pageSize' multiplying 'pageNo' exceeds the upper limit.	Check the values of pageSize and pageNo.
400	IOTDA.000014	Invalid input. The specified parameter 'nextToken' is out of range.	Invalid input. The specified parameter 'nextToken' is out of range.	Check whether the value of nextToken in the request is within the range specified in the documentation.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.000017	Invalid input. The specified parameter 'limit' is out of range.	Invalid input. The specified parameter 'limit' is out of range.	Check whether the value of limit in the request is within the range specified in the documentation.
400	IOTDA.000018	Invalid input. The specified parameter 'marker' is out of range.	Invalid input. The specified parameter 'marker' is out of range.	Check whether the value of marker in the request is within the range specified in the documentation.
400	IOTDA.000030	Failed to register the resource in Stage. Please try again later.	Failed to register the resource in Stage. Please try again later.	Try again later or contact Huawei technical support engineers.
400	IOTDA.000031	Failed to deregister the resource in Stage. Please try again later.	Failed to deregister the resource in Stage. Please try again later.	Try again later or contact Huawei technical support engineers.
400	IOTDA.000032	Failed to update the resource in Stage. Please try again later.	Failed to update the resource in Stage. Please try again later.	Try again later or contact Huawei technical support engineers.
400	IOTDA.001001	Invalid input for this application.	Invalid input for this application.	Refer to the request parameters in the Huawei Cloud documentation at **Create a Resource Space**.
400	IOTDA.001004	AppId is not in request header.	AppId is not in request header.	Carry the corresponding application ID.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.001011	Invalid input. The specified parameter 'app_id' is not carried.	Invalid input. The specified parameter 'app_id' is not carried.	Add the app_id parameter to the request.
400	IOTDA.001012	Invalid input. The appId already exists.	Invalid input. The appId already exists.	Change a resource space ID.
400	IOTDA.001013	Invalid input. The source and target instance IDs cannot be the same.	Invalid input. The source and target instance IDs cannot be the same.	Check whether the instance ID is correct.
400	IOTDA.004002	Invalid input. The tag_key %s cannot start with 'iot_'.	Invalid input. The tag_key %s cannot start with 'iot_'.	Change the value of tag_key and try again.
400	IOTDA.004003	Invalid input. The tag_key %s does not exist.	Invalid input. The tag_key %s does not exist.	Check whether the value of tag_key is correct or whether the tag exists.
400	IOTDA.004004	Invalid input. This tag key %s already exists.	Invalid input. This tag key %s already exists.	Change the value of tag_key and try again.
400	IOTDA.005002	Upgrade Failed. Invalid device version.	Upgrade Failed. Invalid device version.	Check whether the version number reported by the device is empty.
400	IOTDA.005003	Upgrade failed. Verify device version failed.	Upgrade failed. Verify device version failed.	Ensure that the version number reported by the device is the same as the version number of the software and firmware packages.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.005004	Upgrade failed. The current version cannot be upgraded to the target version.	Upgrade failed. The current version cannot be upgraded to the target version.	Ensure that the version number reported by the device is the same as the source version number of the software and firmware packages.
400	IOTDA.005005	Upgrade failed. Invalid update status.	Upgrade failed. Invalid update status.	Check whether the reported update status is correct.
400	IOTDA.005006	Upgrade Failed. ErrorCode: %s, description : %s.	Upgrade Failed. ErrorCode: %s, description : %s.	Check whether the reported code stream is correct, you can refer to the document **Device Reporting the Upgrade Status**.
400	IOTDA.005007	Upgrade failed. The format of the device data is invalid.	Upgrade failed. The format of the device data is invalid.	Check the format of data reported by the device by following the instructions in**Software and Firmware Upgrade**.
400	IOTDA.005008	Upgrade failed. Reported progress %d% % is invalid.	Upgrade failed. Reported progress %d% % is invalid.	Ensure that the reported upgrade progress ranges from 0 to 100.
400	IOTDA.005009	Upgrade failed. Invalid result_code.	Upgrade failed. Invalid result_code.	Check the value of result_code in the paras structure by following the instructions in **Device Reporting the Upgrade Status** page.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.005010	Upgrade failed. Waiting for %s timed out.	Upgrade failed. Waiting for %s timed out.	The device needs to respond in time. Otherwise, the task times out and fails.
400	IOTDA.005011	Upgrade failed. Send %s command failed.	Upgrade failed. Send %s command failed.	Contact Huawei technical support engineers.
400	IOTDA.009002	The resource model does not exist.	The resource model does not exist.	Check whether the request parameters contain resource and event, or notifyType.
400	IOTDA.009004	The subscription subject does not belong to the current application.	The subscription subject does not belong to the current application.	Check whether the mapping between the subscription record and the application is correct.
400	IOTDA.009005	Invalid request callback URL.	Invalid request callback URL.	Modify the callback URL by following the instructions in **Subscription and Push**.
400	IOTDA.009006	The subscription subject already exists.	The subscription subject already exists.	The subscription record already exists. You do not need to subscribe again.
400	IOTDA.009007	The request channel is invalid.	The request channel is invalid.	Check whether the channel parameter in the request meets the value requirements of the corresponding API.
400	IOTDA.009009	The filter is invalid.	The filter is invalid.	Contact Huawei technical support engineers.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.009010	The resource and event do not match.	The resource and event do not match.	Check whether the resource and event parameters in the request meet the requirements in the API reference.
400	IOTDA.010000	Invalid input for this rule.	Invalid input for this rule.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.010001	The rule name already exists.	The rule name already exists.	Use another rule name and try again.
400	IOTDA.010004	Invalid parameter in the rule condition.	Invalid parameter in the rule condition.	Ensure that the parameters related to SQL statements in the request are correct.
400	IOTDA.010005	Invalid parameter in the rule action.	Invalid parameter in the rule action.	Check whether the action parameter in the request meets the requirements in the Huawei Cloud documentation.
400	IOTDA.010006	Duplicate rule condition ID.	Duplicate rule condition ID.	Change the rule condition ID and try again.
400	IOTDA.010007	Duplicate rule action ID.	Duplicate rule action ID.	Change the rule action ID and try again.
400	IOTDA.010008	The device in the rule condition does not exist.	The device in the rule condition does not exist.	Check whether the request parameters are correct or whether the device exists on the platform.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.010009	The device in the rule action does not exist.	The device in the rule action does not exist.	Check whether the request parameters are correct or whether the device exists on the platform.
400	IOTDA.010010	The device information in the rule condition does not exist.	The device information in the rule condition does not exist.	Check whether the request parameters are correct or whether the device exists on the platform.
400	IOTDA.010011	The device information in the rule action does not exist.	The device information in the rule action does not exist.	Check whether the request parameters are correct or whether the device exists on the platform.
400	IOTDA.010012	The tag in the rule condition does not exist.	The tag in the rule condition does not exist.	Check whether the request parameters are correct or whether the tag exists on the platform.
400	IOTDA.010013	Invalid rule parameter.	Invalid rule parameter.	Verify parameters by referring to "Request Parameters".
400	IOTDA.010014	Invalid input. The rule action of the DEVICE_ALARM type can be created only in the condition of the DEVICE_DATA type.	Invalid input. The rule action of the DEVICE_ALARM type can be created only in the condition of the DEVICE_DATA type.	When creating a rule action of the DEVICE_ALARM type, set the rule condition type to the DEVICE_DATA rule condition.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.010015	Max rules (10) reached.	Max rules (10) reached.	Delete unnecessary rules and try again.
400	IOTDA.010016	Invalid input. Only one DEVICE_ALARM rule can be created.	Invalid input. Only one DEVICE_ALARM rule can be created.	The DEVICE_ALARM rule already exists and does not need to be created again.
400	IOTDA.010017	Invalid input. The rule condition of the DEVICE_DATA type cannot contain both product_id and device_id.	Invalid input. The rule condition of the DEVICE_DATA type cannot contain both product_id and device_id.	Ensure that either device_id or product_id in the rule condition of the DEVICE_DATA type is specified.
400	IOTDA.010018	Invalid input. Both device_id and product_id in the rule condition of the DEVICE_DATA type are empty.	Invalid input. Both device_id and product_id in the rule condition of the DEVICE_DATA type are empty.	Ensure that either device_id or product_id in the rule condition of the DEVICE_DATA type is specified.
400	IOTDA.010019	The rule with the same condition already exists.	The rule with the same condition already exists.	The rule already exists and does not need to be created again.
400	IOTDA.010021	Invalid app_id.	Invalid app_id.	Check whether the app_id parameter in the request is correct.
400	IOTDA.010022	The rule has no action and cannot be enabled.	The rule has no action and cannot be enabled.	Use the API for modifying a rule to add an action to the rule.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.010023	Duplicate channeldetail in the rule action.	Duplicate channeldetail in the rule action.	The rule already exists. Do not register it again or delete unnecessary rules and try again.
400	IOTDA.010024	Invalid input. Invalid address.	Invalid input. Invalid address.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.010025	Invalid input. Invalid username or password.	Invalid input. Invalid username or password.	Check whether the username and password in the request meet the requirements in the Huawei Cloud documentation.
400	IOTDA.010026	Invalid input. The streamId or streamName is empty.	Invalid input. The streamId or streamName is empty.	Check whether the stream ID and stream name in the request meet the requirements in the Huawei Cloud documentation.
400	IOTDA.010027	Invalid input. Failed to query the channel.	Invalid input. Failed to query the channel.	Check whether the request parameters are consistent with the actual cloud service product parameters.
400	IOTDA.010028	Invalid input. Invalid log_group_id or log_stream_id.	Invalid input. Invalid log_group_id or log_stream_id.	Check whether log_group_id and log_stream_id in the request meet the requirements in the Huawei Cloud documentation.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.010029	Invalid input. Invalid func_urn.	Invalid input. Invalid func_urn.	Check whether the func_urn parameter in the request meets the requirements in the Huawei Cloud documentation.
400	IOTDA.010030	Invalid input. Connect to the database failed.	Invalid input. Connect to the database failed.	Check whether the database connection parameter in the request meets the requirements in the Huawei Cloud documentation.
400	IOTDA.010031	Invalid input. The table name does not exist.	Invalid input. The table name does not exist.	Check whether the table_name parameter in the request meets the requirements in the Huawei Cloud documentation.
400	IOTDA.010032	Invalid input. The suffix of the krb_file file is .conf, and the suffix of the keytab_file file is .keytab in the request.	Invalid input. The suffix of the krb_file file is .conf, and the suffix of the keytab_file file is .keytab in the request.	Check whether krb_file and keytab_file in the request meet the requirements in the Huawei Cloud documentation.
400	IOTDA.010033	Invalid input. The credential file does not exist.	Invalid input. The credential file does not exist.	Check whether the configuration credential file meets the requirements in the Huawei Cloud documentation.
400	IOTDA.010034	Invalid input. Connect to the cloud service failed.	Invalid input. Connect to the cloud service failed.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.010035	Invalid input. Max time range (24 hours) exceeded.	Invalid input. Max time range (24 hours) exceeded.	Change the parameters and try again.
400	IOTDA.010036	Invalid input. The query time dimension is invalid.	Invalid input. The query time dimension is invalid.	Change the parameters and try again.
400	IOTDA.010038	Invalid input. Connect to database failed due to invalid database info.	Invalid input. Connect to database failed due to invalid database info.	Check whether the forwarding database information in the request meets the requirements in the Huawei Cloud documentation.
400	IOTDA.010039	Invalid input. Repeated column in the action database.	Invalid input. Repeated column in the action database.	Check whether the forwarding parameters in the storage configuration meet the requirements specified in the Huawei Cloud documentation.
400	IOTDA.010040	Invalid input. Column in the action does not match that in the database.	Invalid input. Column in the action does not match that in the database.	Check whether the target storage parameter in the data forwarding configuration exists in the forwarding database.
400	IOTDA.010042	The number of rules in a project has reached the upper limit.	The number of rules in a project has reached the upper limit.	Delete unnecessary rules and try again.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.010044	Invalid input. The stack policy already exists.	Invalid input. The stack policy already exists.	The stack policy configuration already exists. You do not need to create it again. Delete the configuration and try again.
400	IOTDA.010045	Invalid input. The %s parameter is invalid.	Invalid input. The %s parameter is invalid.	Check whether the stack policy configuration parameters meet the Huawei Cloud console requirements.
400	IOTDA.010047	Invalid input. The flow control policy is duplicated.	Invalid input. The flow control policy is duplicated.	The flow control policy already exists. You do not need to create it again. Delete it and try again.
400	IOTDA.010048	Invalid input. The %s parameter is invalid.	Invalid input. The %s parameter is invalid.	Check whether the flow control policy configuration meets the Huawei Cloud console requirements.
400	IOTDA.010050	Invalid input. Invalid address or the topic does not exist.	Invalid input. Invalid address or the topic does not exist.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.013000	The product does not exist or does not belong to the application.	The product does not exist or does not belong to the application.	heck whether the request parameters are correct or whether the product has been registered in the specified resource space. If no resource space is specified, products in the default resource space are searched.
400	IOTDA.013001	The serviceType of the product does not exist.	The serviceType of the product does not exist.	Check whether the product has a service type. If the product does not have a service type, call the API for modifying a product to add a service type.
400	IOTDA.013002	The properties of deviceService Capability do not exist.	The properties of deviceService Capability do not exist.	Check whether the product has properties. If the product does not have properties, call the API for modifying a product to add properties.
400	IOTDA.013003	Operation not allowed. The product is unavailable.	Operation not allowed. The product is unavailable.	Check whether the device has a product ID. If the device does not have a product ID, call the API for modifying a device to add a product ID.
400	IOTDA.013005	The productName has been used in the same application.	The productName has been used in the same application.	Change the product name and try again.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.013008	The product ID has been used in the same application.	The product ID has been used in the same application.	Change the product ID and try again.
400	IOTDA.013010	Invalid input. The content in the product service capability is duplicate. Check the content %s.	Invalid input. The content in the product service capability is duplicate. Check the content %s.	The product model contains duplicate naming information.
400	IOTDA.013012	Invalid input. The size of product profile content has reached or exceeded limit.	Invalid input. The size of product profile content has reached or exceeded %s.	Keep the defined product model content within the specified range.
400	IOTDA.013015	Invalid input. The number of content items %s in the product has reached the limit %s.	Invalid input. The number of content items %s in the product has reached the limit %s.	Check the number of fields in the product.
400	IOTDA.013501	Invalid input. Invalid topic_short_name. If the value of operation_type is not SUBSCRIBE, the value of topic_short_name cannot contain the number sign (#) or plus sign (+).	Invalid input. Invalid topic_short_name. If the value of operation_type is not SUBSCRIBE, the value of topic_short_name cannot contain the number sign (#) or plus sign (+).	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.013502	Invalid input. The topic_short_name is duplicated under the same product.	Invalid input. The topic_short_name is duplicated under the same product.	Change the topic_short_name value and try again.
400	IOTDA.014001	Invalid input. The externalId parameter already exists.	Invalid input. The externalId parameter already exists.	Change the value of the externalId parameter and try again.
400	IOTDA.014002	Invalid input. The type of externalId must be String.	Invalid input. The type of externalId must be String.	Change the type of the externalId parameter to String.
400	IOTDA.014008	Invalid input. Duplicated nodeId.	Invalid input. Duplicated nodeId.	Change the node ID and try again.
400	IOTDA.014009	Invalid input. Duplicated deviceName.	Invalid input. Duplicated deviceName.	Change the device name and try again.
400	IOTDA.014010	Invalid input. The secretDevice cannot be empty when authType is SECRET.	Invalid input. The secretDevice cannot be empty when authType is SECRET.	Modify the parameters and try again.
400	IOTDA.014011	Invalid input. The secretEncryptionType cannot be empty when authType is MQTT.	Invalid input. The secretEncryptionType cannot be empty when authType is MQTT.	Contact Huawei technical support engineers.
400	IOTDA.014012	Invalid input. The pskDevice cannot be empty when authType is PSK.	Invalid input. The pskDevice cannot be empty when authType is PSK.	Change the parameters and try again.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014013	Invalid input. The SecureAccess (isSecure) must be true.	Invalid input. The SecureAccess (isSecure) must be true.	Set isSecure to true.
400	IOTDA.014017	Invalid input. The serviceType does not exist.	Invalid input. The serviceType does not exist.	Check whether the value of serviceType is the same as the value of serviceType of the product.
400	IOTDA.014023	Gateway and sensor authentication types are inconsistent.	Gateway and sensor authentication types are inconsistent.	Ensure that the authentication type of the gateway is the same as that of the child device.
400	IOTDA.014025	Invalid input. Invalid serviceId.	Invalid input. Invalid serviceId.	Check whether the service ID in the request meets the requirements in the Huawei Cloud documentation.
400	IOTDA.014027	Invalid input. The initialization cannot be empty when mode is INITIALIZATION.	Invalid input. The initialization cannot be empty when mode is INITIALIZATION.	Contact Huawei technical support engineers.
400	IOTDA.014028	Invalid input. The gateway is not online.	Invalid input. The gateway is not online.	Try again after the gateway goes online.
400	IOTDA.014031	Invalid input. The device already exists.	Invalid input. The device already exists.	The device ID already exists,Change the parameter, such as the device ID, and try again.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014033	Invalid input. When the product protocol is CoAP, the password must be in hexadecimal format.	Invalid input. When the product protocol is CoAP, the password must be in hexadecimal format.	Change the password to a hexadecimal character string.
400	IOTDA.014034	Invalid input. The serviceId or eventType does not match.	Invalid input. The serviceId or eventType does not match.	Check whether the values of service_id and event_type in the request are the same as those defined in the product model.
400	IOTDA.014035	Invalid input. The size of extension_info has reached or exceeded 1 KB.	Invalid input. The size of extension_info has reached or exceeded 1 KB.	Ensure that the value size of extension_info is not greater than 1 KB.
400	IOTDA.014043	Invalid input. The gateway does not exist.	Invalid input. The gateway does not exist.	Check whether the gateway_id parameter in the request has been registered with the platform.
400	IOTDA.014051	Invalid input. The device does not exist or does not belong to the application.	Invalid input. The device does not exist or does not belong to the application.	Check whether the request parameters are correct and whether the device has been registered with the application.
400	IOTDA.014100	Invalid command status.	Invalid command status.	Contact Huawei technical support engineers.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014104	The device command cannot be canceled because it has been canceled, executed, or expired.	The device command cannot be canceled because it has been canceled, executed, or expired.	Contact Huawei technical support.
400	IOTDA.014105	Invalid parameter 'mode'.	Invalid parameter 'mode'.	Check whether the value of mode is correct. The value can only be null, PASSIVE, or ACTIVE.
400	IOTDA.014107	Invalid input. Invalid parameter 'lifeCycle'.	Invalid input. Invalid parameter 'lifeCycle'.	Contact Huawei technical support engineers.
400	IOTDA.014108	Invalid parameter 'command_name'.	Invalid parameter 'command_name'.	Check whether the values of service_id and command_name are the same as those in the product model.
400	IOTDA.014110	Invalid input. The format of parameter 'commandBody' is not JSON.	Invalid input. The format of parameter 'commandBody' is not JSON.	Ensure that the paras parameter in the request is in JSON format.
400	IOTDA.014112	Send to device failed because the device does not subscribe to the topic.	Send to device failed because the device does not subscribe to the topic.	Check whether the device has subscribed to the topic.
400	IOTDA.014113	Invalid input. The size of paras has exceeded the upper limit.	Invalid input. The size of paras has exceeded the upper limit.	Reduce the length of the paras parameter value in the request.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014114	The API does not support MQTT devices.	The API does not support MQTT devices.	Replace the device with an NB-IoT device or use the API that supports MQTT.
400	IOTDA.014115	The API does not support NB-IoT devices.	The API does not support NB-IoT devices.	Use the API for delivering asynchronous device commands for NB-IoT devices.
400	IOTDA.014116	Invalid input. The size of request body has exceeded the upper limit.	Invalid input. The size of request body has exceeded the upper limit.	Reduce the size of request body.
400	IOTDA.014130	Invalid input. The messageId of the device is not unique.	Invalid input. The messageId of the device is not unique.	The message_id must be unique. Modify the message_id parameter.
400	IOTDA.014150	Invalid input. The topic has no permission.	Invalid input. The topic has no permission.	Check the topic parameter and ensure that the input value is correct and you have the subscription permission or all permissions on the topic in the product.
400	IOTDA.014151	Invalid input. The topic-related parameters in the request are duplicate.	Invalid input. The topic-related parameters in the request are duplicate.	Ensure that either topic or topic_full_name is specified in the request body.
400	IOTDA.014201	Invalid input. The batch task name already exists.	Invalid input. The batch task name already exists.	Change the task name and try again.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014203	Invalid input. The document parameter is invalid. errorMsg : %s.	Invalid input. The document parameter is invalid. errorMsg : %s.	Check the document parameter by following the instructions in "Request Parameters" in **Create a Batch Task**.
400	IOTDA.014204	Invalid input. The targets and targets_filter cannot both be empty.	Invalid input. The targets and targets_filter cannot both be empty.	Specify any one of the parameters.
400	IOTDA.014205	Invalid input. The key of targets_filter only supports %s.	Invalid input. The key of targets_filter only supports %s.	Check the targets_filter parameter by following the instructions in "Request Parameters" in **Create a Batch Task**.
400	IOTDA.014207	Invalid input. The start time cannot be earlier than the current time, and the latest start time cannot exceed %s days.	Invalid input. The start time cannot be earlier than the current time, and the latest start time cannot exceed %s days.	Check the schedule_time parameter by following the instructions in "Request Parameters" in **Create a Batch Task**.
400	IOTDA.014208	Invalid input. retry_count and retry_interval depend on each other and must be assigned at the same time.	Invalid input. retry_count and retry_interval depend on each other and must be assigned at the same time.	Specify the retry_count and retry_interval parameters.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014209	Invalid input. The task cannot be stopped because it is completed or being stopped.	Invalid input. The task cannot be stopped because it is completed or being stopped.	Check whether the task is completed or being stopped.
400	IOTDA.014210	Invalid input. The parameter 'targets_filter' does not support multiple keys.	Invalid input. The parameter 'targets_filter' does not support multiple keys.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.014216	Invalid input. The task cannot retry because the task has succeeded, stopped, being stopped or is waiting.	Invalid input. The task cannot retry because the task has succeeded, stopped, being stopped or is waiting.	Check whether the task has succeeded, stopped, being stopped, or is waiting.
400	IOTDA.014219	Invalid input. The target is not in the task.	Invalid input. The target is not in the task.	Check whether The target is not in the task.
400	IOTDA.014300	Operation not allowed. The number of certificates has reached the upper limit (%s).	Operation not allowed. The number of certificates has reached the upper limit (%s).	Delete unnecessary certificates and try again. The maximum number of certificates for a tenant is 100.
400	IOTDA.014301	Invalid certificate content.	Invalid certificate content.	Check whether the certificate content meets the X.509 standard. For details about how to parse a certificate, visit **parse a certificate**.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014302	Invalid input. The certificate (certificate_id: %s) already exists.	Invalid input. The certificate (certificate_id: %s) already exists.	You have uploaded a certificate with the same content. Do not upload it again.
400	IOTDA.014303	Operation not allowed. Upload certificate failed.	Operation not allowed. Upload certificate failed.	Contact Huawei technical support engineers.
400	IOTDA.014304	Operation not allowed. Delete certificate failed.	Operation not allowed. Delete certificate failed.	Contact Huawei technical support engineers.
400	IOTDA.014305	Operation not allowed. Query certificate failed.	Operation not allowed. Query certificate failed.	Contact Huawei technical support engineers.
400	IOTDA.014307	Operation not allowed. The certificate failed to verify the verifyCode.	Operation not allowed. The certificate failed to verify the verifyCode.	The certificate cn_name is inconsistent with the verification code of the current CA certificate. Check the certificate and generate a new certificate for verification.
400	IOTDA.014308	Operation not allowed. The certificate failed to verify the path.	Operation not allowed. The certificate failed to verify the path.	The certificate path is inconsistent with the path of the current CA certificate. Ensure that the two certificates are issued by the same CA.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014309	The certificate is expired or about to expire. The expiry date must be seven days later than current date.	The certificate is expired or about to expire. The expiry date must be seven days later than current date.	Apply for a valid certificate.
400	IOTDA.014310	The size of the certificate file cannot be larger than 1 MB and the file name must match the pattern ^[.a-zA-Z0-9_-]{1,255}$.	The size of the certificate file cannot be larger than 1 MB and the file name must match the pattern ^[.a-zA-Z0-9_-]{1,255}$.	Upload a certificate that meets the requirements.
400	IOTDA.014311	The certificate is in use and cannot be deleted.	The certificate is in use and cannot be deleted.	Disassociate the certificate and then delete it.
400	IOTDA.014312	Invalid input. The certificate scene does not exist.	Invalid input. The certificate scene does not exist.	Check whether the request parameters are correct.
400	IOTDA.014313	Invalid input. The certificate scene already exists.	Invalid input. The certificate scene already exists.	Check whether the request parameters are correct.
400	IOTDA.014314	Invalid input. The server certificate does not exist.	Invalid input. The server certificate does not exist.	Check whether the request parameters are correct.
400	IOTDA.014315	Invalid input. The CA certificate does not exist.	Invalid input. The CA certificate does not exist.	Check whether the request parameters are correct.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014316	Invalid input. Failed to parse the private key and certificate.	Invalid input. Failed to parse the private key and certificate.	Check whether the private key and certificate are valid and match with each other.
400	IOTDA.014318	Invalid input. The CA certificate not bind provisioning-template.	Invalid input. The CA certificate not bind provisioning-template.	Bind the template and try again.
400	IOTDA.014320	Invalid input. The provisioning-template is not exist.	Invalid input. The provisioning-template is not exist.	Check whether the template exists and try again.
400	IOTDA.014322	Invalid input. The same CA certificate has been bound provisioning-template in another application.	Invalid input. The same CA certificate has been bound provisioning-template in another application.	Unbind certificates template from other applications and try again.
400	IOTDA.014602	Invalid input. The batch task file name already exists.	Invalid input. The batch task file name already exists.	Change the file name and try again.
400	IOTDA.014603	Invalid input. The size of the batch task file exceeds the upper limit.	Invalid input. The size of the batch task file exceeds the upper limit.	The maximum size of a batch task file is 4 MB.
400	IOTDA.014604	Invalid input. The number of lines in the batch task file exceeds the upper limit.	Invalid input. The number of lines in the batch task file exceeds the upper limit.	The maximum number of lines in a batch task file is 30,000.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.014605	Invalid input. The resource-suffix of the batch task file is wrong.	Invalid input. The resource-suffix of the batch task file is wrong.	Check the file parameter by following the instructions in "Request Parameters" in **Upload a Batch Task File**.
400	IOTDA.014606	Invalid input. Invalid batch task file name.	Invalid input. Invalid batch task file name.	Check the file parameter by following the instructions in "Request Parameters" in **Upload a Batch Task File**.
400	IOTDA.014608	Invalid input. The content of batch task file is invalid.	Invalid input. The content of batch task file is invalid.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.014900	Invalid input. The search SQL contains unknown field.%s.	Invalid input. The search SQL contains unknown field.%s.	Please check the search SQL statement, correct invalid fields, and try again.
400	IOTDA.014901	Invalid input.The SQL statement contains invalid parameters.%s.	Invalid input.The SQL statement contains invalid parameters.%s.	Please check the search SQL statement, correct invalid parameters, and try again.
400	IOTDA.014902	Invalid input.The search SQL is invalid.%s.	Invalid input.The search SQL is invalid.%s.	Please check the search SQL statement, use the correct syntax in the reference document, and try again.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.015100	Invalid input. proxy_name is already exists.	Invalid input. proxy_name is already exists.	Change the parameter, such as the proxy_name, and try again.
400	IOTDA.015200	Invalid input. policy_name is already exists.	Invalid input. policy_name is already exists.	Change the parameter, such as the policy_name, and try again.
400	IOTDA.015203	Invalid input. The bind target not exist or does not belong to the application.	Invalid input. The bind target not exist or does not belong to the application.	Check whether the request parameters are correct and whether the bind target exist in the application.
400	IOTDA.015205	Invalid input. The target has been bound to the device-policy.	Invalid input. The target has been bound to the device-policy.	The target has been bound to the device-policy, no need to bind again.
400	IOTDA.015206	Invalid input. The device-policy id does not exist.	Invalid input. The device-policy id does not exist.	The device-policy ID does not exist. Check whether the parameter is correct.
400	IOTDA.015300	Invalid input. template_name is already exists.	Invalid input. template_name is already exists.	Change the parameter, such as the template_name, and try again.
400	IOTDA.015304	Invalid input. The parameter defined in the template does not exist.	Invalid input. The parameter defined in the template does not exist.	Check whether the device certificate information is complete. and try again.
400	IOTDA.015305	Invalid input. The field [template_body.resources.device.product_id] does not exist.	Invalid input. The field [template_body.resources.device.product_id] does not exist.	Check whether the request parameters are correct and whether the product id exist.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.016000	Invalid input, The queue name already exist in the same spUserName.	Invalid input, The queue name already exist in the same spUserName.	Change the queue name and try again.
400	IOTDA.016004	Invalid queue name.	Invalid queue name.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.016005	Invalid AMQP access configuration.	Invalid AMQP access configuration.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.016006	Invalid HTTP access configuration.	Invalid HTTP access configuration.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.016008	The integration configuration already exists.	The integration configuration already exists.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
400	IOTDA.019304	Invalid input. Invalid obs info.	Invalid input. Invalid obs info.	Check whether the request parameters region_name, bucket_name, object_key are correct.

Status Code	Error Codes	Error Message	Description	Solution
400	IOTDA.019305	Invalid input. The format of the upgrade package is incorrect.	Invalid input. The format of the upgrade package is incorrect.	Please refer to the upgrade package format restrictions in the HUAWEI CLOUD document **Software/ Firmware Package Upload**.
400	IOTDA.019306	Invalid input. The upgrade package size exceeds the limit.	Invalid input. The upgrade package size exceeds the limit.	Please refer to the upgrade package size limit in the HUAWEI CLOUD document **Software/ Firmware Package Upload**.
400	IOTDA.019600	Invalid input. The bridge name is duplicated.	Invalid input. The bridge name is duplicated.	Check whether a bridge with the same name has been created.
400	IOTDA.019603	Invalid input. The bridge ID is duplicated.	Invalid input. The bridge ID is duplicated.	Check whether a bridge with the same ID has been created.
400	IOTDA.019702	The tunnel has already closed.	The tunnel has already closed.	The tunnel has been closed. Create a tunnel again.
400	IOTDA.019708	Failed to deliver tunnel information to the device.	Failed to deliver tunnel information to the device.	Ensure that the device is running properly and has subscribed to the topic for receiving tunnel notifications.
400	IOTDA.021304	The number of device log configurations exceeds the upper limit.	The number of device log configurations exceeds the upper limit.	Disable log collection for some devices and then enable it again.
400	IOTDA.021310	Invalid input. The authorizer name is duplicated.	Invalid input. The authorizer name is duplicated.	Check whether a customized authorizer with the same name has been created.

Status Code	Error Codes	Error Message	Description	Solution
401	IOTDA.000002	Authentication failed.	Authentication failed.	Check whether the authentication parameters carried in the request are correct.
401	IOTDA.000025	SP user authentication failed.	SP user authentication failed.	Check whether the SP token is correct.
401	IOTDA.000026	Stage user authentication failed.	Stage user authentication failed.	Check whether the stage token is correct.
401	IOTDA.001003	Incorrect AppId or secret.	Incorrect AppId or secret.	Check whether the secret is correct.
401	IOTDA.014032	Invalid input. The time does not match.	Invalid input. The time does not match.	Contact Huawei technical support engineers.
401	IOTDA.019704	Invalid tunnel access token.	Invalid tunnel access token.	Check whether the request parameters are correct.
401	IOTDA.019705	The tunnel access token is expired.	The tunnel access token is expired.	Create a tunnel and try again.
403	IOTDA.000004	Invalid access token.	Invalid access token.	Check whether the token in the request is valid.
403	IOTDA.000005	Refresh access token failed.	Refresh access token failed.	Check whether the value of refreshToken in the request is correct.
403	IOTDA.000015	The account is frozen.	The account is frozen.	Contact the account owner to unfreeze the account.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.000021	Operation not allowed. User not found by IAM token or the authorized user has not subscribed to IoTDA.	Operation not allowed. User not found by IAM token or the authorized user has not subscribed to IoTDA.	Check whether the user to which the IAM token belongs has subscribed to IoTDA.
403	IOTDA.000022	Operation not allowed. The user does not have the permission.	Operation not allowed. The user does not have the permission.	Check whether you have the required permissions.
403	IOTDA.000023	Request frequency reached the upper limit %s.	Request frequency reached the upper limit %s.	Reduce the request frequency.
403	IOTDA.000024	Operation not allowed. Only one token is allowed in the request header.	Operation not allowed. Only one token is allowed in the request header.	Delete unnecessary tokens from the header.
403	IOTDA.000028	System is being maintained. The configuration cannot be modified.	System is being maintained. The configuration cannot be modified.	Try again later.
403	IOTDA.000033	Operation not allowed. The current instance does not support the parameter.	Operation not allowed. The current instance does not support the parameter.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
403	IOTDA.000034	Operation not allowed. The user does not have the permission.	Operation not allowed. The user does not have the permission.	Check whether you have the required permissions.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.001000	The application does not exist.	The application does not exist.	Check whether the resource space has been created on the platform and whether the resource space ID is correct.
403	IOTDA.001002	Operation not allowed. You do not have the permissions required to access the application.	Operation not allowed. You do not have the permissions required to access the application.	Check whether you have been authorized to access the application.
403	IOTDA.001005	Operation not allowed. The parameter 'app_id' is not carried, and the authorized user has more than one application. Include the parameter 'app_id', or contact Huawei technical support engineers to merge application data.	Operation not allowed. The parameter 'app_id' is not carried, and the authorized user has more than one application. Include the parameter 'app_id', or contact Huawei technical support engineers to merge application data.	Carry the corresponding app_id or contact Huawei engineers to combine application data.
403	IOTDA.001006	Operation not allowed. Application not found by authorized user or the authorized user has no application.	Operation not allowed. Application not found by authorized user or the authorized user has no application.	Check whether you have a resource space or whether there is a specified resource space.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.001007	Operation not allowed. The application does not belong to the authorized user.	Operation not allowed. The application does not belong to the authorized user.	Check whether you have the specified resource space.
403	IOTDA.001008	Operation not allowed. The app name already exists.	Operation not allowed. The app name already exists.	Change a resource space name.
403	IOTDA.001009	Operation not allowed. The maximum number of applications has been reached.	Operation not allowed. The maximum number of applications has been reached.	Delete unnecessary resource spaces and try again.
403	IOTDA.001010	Operation not allowed. The default app cannot be deleted.	Operation not allowed. The default app cannot be deleted.	Change the parameters and try again.
403	IOTDA.001014	Operation not allowed. The application does not belong to the authorized instance.	Operation not allowed. The application does not belong to the authorized instance.	Check whether the specified resource space exists in the instance.
403	IOTDA.003002	Operation not allowed. The group name already exists.	Operation not allowed. The group name already exists.	Change the device group name and try again.
403	IOTDA.003003	Operation not allowed. Max groups (1,000) reached.	Operation not allowed. Max groups (1,000) reached.	Delete unnecessary device groups and try again.
403	IOTDA.003004	Operation not allowed. Max group depths (5) reached.	Operation not allowed. Max group depths (5) reached.	Use the ID of a parent device group with a higher level for registration.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.003005	Operation not allowed. The device already exists in the group.	Operation not allowed. The device already exists in the group.	The device already exists in the device group and does not need to be added again.
403	IOTDA.003006	Operation not allowed. The device does not exist in the group.	Operation not allowed. The device does not exist in the group.	The device does not exist in the device group and does not need to be deleted.
403	IOTDA.003007	Operation not allowed. Max devices (20,000) reached for the group.	Operation not allowed. Max devices (20,000) reached for the group.	Delete unnecessary devices or replace the group.
403	IOTDA.003008	Operation not allowed. A device can be added to up to 10 groups.	Operation not allowed. A device can be added to up to 10 groups.	Remove the device from other groups and try again. The number of groups to which the device belongs has reached the maximum.
403	IOTDA.003009	Operation not allowed. The parent group contains child groups. Delete the child groups and try again.	Operation not allowed. The parent group contains child groups. Delete the child groups and try again.	Delete all child device groups in the group and try again.
403	IOTDA.003010	Operation not allowed. The group contains devices and cannot be deleted.	Operation not allowed. The group contains devices and cannot be deleted.	Delete all devices in the device group and try again.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.003011	Operation not allowed. The group and device do not belong to the same application.	Operation not allowed. The group and device do not belong to the same application.	Use the group and device of the same application.
403	IOTDA.004001	Operation not allowed. Max bindable tags (10) reached for the device.	Operation not allowed. Max bindable tags (10) reached for the device.	Delete unnecessary tags and try again. The number of tags bound to the device has reached the upper limit.
403	IOTDA.005000	Operation not allowed. Only one task can be started for a device at a time.	Operation not allowed. Only one task can be started for a device at a time.	Manually stop the ongoing task or start another task after the ongoing task is complete.
403	IOTDA.005001	Operation not allowed. The protocol of the device does not support upgrade.	Operation not allowed. The protocol of the device does not support upgrade.	**submit a service ticket** and contact Huawei technical support engineers.
403	IOTDA.009001	Max application subscription records reached.	Max application subscription records reached.	Delete unnecessary subscription records.
403	IOTDA.009008	The maximum number of queries has been reached.	The maximum number of queries has been reached.	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
403	IOTDA.010003	The number of rules has reached the upper limit.	The number of rules has reached the upper limit.	Delete unnecessary rules and try again.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.010037	The rule cannot be deleted because actions exist in the rule.	The rule cannot be deleted because actions exist in the rule.	Delete the rule action before deleting the rule.
403	IOTDA.010049	Operation not allowed. Total number of flow control policies exceeds the upper limit (4).	Operation not allowed. Total number of flow control policies exceeds the upper limit (4).	Delete unnecessary flow control policies and try again.
403	IOTDA.013004	Operation not allowed. You have no write permission.	Operation not allowed. You have no write permission.	Check whether the product property is writable. If not, call the API for modifying a product to change the property to writable.
403	IOTDA.013006	The number of products in the application has reached the upper limit.	The number of products in the application has reached the upper limit.	Delete unnecessary products and try again.
403	IOTDA.013007	Operation not allowed. The product is in use and cannot be deleted.	Operation not allowed. The product is in use and cannot be deleted.	Before deleting the product, ensure that no device is under the product, and then try again.
403	IOTDA.013009	Operation not allowed. The value of default_value must be writable.	Operation not allowed. The value of default_value must be writable.	Set the property to writable by calling the API for modifying a product.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.013011	Operation not allowed. The number of assets properties has reached the upper limit (%s).	Operation not allowed. The number of assets properties has reached the upper limit (%s).	Delete unnecessary asset property definitions.
403	IOTDA.013013	Operation not allowed. The protocol type cannot be changed to CoAP.	Operation not allowed. The protocol type cannot be changed to CoAP.	Create a CoAP product again.
403	IOTDA.013503	Operation not allowed. The number of topics in a product exceeds the upper limit (%s).	Operation not allowed. The number of topics in a product exceeds the upper limit (%s).	Delete unnecessary topics and try again.
403	IOTDA.014003	Operation not allowed. The number of frozen devices has reached the upper limit.	Operation not allowed. The number of frozen devices has reached the upper limit.	Unfreeze the devices that do not need to be frozen or contact Huawei engineers.
403	IOTDA.014004	Operation not allowed. This device is online.	Operation not allowed. This device is online.	Set the device status to offline and try again.
403	IOTDA.014005	Operation not allowed. This device is not active.	Operation not allowed. This device is not active.	Activate the device by following the instructions.
403	IOTDA.014006	Operation not allowed. The secret cannot be reset.	Operation not allowed. The secret cannot be reset.	NB-IoT devices do not use secrets. You do not need to reset the secret.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.014007	Operation not allowed. The PSK cannot be reset.	Operation not allowed. The PSK cannot be reset.	Change the parameters and try again.
403	IOTDA.014015	Operation not allowed. The number of concurrent location service requests has exceeded the upper limit.	Operation not allowed. The number of concurrent location service requests has exceeded the upper limit.	Reduce the API calling frequency.
403	IOTDA.014016	Operation not allowed. The device is not online.	Operation not allowed. The device is not online.	Try again after the device goes online.
403	IOTDA.014018	Operation not allowed. The device has been frozen and cannot be operated.	Operation not allowed. The device has been frozen and cannot be operated.	**Unfreeze the device** and try again.
403	IOTDA.014019	Operation not allowed. The number of devices of the user has reached the upper limit.	Operation not allowed. The number of devices of the user has reached the upper limit.	Delete unnecessary devices or contact Huawei technical support engineers.
403	IOTDA.014020	Operation not allowed. The number of devices bound to this application has reached the upper limit.	Operation not allowed. The number of devices bound to this application has reached the upper limit.	Delete unnecessary devices or contact Huawei technical support engineers.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.014021	Operation not allowed. The number of devices has reached the upper limit of the package.	Operation not allowed. The number of devices has reached the upper limit of the package.	Delete unnecessary devices or contact Huawei technical support engineers.
403	IOTDA.014022	Operation not allowed. The number of license resources has reached the upper limit.	Operation not allowed. The number of license resources has reached the upper limit.	Contact Huawei technical support engineers.
403	IOTDA.014024	Operation not allowed. The device already has an endpoint.	Operation not allowed. The device already has an endpoint.	To delete the gateway, ensure that the child devices under the gateway are no longer used or not associated with other gateways, delete the child devices first, and then try again.
403	IOTDA.014026	Operation not allowed. The number of non-security devices has reached the upper limit.	Operation not allowed. The number of non-security devices has reached the upper limit.	Delete unnecessary non-security devices or contact Huawei technical support engineers.
403	IOTDA.014029	Operation not allowed. The timeout parameter cannot be modified when the device has been activated.	Operation not allowed. The timeout parameter cannot be modified when the device has been activated.	Cancel the modification of the timeout parameter or perform the operation on a device that is not activated.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.014036	Operation not allowed. The device service capabilities are not defined in the product.	Operation not allowed. The device service capabilities are not defined in the product.	Use the API for modifying a product to add device service capabilities to the product.
403	IOTDA.014037	Operation not allowed. Max device depths (2 levels) reached.	Operation not allowed. Max device depths (2 levels) reached.	The gateway supports up to two levels of child devices.
403	IOTDA.014038	Operation not allowed. Only gateways or directly connected devices are supported.	Operation not allowed. Only gateways or directly connected devices are supported.	Select a directly connected device or gateway to operate.
403	IOTDA.014039	Operation not allowed. The device has already been frozen.	Operation not allowed. The device has already been frozen.	Unfreeze the device and try again.
403	IOTDA.014040	Operation not allowed. The device is not frozen and cannot be unfrozen.	Operation not allowed. The device is not frozen and cannot be unfrozen.	Only frozen devices can be unfrozen.
403	IOTDA.014041	Operation not allowed. The device does not belong to the currently connected gateway.	Operation not allowed. The device does not belong to the currently connected gateway.	Select a child device under the currently connected gateway.
403	IOTDA.014101	The number of commands reached the upper limit.	The number of commands reached the upper limit.	Try again after the pending commands are delivered or increase the queue for storing pending commands.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.014106	Invalid CommandBody for the MQTT protocol.	Invalid CommandBody for the MQTT protocol.	The command parameter must be in JSON format. Check whether the delivered parameter is correct.
403	IOTDA.014109	Operation not allowed. The command status is not 'PENDING'.	Operation not allowed. The command status is not 'PENDING'.	Contact Huawei technical support engineers.
403	IOTDA.014133	Operation not allowed. The topic has no 'SUBSCRIBE' permission.	Operation not allowed. The topic has no 'SUBSCRIBE' permission.	Contact Huawei technical support engineers.
403	IOTDA.014202	Operation not allowed. The number of unfinished tasks has reached the upper limit (%s).	Operation not allowed. The number of unfinished tasks has reached the upper limit (%s).	The number of batch tasks being executed has exceeded the upper limit. Stop unfinished tasks and try again.
403	IOTDA.014206	Operation not allowed. The number of targets has reached the upper limit (%s).	Operation not allowed. The number of targets has reached the upper limit (%s).	Check whether the request parameters meet the requirements in the Huawei Cloud documentation.
403	IOTDA.014217	Operation not allowed. The target cannot retry because the task of target has succeeded, is waiting or is processing.	Operation not allowed. The target cannot retry because the task of target has succeeded, is waiting or is processing.	Check whether the task of target has succeeded, is waiting or is processing.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.014218	Operation not allowed. The target cannot stop because the task of target has succeeded, failed or stopped.	Operation not allowed. The target cannot stop because the task of target has succeeded, failed or stopped.	Check whether the task of target has succeeded, failed or stopped.
403	IOTDA.014220	Operation not allowed. The task type is not supported.	Operation not allowed. The task type is not supported.	Check whether The task type is supported.
403	IOTDA.014221	Invalid input. The task cannot be deleted because it is not completed.	Invalid input. The task cannot be deleted because it is not completed.	Please check if the task status is Success, Fail, PartialSuccess or Stopped.
403	IOTDA.014319	Operation not allowed. The CA certificate state is not verified, please verify the certificate first.	Operation not allowed. The CA certificate state is not verified, please verify the certificate first.	Verify the CA certificate and try again.
403	IOTDA.014601	Operation not allowed. The number of batch task files of the current user exceeds the upper limit.	Operation not allowed. The number of batch task files of the current user exceeds the upper limit.	Delete unnecessary files and try again.
403	IOTDA.014607	Operation not allowed. The batch task file is in use and cannot be deleted.	Operation not allowed. The batch task file is in use and cannot be deleted.	Perform the operation after the task is complete.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.015101	Operation not allowed. The device already exists in the other device-proxy.	Operation not allowed. The device already exists in the other device-proxy.	Replace the proxy devices and try again.
403	IOTDA.015102	Operation not allowed. The endpoint device is not allowed to be added to device-proxy.	Operation not allowed. The endpoint device is not allowed to be added to device-proxy.	The endpoint device are not allowed to be added to device-proxy. Change the device, and try again.
403	IOTDA.015104	Operation not allowed. The number of device-proxy has reached the upper limit (10).	Operation not allowed. The number of device-proxy has reached the upper limit (10).	Delete unnecessary device-proxy and try again.
403	IOTDA.015201	Operation not allowed. The number of device-policy has reached the upper limit (%s).	Operation not allowed. The number of device-policy has reached the upper limit.	Delete unnecessary device-policy and try again.
403	IOTDA.015204	Operation not allowed. The number of policies bound to the target reaches the upper limit.	Operation not allowed. The number of policies bound to the target reaches the upper limit.	Unbind unnecessary device-policy from the target object and try again.
403	IOTDA.015303	Operation not allowed. The provisioning-template is associated by certificate.	Operation not allowed. The provisioning-template is associated by certificate.	Disassociate the device CA certificate and try again.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.015306	Operation not allowed. The number of provisioning-template has reached the upper limit (10).	Operation not allowed. The number of provisioning-template has reached the upper limit (10).	Delete unnecessary provisioning-template and try again.
403	IOTDA.016001	Operation not allowed, The number of queues exceeds the limit for each spUserName.	Operation not allowed, The number of queues exceeds the limit for each spUserName.	Each tenant can create up to 100 AMQP queues.
403	IOTDA.016003	Operation not allowed. The queue is in use.	Operation not allowed. The queue is in use.	Unsubscribe from the rules and try again.
403	IOTDA.016009	Operation not allowed. The number of queues exceeds the system limit.	Operation not allowed. The number of queues exceeds the system limit.	Contact Huawei technical support engineers.
403	IOTDA.019303	Operation not allowed. The number of packages for the user has reached the upper limit.	Operation not allowed. The number of packages for the user has reached the upper limit.	Delete unnecessary packages.
403	IOTDA.019601	Operation not allowed. The number of bridges in the system has reached the upper limit.	Operation not allowed. The number of bridges in the system has reached the upper limit.	Check whether the number of bridge under the spUser is 20.

Status Code	Error Codes	Error Message	Description	Solution
403	IOTDA.019701	Operation not allowed. The device does not belong to the authorized user.	Operation not allowed. The device does not belong to the authorized user.	Check whether the device ID is correct.
403	IOTDA.019703	The open tunnel cannot be deleted.	The open tunnel cannot be deleted.	Call the API **Disable a Device Tunnel** to disable the tunnel before deleting it.
403	IOTDA.019706	Operation not allowed. The number of tunnels for a device has reached the upper limit.	Operation not allowed. The number of tunnels for a device has reached the upper limit.	Delete old device tunnels and try again.
403	IOTDA.019707	Operation not allowed. The number of device tunnels for a user has reached the upper limit.	Operation not allowed. The number of device tunnels for a user has reached the upper limit.	Delete old device tunnels and try again.
403	IOTDA.021311	Operation not allowed. The number of authorizer has reached the upper limit(10).	Operation not allowed. The number of authorizer has reached the upper limit(10).	Check whether the number of customized authorizer under the spUser is 10.
403	IOTDA.021313	The function does not exist.	The function does not exist.	Check whether the function created in FunctionGraph.
404	IOTDA.000016	Target not found.	Target not found.	Contact Huawei technical support engineers.
404	IOTDA.000029	Invalid input. The path does not exist.	Invalid input. The path does not exist.	Check whether the request path is correct.

Status Code	Error Codes	Error Message	Description	Solution
404	IOTDA.003000	The group does not exist.	The group does not exist.	Check whether the device group exists or whether the device group parameters are correctly carried.
404	IOTDA.003001	The parent group does not exist.	The parent group does not exist.	Check whether the device group exists or whether the device group parameters are correctly carried.
404	IOTDA.004000	The resource does not exist.	The resource does not exist.	Check whether the request parameters, such as the resource ID, are correct.
404	IOTDA.009003	The subscription subject cannot be found.	The subscription subject cannot be found.	Check whether the parameters in the request match the subscription.
404	IOTDA.010002	The rule does not exist.	The rule does not exist.	Check whether the rule exists on the platform or whether the request parameters are correct.
404	IOTDA.010020	The rule action does not exist.	The rule action does not exist.	Check whether the rule action exists on the platform or whether the request parameters are correct.

Status Code	Error Codes	Error Message	Description	Solution
404	IOTDA.010043	The stack policy does not exist.	The stack policy does not exist.	Check whether the stack policy configuration exists on the platform or whether the request parameters are correct.
404	IOTDA.010046	The flow control policy does not exist.	The flow control policy does not exist.	Check whether the corresponding flow control policy configuration exists on the platform or whether the request parameters are correct.
404	IOTDA.013014	The product does not exist.	The product does not exist.	Check whether the request parameters are correct or whether the product has been registered with the platform.
404	IOTDA.013500	The topic does not exist.	The topic does not exist.	Contact Huawei technical support engineers.
404	IOTDA.014000	The device does not exist.	The device does not exist.	Check whether the request parameters are correct and whether the device has been registered with the platform.
404	IOTDA.014014	The IMSI does not exist.	The IMSI does not exist.	Specify the IMSI and try again.

Status Code	Error Codes	Error Message	Description	Solution
404	IOTDA.014042	The device access information does not exist.	The device access information does not exist.	Check whether the device is connected to the platform and activated.
404	IOTDA.014103	The device command does not exist.	The device command does not exist.	Check whether the request parameters are correct.
404	IOTDA.014131	The message does not exist.	The message does not exist.	Contact Huawei technical support engineers.
404	IOTDA.014200	The batch task does not exist.	The batch task does not exist.	Check whether task_id in the request is valid.
404	IOTDA.014306	The certificate does not exist.	The certificate does not exist.	The target certificate ID does not exist. Check whether the request parameter is correct.
404	IOTDA.014600	The batch task file does not exist.	The batch task file does not exist.	Check whether the request parameters are correct.
404	IOTDA.015105	device-proxy not exist or does not belong to the application.	device-proxy not exist or does not belong to the application.	Check whether the request parameters are correct and whether the device-proxy has been registered with the application.
404	IOTDA.015202	Invalid input. device-policy not exist or does not belong to the application.	Invalid input. device-policy not exist or does not belong to the application.	Check whether the request parameters are correct and whether the device-policy exist in the application.

Status Code	Error Codes	Error Message	Description	Solution
404	IOTDA.015301	provisioning-template does not exist.	provisioning-template does not exist.	Check whether the request parameters are correct and whether the provisioning-template exist.
404	IOTDA.016002	The queue ID does not exist.	The queue ID does not exist.	Check whether the request parameters are correct.
404	IOTDA.019300	The package does not exist.	The package does not exist.	Check whether package exists or the request parameters package_id are correct.
404	IOTDA.019602	Invalid input. The bridge does not exist.	Invalid input. The bridge does not exist.	Check whether the bridge is created.
404	IOTDA.019700	Invalid input. The tunnel does not exist.	Invalid input. The tunnel does not exist.	Check whether a tunnel is created and whether the tunnel ID is correct.
404	IOTDA.021312	The authorizer does not exist.	The authorizer does not exist.	Check whether the customized authorizer is created.
405	IOTDA.000003	The request method is not supported.	The request method is not supported.	Check whether the request mode is the same as that in the documentation.
405	IOTDA.000019	Method not allowed.	Method not allowed.	Check whether the request mode is the same as that in the documentation.

Status Code	Error Codes	Error Message	Description	Solution
406	IOTDA.000037	Invalid input. The Accept is missing or not supported in the request header.	Invalid input. The Accept is missing or not supported in the request header.	Check whether the Accept in the request header is valid.
409	IOTDA.014030	The version of the serviceId %s conflicts with another one.	The version of the serviceId %s conflicts with another one.	Use the correct version number.
409	IOTDA.014047	Device status update conflicts occur.	Device status update conflicts occur.	Check whether the request is sent repeatedly or the gateway is offline.
415	IOTDA.000036	Invalid input. The Content-Type is missing or not supported in the request header.	Invalid input. The Content-Type is missing or not supported in the request header.	Check whether the Content-Type in the request header is valid.
429	IOTDA.014102	Congestion occurs. The network is under flow control.	Congestion occurs. The network is under flow control.	Contact Huawei technical support engineers.
500	IOTDA.000001	Internal server error.	Internal server error.	Contact Huawei technical support engineers.
500	IOTDA.000007	The data in database is abnormal.	The data in database is abnormal.	Contact Huawei technical support engineers.
500	IOTDA.000020	Decrypt IAM token failed.	Decrypt IAM token failed.	Contact Huawei technical support engineers.

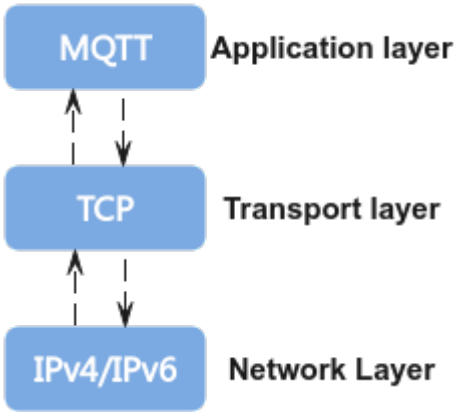
2 MQTT or MQTTS API Reference on the Device Side

2.1 Before You Start

Introduction

Message Queuing Telemetry Transport (MQTT) is a lightweight underlying protocol for publish-subscribe communication. It provides ordered, reliable, and bidirectional byte stream transmission and supports device-cloud message transfer. It can provide real-time and reliable messaging services for connecting to remote devices while requiring little code and limited bandwidth. As an instant messaging protocol with low overhead and low bandwidth usage, MQTT is widely used in IoT, small devices, and mobile applications.

Figure 2-1 MQTT



Data Packets

An MQTT message consists of fixed header, variable header, and payload. **IoT Device SDKs** are recommended for access.

For details about how to define the fixed header and variable header, see the **MQTT V3.1.1**. The payload is defined by applications, that is, by the devices and IoTDA.

- Fixed header: Each MQTT control packet contains a fixed header, which is used to determine the control packet type. In a PUBLISH message, it is also used to determine the maximum quality of service (QoS), such as link establishment, subscription, and release.
- Variable header: Some MQTT control packets contain a variable header, which is located between the fixed header and the payload. The contents of the variable header vary according to the packet type. A packet identifier contains a variable header, and is used to distinguish different data packets on the same link.
- Payload: Some MQTT control packets contain a payload at the end of the packet. For a PUBLISH message, the payload is an application message (defined by users).

NOTE

For details about MQTT syntax and APIs, see **MQTT Version V3.1.1**.

Control Packet Types

Common MQTT message types include CONNECT, SUBSCRIBE, PUBLISH, and PINGREQ.

- CONNECT: A client requests a connection to a server. For details about the parameters in the payload of a CONNECT message, see **2.4 Device Connection Authentication**.
- SUBSCRIBE: A SUBSCRIBE message includes topic filters and maximum QoS. A session can have multiple SUBSCRIBE messages. For details about subscription to the platform, see **2.3 Topics**.
- PUBLISH: An application message is transmitted from the client to the server or from the server to the client.

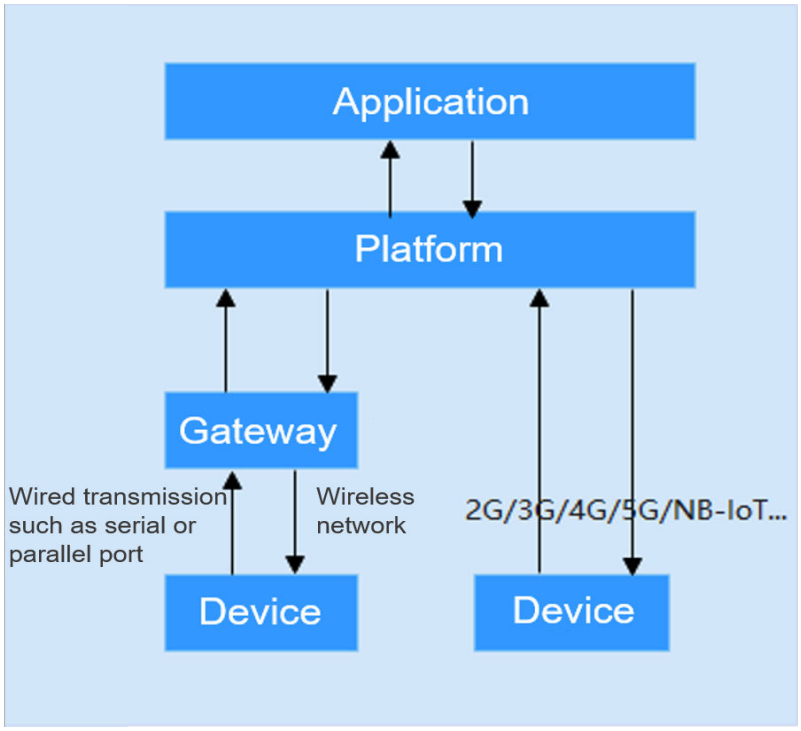
  - The main parameter **Topic name** in the variable header indicates the release topic of the server or client. For details, see **2.3 Topics**.
  - The payload contains the data reported or commands delivered. It is usually a JSON or string object.
- PINGREQ: A heartbeat request is sent from the client to the server. It is used to notify the server that the client is still alive and confirm that the network connection is not disconnected. The sending interval is determined by the negotiated Keep Alive value.

Constraints

- An upstream topic is used by a device to send a request, report data, or return a response to the IoT platform.
- A downstream topic is a command sent by the IoT platform to a device or a response sent by the IoT platform to a device.
- After a device is connected to IoTDA, the device needs to subscribe to downstream topics. Otherwise, the device cannot receive commands delivered or responses returned by IoTDA. (IoTDA uses implicit subscription. Devices subscribe to the **system topic** whose QoS is 0 by default.) The calling of an API on the application side requires the cooperation of the device side. For example, if an application delivers a command, the device needs to subscribe to the downstream topic for command delivery. Otherwise, the device cannot receive commands from the platform, and the APIs for delivering commands will time out.

2.2 Communication Modes

Devices connect to the IoT platform and communicate with an application through the platform. The following figure shows the communication between devices, application, and the platform.



Device Sending Data to the Platform

A device can connect to and communicate with the platform. Data can be sent to the platform through the following APIs:

- API for reporting a message. When a device cannot report data in the property format defined in the product model, it can use this API to report custom data to the platform, which then forwards the data to the application or other Huawei Cloud services for storage and processing.
- API for reporting device properties, which is used by a device to report properties in the format defined in the product model to the platform.
- API for batch reporting device properties, which is used by a gateway to report properties of multiple devices in a batch to the platform.
- API for reporting an event, which is used by a device to report event data in the format defined in the product profile to the platform.

Application Delivering Commands to Devices

An application can use the following APIs to send commands to a device that has been connected to the platform:

- API for delivering a message, which is used by the platform to deliver data in a custom format to a device.
- API for setting device properties. The product model of a device defines the properties that the platform can set for the device. An application can use this API to modify the properties of a device.
- API for querying device properties, which is used by an application to query properties of a device in real time.
- API for delivering a command, which is used by an application to deliver commands in the format defined in the product model to a device.

- API for delivering an event, which is used by an application to deliver an event in the format defined in the product model to a device.

2.3 Topics

When a device connects to IoTDA through MQTT, topics are used in their communication. The following table lists the preset topics in IoTDA.

Category	Topic	Publisher	Subscriber	Function
Device messages	$oc/devices/{device_id}/sys/messages/up	Device	Platform	**Used by a device to report a message**
	$oc/devices/{device_id}/sys/messages/down	Platform	Device	**Used by the platform to deliver a message to a device**
Device commands	$oc/devices/{device_id}/sys/commands/request_id={request_id}	Platform	Device	**Used by the platform to deliver a command to a device**
	$oc/devices/{device_id}/sys/commands/response/request_id={request_id}	Device	Platform	**Used by a device to return a command response**
Device properties	$oc/devices/{device_id}/sys/properties/report	Device	Platform	**Used by a device to report property data**

Category	Topic	Publisher	Subscriber	Function
	$oc/devices/{device_id}/sys/gateway/sub_devices/properties/report	Device	Platform	**Used by a gateway to report property data in batches**
	$oc/devices/{device_id}/sys/properties/set/request_id={request_id}	Platform	Device	**Used by the platform to set device properties**
	$oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}	Device	Platform	**Used by a device to return a response for property setting**
	$oc/devices/{device_id}/sys/properties/get/request_id={request_id}	Platform	Device	**Used by the platform to query device properties**
	$oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}	Device	Platform	**Used by a device to return a response for a property query**. The response does not affect device properties and shadows.

Category	Topic	Publisher	Subscriber	Function
	$oc/devices/{device_id}/sys/shadow/get/request_id={request_id}	Device	Platform	**Used by a device to query device shadow data from the platform**
	$oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}	Platform	Device	**Used by the platform to return a response to a device's query of device shadow data**
Device events	$oc/devices/{device_id}/sys/events/up	Device	Platform	Used by a device to report an event
	$oc/devices/{device_id}/sys/events/down	Platform	Device	Used by the platform to deliver an event

NOTE

- **{device_id}** identifies the target device that messages with a specific topic will be routed to. When a device subscribes to a topic or pushes messages to a topic, the value of this parameter must be replaced with the device ID used for establishing an MQTT connection between the device and the platform.
- **{request_id}** is used to uniquely identify a request. If this parameter is carried in a message sent by a device, ensure that the parameter value is unique on the device by using an incremental number or UUID. **request_id** in the response returned by the device must be the same as that contained in the topic received by the device from IoTDA.
- When a device subscribes to a topic with the **{request_id}** parameter, the parameter can be replaced with the number sign (#). For example, when a device subscribes to the command delivery topic **$oc/devices/{device_id}/sys/commands/request_id={request_id}**, the topic can be specified as **$oc/devices/{device_id}/sys/commands/#**.
- IoTDA uses implicit subscription. Devices do not need to subscribe to the downstream system topic. Devices subscribe to the system topic whose QoS is 0 by default. If the downstream system topic whose QoS is 1 is required, devices need to be configured to subscribe to the topic.
- Except **device_id** and **request_id**, other fields are system fields.

2.4 Device Connection Authentication

API Description

Devices can use this API to sent MQTT CONNECT messages to the platform for authentication. MQTT connections can be established between devices and the platform after successful authentication.

Parameters

Authentication is performed through an MQTT CONNECT message. The information contained in **clientId** must be intact. When receiving a CONNECT message, the platform checks the authentication type and password digest algorithm of the device.

- When the timestamp is verified using the HMAC-SHA256 algorithm, the platform checks whether the message timestamp is consistent with the platform time and then checks whether the password is correct.
- The timestamp must be contained in the CONNECT message even when the timestamp is not verified using the HMAC-SHA256 algorithm. In this case, the platform only checks the password.

If the authentication fails, the platform returns an error message and automatically disconnects the MQTT link.

NOTE

Access the **parameter generation tool** and enter the device ID (**DeviceId**) and secret (**DeviceSecret**) generated after registration to generate the parameters (**ClientId**, **Username**, and **Password**) required for device connection authentication.

Parameter	Mandatory or Optional	Type	Description
clientId	Mandatory	String(256)	The value of this parameter consists of a device ID, device type, password signature type, and timestamp. They are separated by underscores (_). ● Device ID: A device ID uniquely identifies a device and is generated when the device is registered with IoTDA. The value usually consists of a device's product ID and node ID which are separated by an underscore (_). ● Device type: The value is fixed at **0**, indicating a device ID. ● Password signature type: The length is 1 byte, and the value can be **0** or **1**. – **0**: The timestamp is not verified using the HMAC-SHA256 algorithm. – **1**: The timestamp is verified using the HMAC-SHA256 algorithm. ● Timestamp: The UTC time when the device was connected to IoTDA. The format is YYYYMMDDHH. For example, if the UTC time is 2018/7/24 17:56:20, the timestamp is **2018072417**.
userName	Mandatory	String(256)	Device ID.
password	Mandatory	String(256)	A password is the value of **secret** encrypted using the HMAC-SHA256 algorithm with the timestamp as the key. The device secret is returned by IoTDA upon successful device registration.

Return Codes for Connection Setup Using Native MQTT

When a device attempts to establish a connection with the platform through native MQTT, one of the following codes may be returned:

Return Code	Description	Cause
0x00	Successful connection	The connection is successful.

Return Code	Description	Cause
0x01	Request rejected due to incorrect protocol version	The server does not support the MQTT version used by the client request.
0x02	Request rejected due to invalid client ID	The value of **clientId** is in an invalid format or the heartbeat interval does not meet the platform requirements.
0x03	Request rejected due to unavailable server	The platform service is unavailable.
0x04	Request rejected due to incorrect username or password	The username or password is incorrect.
0x05	Unauthorized request	The connection request of the client is not authorized.

2.5 Device Commands

2.5.1 Platform Delivering a Command

Function

This API is used by the platform to deliver a command to a device. After the platform delivers a command, the device needs to return the command execution result to the platform in a timely manner. Otherwise, the platform considers that the command execution has timed out. For differences between command delivery and message delivery, see **Message Communications**.

NOTE

It is not suitable to report data in JSON format for devices with low configuration and limited resources or with limits on bandwidth usage. In this case, devices can transparently transmit the original binary data to the platform, but a codec is required to convert binary data to JSON format. For details about how to develop codecs, see **Developing a Codec**.

Topic

Downstream: $oc/devices/{device_id}/sys/commands/request_id={request_id}

Upstream: $oc/devices/{device_id}/sys/commands/response/request_id={request_id}

NOTE

- **{request_id}** is used to uniquely identify a request. If a device receives a downstream request with a topic carrying a request ID, the request ID must be included in the topic of the upstream response returned by the device.
- When an application **delivers a device command** through the platform, the platform generates a unique ID (**command_id**) to identify the command and returns the ID to the application. In addition, the ID is sent to the device through **requst_id** in the downstream topic of the device command.
- The device cannot detect the **request_id** in advance. When a device subscribes to a topic that ends with **request_id={request_id}**, the number sign (#) can be used to replaced **request_id={request_id}**, for example, **$oc/devices/{device_id}/sys/commands/#**.

Downstream Request Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	• For gateway child device: The value is the child device ID in the topic. • For directly connected device: The value is the same that of **device_id** in the topic.
service_id	Optional	String	Device service ID, which is defined in the **product model** associated with the device.
command_name	Optional	String	Device command name, which is defined in the **product model** associated with the device.
paras	Mandatory	Object	Command execution parameters, which are defined in the **product model** associated with the device.

Upstream Response Parameters

Responses are in JSON format.

Parameter	Mandatory or Optional	Type	Description
result_code	Optional	Integer	Command execution result. **0** indicates a successful execution, whereas other values indicate an execution failure. If this parameter is not specified, the default value **0** is used.

Parameter	Mandatory or Optional	Type	Description
response_name	Optional	String	Response name of the command.
paras	Optional	Object	Indicates the response parameters, which are defined in the **product model** associated with the device.

Example Downstream Request

```
Topic: $oc/devices/{device_id}/sys/commands/request_id={request_id}
Data format:
{
  "object_device_id": "{object_device_id}",
  "command_name": "ON_OFF",
  "service_id": "WaterMeter",
  "paras": {
    "value": "1"
  }
}
```

Example Upstream Response

```
Topic: $oc/devices/{device_id}/sys/commands/response/request_id={request_id}
Data format:
{
    "result_code": 0,
    "response_name": "COMMAND_RESPONSE",
    "paras": {
        "result": "success"
    }
}
```

2.6 Device Messages

2.6.1 Device Reporting a Message

Function

If a device cannot report data in the format defined in the product model, the device can use this API to report data in a custom format to the platform. The platform does not parse the message but forwards it to an application or other Huawei Cloud services for storage and processing.

For differences between message reporting and property reporting, see **Message Communications Overview**.

> **NOTE**
>
> It is not suitable to report data in JSON format for devices with low configuration and limited resources or with limits on bandwidth usage. In this case, devices can transparently transmit the original binary data to the platform, but a codec is required to convert binary data to JSON format. For details about how to develop codecs, see **Developing a Codec**.

Topic

Upstream: $oc/devices/{device_id}/sys/messages/up

> **NOTE**
>
> - In addition to the preset system topics, the device can also use a custom topic that is not **declared by the platform**, for example, **$oc/devices/***{device_id}***/user/***{customized}*.
> - You can add **?request_id** to the end of the topic to specify the request ID for the data reporting. Message reporting example: **$oc/devices/***{device_id}***/sys/messages/up?request_id=***{request_id}*. If this parameter is not specified, the platform automatically generates a request ID to identify the request.

Parameters

Message reporting has no fixed requirements on the data content. The following table lists the parameters involved when the system format is used for delivery.

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	**Explanation:** • Mandatory when the child device reports data during the gateway device data reporting. Specify *object_device_id* as the child device ID of the device in the topic. Otherwise, the request fails. • When a directly connected device reports data, the value of *object_device_id* must be the same as that of *device_id* in the topic. • If this parameter is left blank, the value of this parameter is the same as that of *device_id* in the topic by default.
name	Optional	String	**Explanation:** Message name, which is optional and is used for description.

Parameter	Mandatory or Optional	Type	Description
id	Optional	String	**Explanation:** Unique ID of a message, which is used to distinguish and search for messages. If you do not enter a value, the system automatically generates a message ID. The message ID must be unique.
content	Mandatory	Object, String	**Explanation:** Message content, which can be encoded in Base64 format.

NOTE

IoTDA does not verify the format of the message reported by a device. The parameter description and example are for reference only. Devices can customize the data format as required.

Example

Assume that the data reported by the device is "hello!". The request examples are as follows:

- Sending only the message content:
```
Topic: $oc/devices/{device_id}/sys/messages/up
Data format:
    hello!
```
- Reporting data in the system format:
```
Topic: $oc/devices/{device_id}/sys/messages/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "name":null,
    "id":"aca6a906-c74c-4302-a2ce-b17ba2ce630c",
    "content":"hello!"
}
```

2.6.2 Platform Delivering a Message

Function

This API is used by an application to deliver a message in custom format to a device when the application cannot deliver data in the format defined in the product model.

For differences between command delivery and message delivery, see **Message Communications Overview**.

NOTE

It is not suitable to report data in JSON format for devices with low configuration and limited resources or with limits on bandwidth usage. In this case, devices can transparently transmit the original binary data to the platform, but a codec is required to convert binary data to JSON format. For details about how to develop codecs, see **Developing a Codec**.

Topic

Downstream: $oc/devices/{device_id}/sys/messages/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device to which the message is sent. If this parameter is not carried, the device specified in the topic is considered to be the intended device.
name	Optional	String	Indicates the message name.
id	Optional	String	Uniquely identifies a message.
content	Mandatory	String	Indicates the message content.

Example

```
Topic: $oc/devices/{device_id}/sys/messages/down
Data format:
{
    "object_device_id": "{object_device_id}",
    "name": "name",
    "id": "id",
    "content": "hello"
}
```

2.7 Device Properties

2.7.1 Device Reporting Properties

Function

This API is used by a device to report property data in the format defined in the product model to the platform. For differences between property reporting and message reporting, see **Message Communications Overview**.

NOTE

It is not suitable to report data in JSON format for devices with low configuration and limited resources or with limits on bandwidth usage. In this case, devices can transparently transmit the original binary data to the platform, but a codec is required to convert binary data to JSON format. For details about how to develop codecs, see **Developing a Codec**.

Topic

Upstream: $oc/devices/{device_id}/sys/properties/report

Parameters

Parameter	Mandatory or Optional	Type	Description
services	Mandatory	List<ServiceProperty>	Indicates a list of device services. For details, see **ServiceProperty structure**.

ServiceProperty structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Identifies a service of the device.
properties	Mandatory	Object	Indicates the property list of the device service. Multiple property parameters can be defined in the product model associated with the device.
event_time	Optional	String	Indicates the UTC time when the device collects data. The value is in the format of **yyyyMMdd'T'HHmmss'Z'** or **yyyy-MM-dd'T'HH:mm:ss.SSS'Z'**, for example, 20161219T114920Z or 2020-08-12T12:12:12.333Z. If this parameter is not carried in the reported data or is in incorrect format, the time when the platform receives the data is used.

Example

```
Topic: $oc/devices/{device_id}/sys/properties/report
Data format:
{
    "services": [{
        "service_id": "Temperature",
```

```
            "properties": {
                "value": 57,
                "value2": 60
            },
            "event_time": "20151212T121212Z"
        },
        {
            "service_id": "Battery",
            "properties": {
                "level": 80,
                "level2": 90
            },
            "event_time": "20151212T121212Z"
        }
    ]
}
```

2.7.2 Gateway Reporting Device Properties in Batches

Function

This API is used by a gateway to report property data of multiple child devices in batches to the platform.

For differences between property reporting and message reporting, see **Message Communications Overview**.

NOTE

A gateway can report the properties of a maximum of 100 child devices in each batch. If the number of child devices exceeds 100, you are advised to report their properties in different batches.

Topic

Upstream: $oc/devices/{device_id}/sys/gateway/sub_devices/properties/report

Parameters

Parameter	Mandatory or Optional	Type	Description
devices	Mandatory	List<DeviceProperty>	Indicates the device data.

DeviceService structure

Parameter	Mandatory or Optional	Type	Description
device_id	Mandatory	String	Identifies a device.
services	Mandatory	List<ServiceProperty>	Indicates a list of device services.

ServiceProperty structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Identifies a service of the device.
properties	Mandatory	Object	Indicates the property list of the device service. Multiple property parameters can be defined in the product model associated with the device.
event_time	Optional	String	Indicates the UTC time when the device collects data. The value is in the format of **yyyyMMdd'T'HHmmss'Z'** or **yyyy-MM-dd'T'HH:mm:ss.SSS'Z'**, for example, 20161219T114920Z or 2020-08-12T12:12:12.333Z. If this parameter is not carried in the reported data or is in incorrect format, the time when the platform receives the data is used.

Example

```
Topic: $oc/devices/{device_id}/sys/gateway/sub_devices/properties/report
Data format:
{
    "devices":[
        {
            "device_id":"bf40f0c4-4022-41c6-a201-c5133122054a",
            "services":[
                {
                    "service_id":"analog",
                    "properties":{
                        "PhV_phsA":"1",
                        "PhV_phsB":"2"
                    },
                    "event_time":"20190606T121212Z"
                }
            ]
        },
        {
            "device_id":"42aa08ea-84c1-4025-a7b2-c1f6efe547c2",
            "services":[
                {
                    "service_id":"analog",
                    "properties":{
                        "PhV_phsA":"3",
                        "PhV_phsB":"5"
                    },
                    "event_time":"20190606T121212Z"
                },
                {
```



```
                "service_id":"parameter",
                "properties":{
                    "Load":"6",
                    "ImbA_strVal":"8"
                },
                "event_time":"20190606T121212Z"
            }
        ]
    }
  ]
}
```

2.7.3 Platform Setting Device Properties

Function

This API is used by the platform to set properties of a device. The product model of a device defines the device properties that can be set by the platform. When a device receives a property setting request, the device needs to return the execution result to the platform. Otherwise, the request is considered to have timed out by the platform.

NOTE

It is not suitable to report data in JSON format for devices with low configuration and limited resources or with limits on bandwidth usage. In this case, devices can transparently transmit the original binary data to the platform, but a codec is required to convert binary data to JSON format. For details about how to develop codecs, see **Developing a Codec**.

Topic

Downstream: $oc/devices/{device_id}/sys/properties/set/request_id={request_id}

Upstream: $oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}

NOTE

- **{request_id}** is used to uniquely identify a request. If a device receives a downstream request with a topic carrying a request ID, the request ID must be included in the topic of the upstream response returned by the device.
- When a device subscribes to a topic that ends with **request_id={request_id}**, such as the downstream topic mentioned above, the number sign (#) can be used to replaced **request_id={request_id}**, for example, **$oc/devices/{device_id}/sys/properties/set/#**.

Downstream Request Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies the device whose properties are to be set.

Parameter	Mandatory or Optional	Type	Description
services	Mandatory	List<ServiceProperty>	Indicates a list of device services.

ServiceProperty structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Identifies a service of the device.
properties	Mandatory	Object	Indicates the property list of the device service. Multiple property parameters can be defined in the product model associated with the device.

Upstream Response Parameters

Parameter	Mandatory or Optional	Type	Description
result_code	Optional	Integer	Indicates the command execution result. **0** indicates a successful execution, whereas other values indicate an execution failure. If this parameter is not carried, the execution is considered to be successful.
result_desc	Optional	String	Describes the response to the request for setting properties.

Example Downstream Request

```
Topic: $oc/devices/{device_id}/sys/properties/set/request_id={request_id}
Data format:
{
"object_device_id": "{object_device_id} ",
   "services": [{
         "service_id": "Temperature",
         "properties": {
            "value": 57,
            "value2": 60
```



```
            }
        },
        {
            "service_id": "Battery",
            "properties": {
                "level": 80,
                "level2": 90
            }
        }
    ]
}
```

Example Upstream Response

```
Topic: $oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}
Data format:
{
    "result_code": 0,
    "result_desc": "success"
}
```

2.7.4 Platform Querying Device Properties

Function

This API is used by the platform to query device properties. When a device receives a property query request, the device needs to return its property data to the platform. Otherwise, the request is considered to have timed out by the platform.

NOTE

It is not suitable to report data in JSON format for devices with low configuration and limited resources or with limits on bandwidth usage. In this case, devices can transparently transmit the original binary data to the platform, but a codec is required to convert binary data to JSON format. For details about how to develop codecs, see **Developing a Codec**.

Topic

Downstream: $oc/devices/{device_id}/sys/properties/get/request_id={request_id}

Upstream: $oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}

NOTE

- **{request_id}** is used to uniquely identify a request. If a device receives a downstream request with a topic carrying a request ID, the request ID must be included in the topic of the upstream response returned by the device.
- When a device subscribes to a topic that ends with **request_id={request_id}**, such as the downstream topic mentioned above, the number sign (#) can be used to replaced **request_id={request_id}**, for example, **$oc/devices/{device_id}/sys/properties/get/#**.

Downstream Request Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies the device whose properties are to be queried.
service_id	Optional	String	Identifies a service of the device.

Upstream Response Parameters

Parameter	Mandatory or Optional	Type	Description
services	Optional	List<ServiceProperty>	Indicates a list of device services.

ServiceProperty structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Identifies a service of the device.
properties	Mandatory	Object	Indicates the property list of the device service. Multiple property parameters can be defined in the product model associated with the device.
event_time	Optional	String	Indicates the UTC time when the device collects data. The format is **yyyyMMdd'T'HHmmss'Z'**, for example, **20161219T114920Z**. If this parameter is not carried in the reported data or is in incorrect format, the time when the platform receives the data is used.

Example Downstream Request

```
Topic: $oc/devices/{device_id}/sys/properties/get/request_id={request_id}
Data format:
{
    "object_device_id": "{object_device_id}",
    "service_id": "Temperature"
}
```

Example Upstream Response

```
Topic: $oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}
Data format:
{
    "services": [
        {
            "service_id": "Temperature",
            "properties": {
                "PhV_phsA": "1",
                "PhV_phsB": "2"
            },
            "event_time": "20190606T121212Z"
        }
    ]
}
```

2.7.5 Device Obtaining Device Shadow Data from the Platform

Function

This API is used by a device to obtain device shadow data from IoTDA. You can configure desired data in a device shadow through an application or the platform. When a device goes online and subscribes to this topic, the device can obtain the device shadow data from the platform and synchronize the desired device properties.

The interaction logic is described as follows:

1. The application calls the API for configuring desired properties in a device shadow or configures the device shadow data on the console.

2. The device (which has subscribed to the corresponding topic) proactively sends a request to obtain the device shadow data from the platform.

3. The platform responds to the device request and returns the device shadow data.

4. The device parses the desired properties and modifies device properties.

Topic

Upstream: $oc/devices/{device_id}/sys/shadow/get/request_id={request_id}

Downstream: $oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}

NOTE

- **{request_id}** is used to uniquely identify a request. If this parameter is carried in a message sent by a device, ensure that the parameter value is unique on the device by using an incremental number or UUID.
- When a device subscribes to a topic that ends with **request_id={request_id}**, such as the downstream topic mentioned above, the number sign (#) can be used to replaced **request_id={request_id}**, for example, **$oc/devices/{device_id}/sys/shadow/get/response/#**.

Upstream Request Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies a device that requests its device shadow.
service_id	Optional	String	Identifies a service of the device that requests the device shadow. If this parameter is not specified, device shadow data of all services will be returned.

Downstream Response Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Mandatory	String	Identifies a device that requests its device shadow.
shadow	Optional	List<ShadowData>	Indicates the shadow data.

ShadowData structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Identifies a service of the device.
desired	Optional	PropertiesData	Indicates the property list in the desired section of the device shadow.

Parameter	Mandatory or Optional	Type	Description
reported	Optional	PropertiesData	Indicates the property list in the reported section of the device shadow.
version	Optional	Integer	Indicates the device shadow version.

PropertiesData structure

Parameter	Mandatory or Optional	Type	Description
properties	Mandatory	Object	Indicates the property list of the device service. Multiple property parameters can be defined in the product model associated with the device.
event_time	Optional	String	Indicates the UTC time of the device property data. The format is **yyyyMMdd'T'HHmmss'Z'**, for example, 20161219T114920Z.

Example Upstream Request

```
Topic: $oc/devices/{device_id}/sys/shadow/get/request_id={request_id}
Data format:
{
    "object_device_id": "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",
    "service_id": "WaterMeter"
}
```

Example Downstream Response

```
Topic: $oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}
Data format:
{
  "object_device_id": "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",
  "shadow": [
    {
      "service_id": "WaterMeter",
      "desired": {
        "properties": {
          "temperature": "60"
        },
        "event_time": "20151212T121212Z"
      },
      "reported": {
        "properties": {
          "temperature": "60"
        },
```



```
      "event_time": "20151212T121212Z"
    },
    "version": 1
  }
 ]
}
```

2.8 Gateway and Child Device Management

2.8.1 Platform Notifying a Gateway of New Child Device Connection

Function

This API is used by the platform to notify a gateway that a new child device is connected.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.
event_type	Mandatory	String	System field. The value is fixed to **add_sub_device_notify**.
event_time	Optional	String	Indicates the time when the event occurs.

Parameter	Mandatory or Optional	Type	Description
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
devices	Mandatory	List<DeviceInfo>	Indicates the device list.
version	Mandatory	Long	Indicates the version of the child device information. The gateway can save this value and include it in the request for synchronizing the child device list.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Description
parent_device_id	Mandatory	String	Identifies the parent device.
node_id	Mandatory	String	Indicates the node ID.
device_id	Mandatory	String	Identifies a device.
name	Optional	String	Indicates the device name.
description	Optional	String	Indicates the device description.
manufacturer_id	Optional	String	Identifies a manufacturer.
model	Optional	String	Indicates the device model.

Parameter	Mandatory or Optional	Type	Description
product_id	Optional	String	Identifies a product.
fw_version	Optional	String	Indicates the firmware version.
sw_version	Optional	String	Indicates the software version.
status	Optional	String	Indicates the device status. **ONLINE**: The device is online. **OFFLINE**: The device is offline. **INACTIVE**: The device is not activated.
extension_info	Optional	Object	Indicates the custom extended information.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$sub_device_manager",
        "event_type": "add_sub_device_notify",
        "event_time": "20151212T121212Z",
        "paras": {
            "devices": [{
                "parent_device_id": "c6b39067b0325db34663d3ef421a42f6_12345678",
                "node_id": "subdevice11",
                "device_id": "2bb4ddba-fb56-4566-8577-063ad2f5a6cc",
                "name": "subDevice11",
                "description": null,
                "manufacturer_id": "ofo",
                "model": "twx2",
                "product_id": "c6b39067b0325db34663d3ef421a42f6",
                "fw_version": null,
                "sw_version": null,
                "status": "ONLINE"
            }],
            "version": 1
        }
    }]
}
```

2.8.2 Platform Notifying a Gateway of Child Device Deletion

Function

This API is used by the platform to notify a gateway of the information about a deleted child device.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.
event_type	Mandatory	String	System field. The value is fixed to **delete_sub_device_notify**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
devices	Mandatory	List<DeviceInfo>	Indicates the device list.
version	Mandatory	Long	Indicates the version of the child device information.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Description
parent_device_id	Mandatory	String	Identifies the parent device.
node_id	Optional	String	Indicates the node ID.
device_id	Mandatory	String	Identifies a device.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$sub_device_manager",
        "event_type": "delete_sub_device_notify",
        "event_time": "20151212T121212Z",
        "paras": {
            "devices": [{
                "parent_device_id": "c6b39067b0325db34663d3ef421a42f6_12345678",
                "node_id": "subdevice11",
                "device_id": "2bb4ddba-fb56-4566-8577-063ad2f5a6cc"
            }],
            "version": 1
        }
    }]
}
```

2.8.3 Gateway Synchronizing Child Device Information

Function

This API is used by a gateway to synchronize child device information from the platform. The platform can notify an online gateway of child device addition or deletion by respectively using the topics described in **2.8.1 Platform Notifying a Gateway of New Child Device Connection** and **2.8.2 Platform Notifying a Gateway of Child Device Deletion**. However, the platform cannot send notifications to a gateway that is offline. When the gateway goes online, it can use the topic described here to synchronize information about child device addition or deletion from the platform.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<EventService>	Indicates a list of services that the event is about.

EventService structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.
event_type	Mandatory	String	System field. The value is fixed to **sub_device_sync_request**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
version	Optional	Long	Indicates the last version of child device addition or deletion information received by the gateway. The platform will send the information of later versions to the gateway during the synchronization.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$sub_device_manager",
        "event_type": "sub_device_sync_request",
        "event_time": "20151212T121212Z",
        "paras": {"version": 1}
    }]
}
```

2.8.4 Gateway Updating Child Device Status

Function

This API is used by a gateway to update child device status.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<EventService>	Indicates a list of services that the event is about.

EventService structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.
event_type	Mandatory	String	System field. The value is fixed to **sub_device_update_status**.
event_time	Optional	String	Indicates the time when the event occurs.

Parameter	Mandatory or Optional	Type	Description
event_id	Optional	String	Uniquely identifies an event. If this parameter is not carried, an event ID will be generated by the platform. An event ID is a random string of 36 characters consisting of letters, numbers, and hyphens (-).
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
device_statuses	Mandatory	List<DeviceStatus>	Indicates the device status list. The list can contain status information of 1 to 100 devices.

DeviceStatus structure

Parameter	Mandatory or Optional	Type	Description
device_id	Mandatory	String	Identifies a child device.
status	Mandatory	String	Indicates the child device status. • **OFFLINE**: The device is offline. • **ONLINE**: The device is online.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "services": [{
        "service_id": "$sub_device_manager",
        "event_type": "sub_device_update_status",
        "event_time": "20151212T121212Z",
        "paras": {
            "device_statuses": [{
                    "device_id": "bf40f0c4-4022-41c6-a201-c5133122054a",
                    "status": "ONLINE"
                },
                {
```



```
                "device_id": "4459c0f7-10bb-4718-9b07-7a82c2d508a5",
                "status": "ONLINE"
            }
        ]
    }
  }]
}
```

2.8.5 Responding to a Request for Updating Child Device Statuses

Function

This API is used by IoTDA to respond to a gateway's request for updating child devices. The response is returned within 30 seconds after IoTDA receives the request and includes a list of updated statuses of child devices.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies a gateway.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	$sub_device_manager
event_type	Mandatory	String	sub_device_update_status
event_time	Optional	String	Indicates the time when the event occurs.
event_id	Mandatory	String	Identifies a request event. The value is used to associate with the corresponding request event.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
successful_devices	Mandatory	List<DeviceStatus>	Indicates the details about the child devices that are updated.
failed_devices	Mandatory	List<Reason>	Indicates the cause of the failure to update the child device status.

DeviceStatus structure

Parameter	Mandatory or Optional	Type	Description
device_id	Mandatory	String	Identifies a device.
status	Optional	String	Indicates the device status update result. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline.

Reason structure

Parameter	Mandatory or Optional	Type	Description
device_id	Mandatory	String	The value is that of **device_id** of the corresponding device in the corresponding request.
error_code	Mandatory	String	Error code.
error_msg	Mandatory	String	Error description.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
```



```
 "object_device_id": "{object_device_id}",
 "services": [
  {
    "service_id": "$sub_device_manager",
    "event_type": "sub_device_update_status_response",
    "event_time": "20151212T121212Z",
    "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
    "paras": {
        "successful_devices": [
        {
          "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice11",
          "status": "ONLINE"
        }
      ],
      "failed_devices": [
        {
          "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice11",
          "error_code": "XXX",
          "error_msg": "XXXX"
        }
      ]

    }
  }
 ]
}
```

2.8.6 Gateway Requesting for Adding Child Devices

Function

This API is used by a gateway to register new child devices on the platform.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies a gateway.
services	Mandatory	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.

Parameter	Mandatory or Optional	Type	Description
event_type	Mandatory	String	System field. The value is fixed to **add_sub_device_request**.
event_time	Optional	String	Indicates the time when the event occurs.
event_id	Optional	String	Uniquely identifies an event. If this parameter is not carried, an event ID will be generated by the platform. An event ID is a random string of 36 characters consisting of letters, numbers, and hyphens (-).
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
devices	Mandatory	List<DeviceInfo>	Indicates a list of child devices to add. Up to 50 child devices can be added at a time.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Description
parent_device_id	Optional	String	Identifies the parent device, that is, the gateway.
node_id	Mandatory	String	Indicates the node ID.

Parameter	Mandatory or Optional	Type	Description
device_id	Optional	String	Identifies a device. If this parameter is carried, the platform will use the parameter value as the device ID. Otherwise, the platform will allocate a device ID, which is in the format of *[product_id]_ [node_id]*.
name	Optional	String	Indicates the device name.
description	Optional	String	Indicates the device description.
product_id	Mandatory	String	Uniquely identifies the product associated which the device. The value is allocated by the platform after the product model is imported to the IoTDA console.
extension_info	Optional	Object	Indicates the device extended information, which can be customized. The maximum size of the value is 1 KB.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "add_sub_device_request",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "devices": [
          {
            "name": "subdevice11",
            "node_id": "subdevice11",
            "product_id": "c6b39067b0325db34663d3ef421a42f6",
            "description": "subdevice11"
          },
          {
            "name": "subdevice12",
            "node_id": "subdevice12",
            "product_id": "c6b39067b0325db34663d3ef421a42f6",
            "description": "subdevice12"
          }
        ]
      }
    }
```



```
 ]
}
```

2.8.7 Platform Responding to a Request for Adding Child Devices

Function

This API is used by the platform to respond to a gateway's request for adding child devices. The response is returned within 30 seconds after the platform receives the request and includes a list of child devices that are added.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies a gateway.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	The value is **$sub_device_manager**.
event_type	Mandatory	String	The value is **add_sub_device_response**.
event_time	Optional	String	Indicates the time when the event occurs.
event_id	Mandatory	String	Identifies a request event. The value is used to associate with the corresponding request event.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
successful_devices	Mandatory	List<DeviceInfo>	Indicates the details about the child devices that are added.
failed_devices	Mandatory	List<Reason>	Indicates the cause for the failure to add child devices.

DeviceInfo structure

Parameter	Mandatory or Optional	Type	Description
parent_device_id	Mandatory	String	Identifies the parent device.
node_id	Mandatory	String	Indicates the node ID.
device_id	Mandatory	String	Identifies a device.
name	Optional	String	Indicates the device name.
description	Optional	String	Indicates the device description.
manufacturer_id	Optional	String	Identifies a manufacturer.
model	Optional	String	Indicates the device model.
product_id	Optional	String	Identifies a product.
fw_version	Optional	String	Indicates the firmware version.
sw_version	Optional	String	Indicates the software version.

Parameter	Mandatory or Optional	Type	Description
status	Optional	String	Indicates the device status. • **ONLINE**: The device is online. • **OFFLINE**: The device is offline. • **INACTIVE**: The device is not activated.
extension_info	Optional	Object	Indicates the device extended information, which can be customized.

Reason structure

Parameter	Mandatory or Optional	Type	Description
node_id	Mandatory	String	The value is that of **node_id** of the corresponding device in the corresponding request.
product_id	Mandatory	String	The value is that of **product_id** of the corresponding device in the corresponding request.
error_code	Mandatory	String	Indicates the error code.
error_msg	Mandatory	String	Indicates the error description.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "add_sub_device_response",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
          "successful_devices": [
          {
            "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice11",
            "name": "subdevice11",
```



```
        "node_id": "subdevice11",
        "product_id": "c6b39067b0325db34663d3ef421a42f6",
        "description": "subdevice11",
        "manufacturer_id": "ofo",
        "model": "twx2",
        "fw_version": null,
        "sw_version": null,
        "status": "ONLINE",
        "extension_info" : null,
        "parent_device_id" : null
       }
      ],
      "failed_devices": [
       {
        "node_id": "subdevice12",
        "product_id": "c6b39067b0325db34663d3ef421a42f6",
        "error_code": "XXX",
        "error_msg": "XXXX"
       }
      ]

    }
   }
  ]
}
```

2.8.8 Gateway Requesting for Deleting Child Devices

Function

This API is used by a gateway to deregister child devices on the platform.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies a gateway.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.

Parameter	Mandatory or Optional	Type	Description
event_type	Mandatory	String	System field. The value is fixed to **delete_sub_device_request**.
event_time	Optional	String	Indicates the time when the event occurs.
event_id	Optional	String	Uniquely identifies an event. If this parameter is not carried, an event ID will be generated by the platform. An event ID is a random string of 36 characters consisting of letters, numbers, and hyphens (-).
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
devices	Mandatory	List<String>	Identifies the child devices to delete. Up to 50 child devices can be deleted at a time.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "delete_sub_device_request",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "devices": [
          "c6b39067b0325db34663d3ef421a42f6_subdevice11",
          "c6b39067b0325db34663d3ef421a42f6_subdevice12"
        ]
      }
    }
  ]
}
```


2.8.9 Platform Responding to a Request for Deleting Child Devices

Function

This API is used by the platform to respond to a gateway's request for deleting child devices. The response is returned within 30 seconds after the platform receives the request and includes a list of child devices that are deleted.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Identifies a gateway.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$sub_device_manager**.
event_type	Mandatory	String	System field. The value is fixed to **delete_sub_device_response**.
event_time	Optional	String	Indicates the time when the event occurs.
event_id	Mandatory	String	Identifies a request event. The value is used to associate with the corresponding request event.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
successful_devices	Mandatory	List<String>	Indicates the device IDs of child devices that are deleted.
failed_devices	Mandatory	List<Reason>	Indicates the cause for the deletion failure.

Reason structure

Parameter	Mandatory or Optional	Type	Description
device_id	Mandatory	String	The value is that of **device_id** of the corresponding device in the corresponding request.
error_code	Mandatory	String	Indicates the error code.
error_msg	Mandatory	String	Indicates the error description.

Example

Topic: $oc/devices/{device_id}/sys/events/down
Data format:

```
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "delete_sub_device_response",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "successful_devices": [
          "c6b39067b0325db34663d3ef421a42f6_subdevice11"
        ],
        "failed_devices": [
          {
            "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice12",
            "error_code": "XXX",
            "error_msg": "XXXX"
          }
        ]
      }
    }
  ]
}
```


2.9 Software and Firmware Upgrade

2.9.1 Platform Delivering an Event to Obtain Version Information

Function

This API is used by the platform to deliver an event to obtain version information.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$ota**.
event_type	Mandatory	String	System field. The value is fixed to **version_query**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Optional	Object	Indicates the JSON object of the event parameter. No specific field is delivered when the event to obtain version information is delivered.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$ota",
        "event_type": "version_query",
        "event_time": "20151212T121212Z",
         "paras": {
         }
    }]
}
```

2.9.2 Device Reporting the Software and Firmware Versions

Function

This API is used by a device to report the software and firmware versions.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$ota**.
event_type	Mandatory	String	System field. The value is fixed to **version_report**.
event_time	Optional	String	Indicates the time when the event occurs.

Parameter	Mandatory or Optional	Type	Description
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
sw_version	Optional	String	Indicates the software version.
fw_version	Optional	String	Indicates the firmware version.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$ota",
        "event_type": "version_report",
        "event_time": "20151212T121212Z",
        "paras": {
            "sw_version": "v1.0",
            "fw_version": "v1.0"
        }
    }]
}
```

2.9.3 Platform Delivering an Upgrade Event

Function

This API is used by IoTDA to deliver an upgrade notification to a device.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$ota**.
event_type	Mandatory	String	If the software or firmware is stored in IoTDA, the values of **event_type** are as follows: **firmware_upgrade**: firmware upgrade. **software_upgrade**: software upgrade. If the software or firmware is stored in OBS, the values of **event_type** are as follows: **firmware_upgrade_v2**: firmware upgrade. **software_upgrade_v2**: software upgrade.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

Parameters when event_type is set to firmware_upgrade or software_upgrade

Parameter	Mandatory or Optional	Type	Description
version	Mandatory	String	Indicates the software or firmware package version.
url	Mandatory	String	Indicates the address for downloading the software or firmware package.
file_size	Mandatory	Integer	Indicates the software or firmware package size.
access_token	Optional	String	Indicates the temporary token of the URL for downloading the software or firmware package.
expires	Optional	Integer	Indicates the time when the token expires.
sign	Mandatory	String	Indicates the SHA-256 value of the software or firmware package.
custom_info	Optional	String	Indicates the custom information pushed to a device.

Parameters when event_type is set to firmware_upgrade_v2 or software_upgrade_v2

Parameter	Mandatory or Optional	Type	Description
version	Mandatory	String	Indicates the software or firmware package version.
url	Mandatory	String	Indicates the software or firmware package download address (OBS address)
expires	Optional	Integer	Indicates the time when the URL expires.
custom_info	Optional	String	Indicates the custom information pushed to a device.

Example 1

If the software or firmware is stored in IoTDA. During the upgrade, the device receives the following information:
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{


```
   "object_device_id": "{object_device_id}",
   "services": [{
      "service_id": "$ota",
      "event_type": "firmware_upgrade",
      "event_time": "20151212T121212Z",
      "paras": {
         "version": "v1.2",
         "url": "https://10.1.1.1:8943/iodm/inner/v1.3.0/firmwarefiles/ca1d954771ae61e5098c7f83",
         "file_size": 81362928,
         "access_token": "595124473f866b033dfa1f",
         "expires": 86400,
         "sign": "595124473f866b033dfa1f7e831c8c99a12f6143f392dfa996a819010842c99d",
         "custom_info":"This upgrade package adds some new features."
      }
   }]
}
```

Example 2

```
If the software or firmware is stored in OBS. During the upgrade, the device receives the following
information:
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
   "object_device_id": "{object_device_id}",
   "services": [{
      "service_id": "$ota",
      "event_type": "firmware_upgrade_v2",
      "event_time": "20151212T121212Z",
      "paras": {
         "version": "v1.2",
         "url": "https://******.obs.cn-north-4.myhuaweicloud.com:443/test.bin?AccessKeyId=DX5G7W*********",
         "expires": 3600,
         "custom_info":"This upgrade package adds some new features."
      }
   }]
}
```

Downloading an Upgrade Package

After a device receives an upgrade notification, the device downloads the upgrade package based on the URL in the notification through HTTPS. For basic and standard editions, you are advised not to verify certificates to prevent download failures. If the exclusive edition requires certification verification, submit a service ticket for the back end personnel to configure the domain name.

Request Method

The request method for downloading the upgrade package is GET.

Request Header

You can add header fields, for example, fields required by a specific URI or HTTP method, to the request header. For example, to request authentication information, add **Content-Type**, which specifies the request body type.

Parameter	Description
Content-Type	Media type of the message body. The default value is **application/json**.

Parameter	Description
Authorization	Authentication information for accessing IoTDA. The value is **Bearer** *{access_token}*, in which *{access_token}* is the value of **access_token** in the received upgrade notification.

Example

```
GET https://10.1.1.1:8943/iodm/inner/v1.3.0/firmwarefiles/ca1d954771ae61e5098c7f83
Content-Type: application/json
Authorization: Bearer ******
```

CAUTION

If **event_type** is set to **firmware_upgrade_v2** or **software_upgrade_v2**, the request header does not need to be carried in the request for downloading the software or firmware package. An example request is as follows:

GET https://******.obs.cn-north-4.myhuaweicloud.com:443/test.bin?AccessKeyId=DX5G7W*********

2.9.4 Device Reporting the Upgrade Status

Function

This API is used by a device to report the upgrade status.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$ota**.
event_type	Mandatory	String	System field. The value is fixed to **upgrade_progress_report**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
result_code	Mandatory	Integer	Indicates the device upgrade status. ● **0**: successful upgrade ● **1**: device in use ● **2**: poor signal ● **3**: already the latest version ● **4**: low battery ● **5**: insufficient storage space ● **6**: download timeout ● **7**: upgrade package verification failure ● **8**: unsupported upgrade package type ● **9**: insufficient memory ● **10**: upgrade package installation failure ● **255**: internal exception
progress	Optional	Integer	Indicates the device upgrade progress. The value ranges from 0 to 100.

Parameter	Mandatory or Optional	Type	Description
version	Mandatory	String	Indicates the current version of the device. (The version reported after the upgrade must be the same as that set in IoTDA when you upload the software or firmware package.)
description	Optional	String	Describes the upgrade status, such as the cause for upgrade failure.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$ota",
        "event_type": "upgrade_progress_report",
        "event_time": "20151212T121212Z",
        "paras": {
            "result_code": 0,
            "progress": 80,
            "version": "V2.0",
            "description": "upgrade processing"
        }
    }]
}
```

2.10 File Upload and Download

2.10.1 Device Requesting a URL for File Upload

Function

This API is used by a device to report a request for obtaining a URL for file upload.

Topic

$oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$file_manager**.
event_type	Mandatory	String	System field. The value is fixed to **get_upload_url**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
file_name	Mandatory	String	Indicates the name of the file to upload.
file_attributes	Optional	Object	Indicates the file attributes in JSON format.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
```



```
  "services": [{
     "service_id": "$file_manager",
     "event_type": "get_upload_url",
     "event_time": "20151212T121212Z",
      "paras": {
         "file_name": "a.jpg",
         "file_attributes": {
            "hash_code": "58059181f378062f9b446e884362a526",
            "size": 1024
         }
      }
  }]
}
```

2.10.2 Platform Delivering a Temporary URL for File Upload

Function

This API is used by the platform to deliver a temporary URL for file upload.

Topic

$oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$file_manager**.
event_type	Mandatory	String	System field. The value is fixed to **get_upload_url_response**.
event_time	Optional	String	Indicates the time when the event occurs.

Parameter	Mandatory or Optional	Type	Description
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
url	Mandatory	String	Indicates the URL for file upload.
bucket_name	Optional	String	Indicates the name of an OBS bucket.
object_name	Optional	String	Indicates the name of the object to be uploaded to OBS, which is the same as the value of **file_name**.
expire	Optional	Integer	Indicates the URL validity period, in seconds.
file_attributes	Optional	Object	Indicates the file attributes in JSON format.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$file_manager",
        "event_type": "get_upload_url_response",
        "event_time": "20151212T121212Z",
         "paras": {
            "url":"https://bucket.obs.cn-north-4.com/device_file/aGEKIpp5NAGxdP2oo90000/a.jpg?
Expires=1553162075&OSSAccessKeyId=LTAIYLScbHiV****&Signature=%2F88xdEFPukJ****%2F8****%2Bdv3io
%3D",
            "bucket_name":"bucket",
            "object_name": "c6b39067b0325db34663d3ef421a42f6_12345678_a.jpg",
            "expire":3600,
            "file_attributes": {
                "hash_code": "58059181f378062f9b446e884362a526",
                "size": 1024
            }
        }
    }]
}
```


2.10.3 Device Reporting File Upload Results

Function

This API is used by a device to report the file upload result.

Topic

$oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$file_manager**.
event_type	Mandatory	String	System field. The value is fixed to **upload_result_report**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
object_name	Mandatory	String	Indicates the name of the object uploaded to OBS.
result_code	Mandatory	Integer	Indicates the file upload result. • **0**: successful upload • **1**: upload failure
status_code	Optional	Integer	Indicates the status code returned by OBS after a file upload.
status_description	Optional	String	Indicates the status description returned by OBS after a file upload.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$file_manager",
        "event_type": "upload_result_report",
        "event_time": "20151212T121212Z",
         "paras": {
            "object_name": "c6b39067b0325db34663d3ef421a42f6_12345678_a.jpg",
            "result_code": 0,
            "status_code": 200,
            "status_description": "upload success"
         }
    }]
}
```

2.10.4 Device Requesting a URL for File Download

Function

This API is used by a device to report a request for obtaining a URL for file download.

Topic

$oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$file_manager**.
event_type	Mandatory	String	System field. The value is fixed to **get_download_url**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
fileName	Mandatory	String	Indicates the name of the file to download.
file_attributes	Optional	Object	Indicates the file attributes in JSON format.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
```



```
  "services": [{
     "service_id": "$file_manager",
     "event_type": "get_download_url",
     "event_time": "20151212T121212Z",
      "paras": {
         "file_name": "a.jpg",
         "file_attributes": {
            "hash_code": "58059181f378062f9b446e884362a526",
            "size": 1024
         }
      }
  }]
}
```

2.10.5 Platform Delivering a Temporary URL for File Download

Function

This API is used by the platform to deliver a temporary URL for file download to a device.

Topic

$oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$file_manager**.
event_type	Mandatory	String	System field. The value is fixed to **get_download_url_response**.
event_time	Optional	String	Indicates the time when the event occurs.

Parameter	Mandatory or Optional	Type	Description
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
url	Mandatory	String	Indicates the URL for file download.
bucket_name	Optional	String	Indicates the name of an OBS bucket.
object_name	Optional	String	Indicates the name of the object to be downloaded from OBS, which is the value of **file_name**.
expire	Optional	Integer	Indicates the URL validity period, in seconds.
file_attributes	Optional	Object	Indicates the file attributes in JSON format.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$file_manager",
        "event_type": "get_download_url_response",
        "event_time": "20151212T121212Z",
         "paras": {
            "url":"https://bucket.obs.cn-north-4.com/device_file/aGEKIpp5NAGxdP2oo90000/a.jpg?
Expires=1553162075&OSSAccessKeyId=LTAIYLScbHiV****&Signature=%2F88xdEFPukJ****%2F8****%2Bdv3io
%3D",
            "bucket_name":"bucket",
            "object_name": "c6b39067b0325db34663d3ef421a42f6_12345678_a.jpg",
            "expire":3600,
            "file_attributes": {
                "hash_code": "58059181f378062f9b446e884362a526",
                "size": 1024
            }

        }
    }]
}
```


2.10.6 Device Reporting File Download Results

Function

This API is used by a device to report the file download result.

Topic

$oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$file_manager**.
event_type	Mandatory	String	System field. The value is fixed to **download_result_report**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
object_name	Mandatory	String	Indicates the name of the object downloaded from OBS.
result_code	Mandatory	Integer	Indicates the download result. • **0**: successful download • **1**: download failure
status_code	Optional	Integer	Indicates the status code returned by OBS after a file download.
status_description	Optional	String	Indicates the status description returned by OBS after a file download.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$file_manager",
        "event_type": "download_result_report",
        "event_time": "20151212T121212Z",
        "paras": {
            "object_name": "c6b39067b0325db34663d3ef421a42f6_12345678_a.jpg",
            "result_code": 0,
            "status_code": 200,
            "status_description": "download success"
        }
    }]
}
```

2.11 Device Time Synchronization

2.11.1 Device Requesting Time Synchronization

Function

This API is used by a device to send a request for synchronizing time with the platform.

Topic

Topic: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$time_sync**.
event_type	Mandatory	String	System field. The value is fixed to **time_sync_request**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
device_send_time	Mandatory	long	Indicates the timestamp when the device sends the request. The timestamp is the number of milliseconds since 00:00:00 on January 1, 1970 in Greenwich Mean Time (GMT).

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
```



```
{
   "object_device_id": "{object_device_id}",
   "services": [{
      "service_id": "$time_sync",
      "event_type": "time_sync_request",
      "event_time": "20151212T121212Z",
       "paras": {
          "device_send_time": 1582685678789
       }
   }]
}
```

2.11.2 Platform Responding to a Request for Time Synchronization

Function

This API is used by the platform to send a response to the device. The device sending parameter **device_send_time** is carried. **server_recv_time** indicates the time when the platform receives the device request. **server_send_time** indicates the time when the platform sends a response to the device.

Assume that the time when the device receives the response is **device_recv_time**. The device calculates the accurate time as follows:

(server_recv_time + server_send_time + device_recv_time - device_send_time) / 2

Topic

Topic: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$time_sync**.
event_type	Mandatory	String	System field. The value is fixed to **time_sync_response**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
device_send_time	Mandatory	long	Indicates the timestamp when the device sends the request. The timestamp is the number of milliseconds since 00:00:00 on January 1, 1970 in GMT.
server_recv_time	Mandatory	long	Indicates the timestamp when the platform receives the request.
server_send_time	Mandatory	long	Indicates the timestamp when the platform sends the response.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
   "object_device_id": "{object_device_id}",
   "services": [{
      "service_id": "$time_sync",
      "event_type": "time_sync_response",
      "event_time": "20151212T121212Z",
       "paras": {
          "device_send_time": 1582685678789,
          "server_recv_time": 1582685696152,
          "server_send_time": 1582685708109
       }
   }]
}
```

2.12 Device Reporting Information

2.12.1 Device Reporting Information

Function

This API is used by a device to report device information to the platform.

Topic

Topic: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Indicates the system field. The value is fixed to **$sdk_info**.
event_type	Mandatory	String	Indicates the system field. The value is fixed to **sdk_info_report**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Optional	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
device_sdk_version	Optional	String	The value is in the format of *access mode_version number*, for example, **C_v0.5.0**, **JAVA_v0.5.0**, or **Tiny SDK_v1.0.0**.
sw_version	Optional	String	Indicates the software version.
fw_version	Optional	String	Indicates the firmware version.
device_ip	Optional	String	Device IP address.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id": "{object_device_id}",
    "services": [{
        "service_id": "$sdk_info",
        "event_type": "sdk_info_report",
        "event_time": "20151212T121212Z",
         "paras": {
            "device_sdk_version": "C_v0.5.0",
            "sw_version": "v1.0",
            "fw_version": "v1.0",
            "device_ip": "127.0.0.1"
         }
    }]
}
```

2.13 Device Log Collection

2.13.1 Platform Delivering a Log Collection Notification

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Function

This API is used by the platform to deliver a log collection notification to devices.

Parameters

Parameter	Mandatory or Optional	Type	Description
services	Optional	List<EventService>	Indicates a list of services that the event is about.

EventService structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$log**.
event_type	Mandatory	String	System field. The value is fixed to **log_config**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the list of service events, which are defined in the product model associated with the device.

paras structure

Parameter	Mandatory or Optional	Type	Description
switch	Optional	String	Indicates the device log collection switch. **on**: enables device log collection. **off**: disables device log collection.
end_time	Optional	String	Indicates the time when log collection ends. yyyyMMdd'T'HHmmss'Z'

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "services": [{
        "service_id": "$log",
```



```
        "event_type": "log_config",
        "event_time": "20151212T121212Z",
        "paras": {
            "switch": "on",
            "end_time": "20151212T131212Z"
        }
    }]
}
```

2.13.2 Device Reporting Log Content

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Function

This API is used by a device to report log content to the platform when log collection is enabled. The maximum size of the content is 1 MB.

Parameters

Parameter	Mandatory or Optional	Type	Description
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	System field. The value is fixed to **$log**.
event_type	Mandatory	String	System field. The value is fixed to **log_report**.
event_time	Optional	String	Indicates the time when the event occurs.
paras	Mandatory	Object	Indicates the list of service events, which are defined in the product model associated with the device.

paras structure

Parameter	Mandatory or Optional	Type	Description
timestamp	Optional	String	Indicates the log generation time.
type	Mandatory	String	Indicates the log type. Options: **DEVICE_STATUS**: device status. **DEVICE_PROPERTY**: device property. **DEVICE_MESSAGE**: device message. **DEVICE_COMMAND**: device command.
content	Mandatory	String	Indicates the log content.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "services": [{
        "service_id": "$log",
        "event_type": "log_report",
        "event_time": "20151212T121212Z",
        "paras": {
            "timestamp": "1235668997",
            "type": "DEVICE_MESSAGE",
            "content": "log content"
        }
    }]
}
```

2.14 Remote Configuration

2.14.1 Platform Delivering a Configuration Notification

Function

This API is used by the platform to deliver a configuration notification to a device. The platform allows you to perform remote configuration. You can remotely update device configuration items such as system and running parameters without interrupting device running. For example, you can remotely modify system parameters of cashiers running in Windows and the data reporting frequency of T-Boxes in the Internet of Vehicles (IoV) scenarios.

Topic

Downstream: $oc/devices/{device_id}/sys/events/down

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Indicates the system field. The value is fixed to **$device_config**.
event_type	Mandatory	String	Indicates the system field. The value is fixed to **config_update**.
event_time	Optional	String	Indicates the time when the event occurs. UTC time format: yyyyMMdd'T'HHmmss'Z'.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
config_content	Mandatory	Object	Indicates the configuration content.

Example

```
Topic: $oc/devices/{device_id}/sys/events/down
Data format:
{
    "object_device_id":"{object_device_id}",
```



```
  "services":[
    {
      "service_id":"$device_config",
      "event_type":"config_update",
      "event_time":"20151212T121212Z",
      "paras":{
        "config_content":{
          "config_key1":"device config1",
          "config_key2":"device config2"
        }
      }
    }
  ]
}
```

2.14.2 Device Reporting the Configuration Response

Function

This API is used by a device to report the configuration result. When creating a remote configuration task, you can configure a timeout interval (1 to 30 days). If not configured, the timeout interval is set to 30 days by default. The platform delivers configuration items to the device every 24 hours within the timeout interval until the device returns a configuration response. If the device does not return a configuration response within the timeout interval, the platform will display a message indicating that the configuration task times out.

Topic

Upstream: $oc/devices/{device_id}/sys/events/up

Parameters

Parameter	Mandatory or Optional	Type	Description
object_device_id	Optional	String	Indicates the device that the event is about. If this parameter is not carried, the device specified in the topic is considered to be the device involved.
services	Optional	List<ServiceEvent>	Indicates a list of services that the event is about.

ServiceEvent structure

Parameter	Mandatory or Optional	Type	Description
service_id	Mandatory	String	Indicates the system field. The value is fixed to **$device_config**.

Parameter	Mandatory or Optional	Type	Description
event_type	Mandatory	String	Indicates the system field. The value is fixed to **config_update_response**.
event_time	Optional	String	Indicates the time when the event occurs. UTC time format: yyyyMMdd'T'HHmmss'Z'.
paras	Mandatory	Object	Indicates the event parameters in JSON format.

paras structure

Parameter	Mandatory or Optional	Type	Description
result_code	Mandatory	Integer	Indicates the device configuration result. ● **0**: successful ● Other integers: abnormal
description	Optional	String	Describes the configuration result, such as the cause for configuration failure.

Example

```
Topic: $oc/devices/{device_id}/sys/events/up
Data format:
{
    "object_device_id":"{object_device_id}",
    "services":[
        {
            "service_id":"$device_config",
            "event_type":"config_update_response",
            "event_time":"20151212T121212Z",
            "paras":{
                "result_code":0,
                "description":"update config success"
            }
        }
    ]
}
```


3 HTTPS API Reference on the Device Side

3.1 Using HTTPS For Access

Overview

Hypertext Transfer Protocol Secure (HTTPS) is a secure communication protocol that uses the SSL encryption over the HTTP protocol. It is supported by IoTDA.

Constraints

Item	Limit
Supported HTTP version	HTTP 1.0 HTTP 1.1
Supported HTTPS	The IoT platform supports only the HTTPS protocol. For details about how to download a certificate, see **Certificates**.
Supported TLS version	TLS 1.2
Maximum body length	1 MB
API specifications	For details, see **Specifications**.
Maximum number of child devices of which properties can be reported by a gateway at a time	50

API Calling

For details about the platform endpoint, see **Platform Connection Information**.

NOTE

Use the endpoint of IoTDA and the HTTPS port number 443.

Communication Between HTTPS Devices and the Platform

When a device connects to the platform through HTTPS, HTTPS APIs are used for their communication. These APIs can be used for device authentication as well as message and property reporting.

Message Type	Description
Device authentication	Devices obtain access tokens.
Device reporting properties	Devices report property data in the format defined in the product model.
Device reporting messages	Devices report custom data to IoTDA, which then forwards reported messages to an application or other Huawei Cloud services for storage and processing.
Gateway reporting device properties in batches	A gateway reports property data of multiple child devices to the platform.

Service Flow

1. Create a product on the IoTDA console or by calling the API for **creating a product**.
2. Register a device on the **IoTDA console** or calling the API **Creating a Device**.
3. Call the device authentication API to obtain the access token of the device.
   a. **Enter device ID and secret**, and obtain the timestamp and encrypted password.
   b. Edit the JSON authentication message body based on **Table 3-1**. **Figure 3-2** is an example.

Table 3-1 JSON authentication message body

Item	Content
device_id	Device ID.

Item	Content
sign_type	The recommended value is **0**, indicating that the system does not check whether the message timestamp is the same as the platform time and only checks whether the password is correct.
timestamp	Timestamp, for example, 2024062602. Obtain the value based on the client ID in **Figure 3-1**.
password	Encrypted password, which is the value of **Password** in **Figure 3-1**.

Figure 3-1 Client ID generator

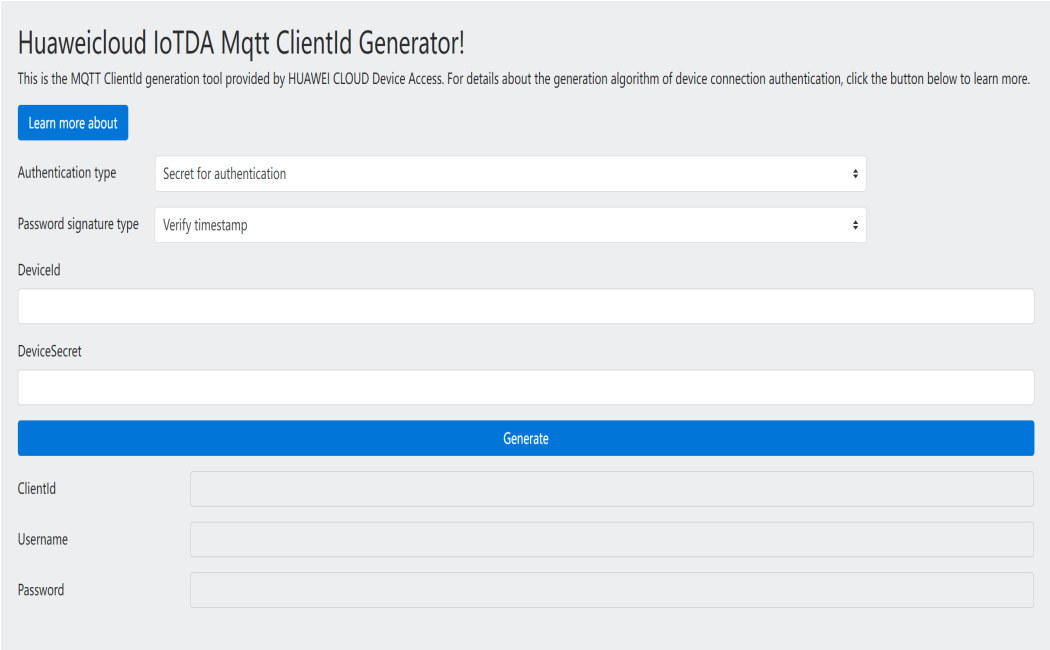
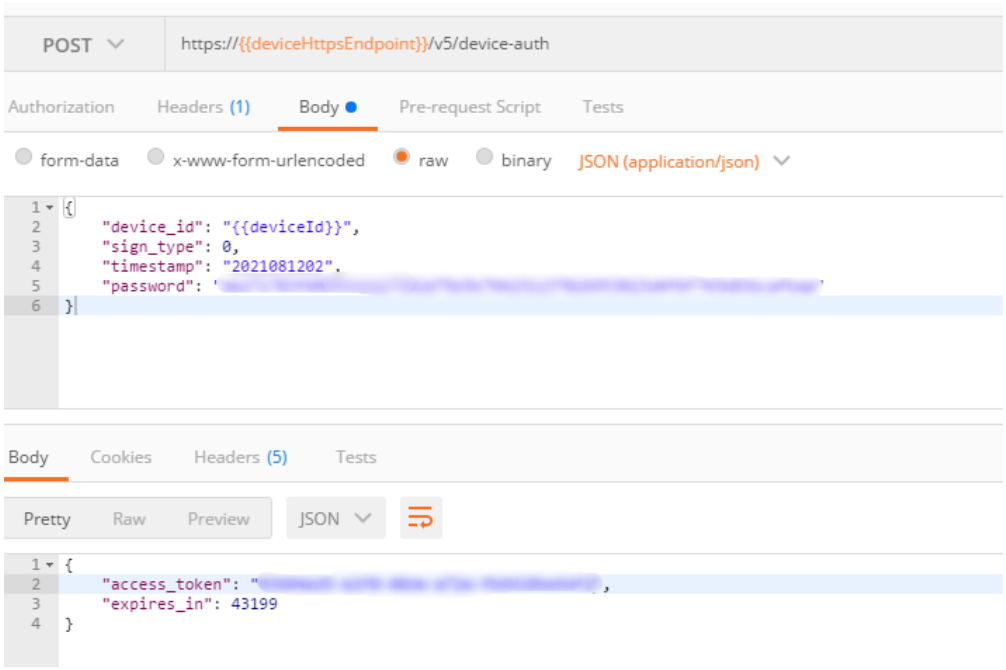


Figure 3-2 Obtaining the access token



c. Obtain the access address by referring to **Platform Connection Information**, combine the access address into a URL by referring to **Figure 3-2**, and send the URL to obtain the access_token.

4. The obtained access token can be used by devices to report messages and properties. The access token is in the message header. The following uses property reporting as an example.

Figure 3-3 Reporting properties

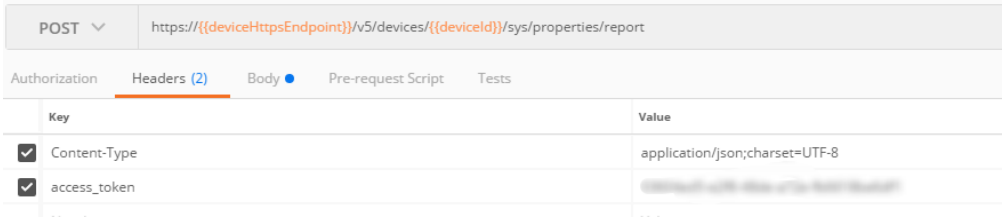
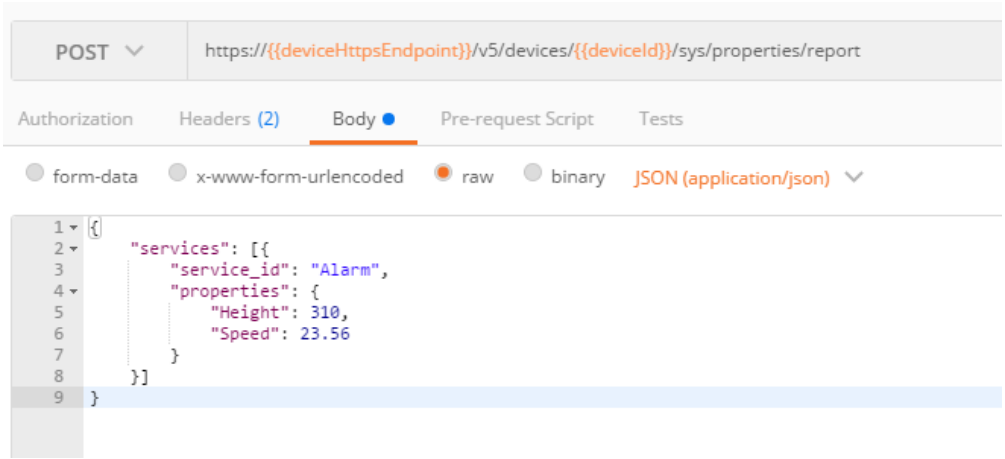


Figure 3-4 Reporting properties



HTTP APIs

The following table describes the platform APIs.

API Category	Function	API
Device authenticati on API	**Authenticating a device**	/v5/device-auth
Device message reporting API	**Reporting a device message**	/v5/devices/{device_id}/sys/messages/up
Device property reporting APIs	**Reporting device properties**	/v5/devices/{device_id}/sys/properties/report
	Gateway reporting child device properties	/v5/devices/{device_id}/sys/gateway/sub-devices/properties/report

3.2 API Overview

Device Authentication

API	Description
3.3.1 Authenticating a Device	This API is used to authenticate a device. Connections can be established between devices and IoTDA after successful authentication. After the authentication is successful, IoTDA returns an access token. An access token is required when APIs for property reporting and message reporting are called. If an access token expires, you need to authenticate the device again to obtain an access token. If you obtain a new access token before the old one expires, the old access token will be valid for 30 seconds before expiration.

Device Message Reporting

API	Description
3.4.1 Reporting a Device Message	This API is used for a device to report custom data to IoTDA, which then forwards reported messages to an application or other Huawei Cloud services for storage and processing.

Device Reporting Properties

API	Description
3.5.1 Reporting Device Properties	This API is used by a device to report property data in the format defined in the product model to IoTDA.
3.5.2 Gateway Reporting Child Device Properties	This API is used to report device data in batches to IoTDA. A gateway can use this API to report the property data of a maximum of 50 child devices at the same time.

3.3 Device Authentication

3.3.1 Authenticating a Device

Function

This API is used to authenticate a device. Connections can be established between devices and IoTDA after successful authentication. After the authentication is successful, IoTDA returns an access token. An access token is required when APIs for property reporting and message reporting are called. If an access token expires, you need to authenticate the device again to obtain an access token. If you obtain a new access token before the old one expires, the old access token will be valid for 30 seconds before expiration.

URI

Request Method	POST

URI	/v5/device-auth
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory	Type	Location	Description
device_id	Yes	String	Body	Device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by IoTDA. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. The value is a string of no more than 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Value length: 1–128 characters
sign_type	Yes	Integer	Body	Password verification mode. **0**: When the timestamp is verified using the HMAC-SHA256 algorithm, IoTDA does not check whether the message timestamp is consistent with the IoTDA time but only checks whether the password is correct. **1**: When the timestamp is verified using the HMAC-SHA256 algorithm, IoTDA checks whether the message timestamp is consistent with the IoTDA time and then checks whether the password is correct. Value range: 0–1
timestamp	Yes	String	Body	The timestamp is the UTC time when the device was connected to IoTDA, in the format of YYYYMMDDHH. For example, if the UTC time is 2018/7/24 17:56:20, the timestamp is **2018072417**. Value length: a fixed length of 10 characters

Parameter	Mandatory	Type	Location	Description
password	Yes	String	Body	The value of this parameter is the value of the device secret encrypted by using the HMAC-SHA256 algorithm with the timestamp as the key. The device secret is returned by IoTDA upon successful device registration. Value length: a fixed length of 64 characters

Response Parameters

Parameter	Type	Description
access_token	String	Device token, which is used for device authentication. Value length: 32–256 characters
expires_in	Integer	Remaining validity period of the authentication information, in seconds.

Example Request

```
POST https://{endpoint}/v5/device-auth
Content-Type: application/json

{
  "device_id" : "60a87ffebaccd902c2f1abbb_0001",
  "sign_type" : 0,
  "timestamp" : "2019120219",
  "password" : "******"
}
```

Example Response

Status Code: 200 OK

```
Content-Type: application/json

{
  "access_token" : "******",
  "expires_in" : 86399
}
```


Error Code

HTTP Status Code	HTTP Status Code Description	Error Code	Error Message	Error Description
400	Bad Request	IOTDA.000006	Invalid input data.	Invalid request parameters.
401	Unauthorized	IOTDA.000002	The request is unauthorized.	Authentication failed.
403	Forbidden	IOTDA.021101	Request reached the maximum rate limit.	The request frequency has reached the upper limit.
		IOTDA.021102	The request rate has reached the upper limit of the tenant, limit %s.	The request frequency has reached the upper limit of the tenant.

3.4 Device Message Reporting

3.4.1 Reporting a Device Message

Function

This API is used for a device to report custom data to IoTDA, which then forwards reported messages to an application or other Huawei Cloud services for storage and processing.

URI

Request Method	POST
URI	/v5/devices/{device_id}/sys/messages/up
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory	Type	Location	Description
access_token	Yes	String	Header	Access token returned after the device authentication API is called. Value length: 1–256 characters
device_id	Yes	String	Path	Device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by IoTDA. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. The value is a string of no more than 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Value length: 1–128 characters

NOTE

This API allows a device to use a request body to report custom data to IoTDA, which then forwards the body content to an application or other Huawei Cloud services for storage and processing. IoTDA has no specific format requirements on the body content. This API can carry data whose size is smaller than 1 MB.

Example Request

```
POST https://{endpoint}/v5/devices/{device_id}/sys/messages/up
Content-Type: application/json
access_token: ******
{
  "name" : "name",
  "id" : "id",
  "content" : "messageUp"
}
```

Example Response

Status Code: 200 ok

Error Code

HTTP Status Code	HTTP Status Code Description	Error Code	Error Message	Error Description
400	Bad Request	IOTDA.000006	Invalid input data.	Invalid request parameters.
403	Forbidden	IOTDA.000004	Invalid access token.	Invalid token.
		IOTDA.021101	Request reached the maximum rate limit.	The request frequency has reached the upper limit.
		IOTDA.021102	The request rate has reached the upper limit of the tenant, limit %s.	The request frequency has reached the upper limit of the tenant.

3.5 Device Property Reporting

3.5.1 Reporting Device Properties

Function

This API is used by a device to report property data in the format defined in the product model to IoTDA.

URI

Request Method	POST
URI	/v5/devices/{device_id}/sys/properties/report
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory	Type	Location	Description
access_token	Yes	String	Header	Access token returned after the device authentication API is called. Value length: 1–256 characters
device_id	Yes	String	Path	Device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by IoTDA. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. The value is a string of no more than 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Value length: 1–128 characters
services	Yes	List<**ServiceProperty**>	Body	Device service data list.

Table 3-2 ServiceProperty

Parameter	Mandatory	Type	Description
service_id	Yes	String	Service ID of the device.
properties	Yes	Object	Service properties, which are defined in the product model of the device.
event_time	No	String	UTC time when the device collects data. The format is yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2021-08-13T10:10:10.555Z. If the data reported by a device does not contain this parameter or the parameter format is incorrect, the data reporting time is the platform time. The format is yyyyMMdd'T'HHmmss'Z', for example, 20230523T014506Z.

Example Request

```
POST https://{endpoint}/v5/devices/{device_id}/sys/properties/report
Content-Type: application/json
access_token: ******

{
  "services" : [ {
    "service_id" : "serviceId",
    "properties" : {
      "Height" : 124,
      "Speed" : 23.24
    },
    "event_time" : "2021-08-13T10:10:10.555Z"
  } ]
}
```

Example Response

If the status code is 200, reporting is successful.

Error Code

HTTP Status Code	HTTP Status Code Description	Error Code	Error Message	Error Description
400	Bad Request	IOTDA.000006	Invalid input data.	Invalid request parameters.
		IOTDA.021104	Subdevices in the request does not exist or does not belong to the gateway.	Some child devices in the request do not exist or do not belong to the gateway.
403	Forbidden	IOTDA.000004	Invalid access token.	Invalid token.
		IOTDA.021101	Request reached the maximum rate limit.	The request frequency has reached the upper limit.
		IOTDA.021102	The request rate has reached the upper limit of the tenant, limit %s.	The request frequency has reached the upper limit of the tenant.
		IOTDA.021105	The content reported in a single request cannot exceed 1 MB.	The content reported in a single request cannot exceed 1 MB.

3.5.2 Gateway Reporting Child Device Properties

Function

This API is used to report device data in batches to IoTDA. A gateway can use this API to report the property data of a maximum of 50 child devices at the same time.

URI

Request Method	POST
URI	/v5/devices/{device_id}/sys/gateway/sub-devices/properties/report
Transport Protocol	HTTPS

Request Parameters

Parameter	Mandatory	Type	Location	Description
access_token	Yes	String	Header	Access token returned after the device authentication API is called. Value length: 1–256 characters
device_id	Yes	String	Path	Device ID, which uniquely identifies a device. The value of this parameter is specified during device registration or allocated by IoTDA. If the value is allocated by the platform, the value is in the format of *[product_id]_[node_id]*. The value is a string of no more than 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Value length: 1–128 characters
devices	Yes	List<**DeviceProperty**>	Body	Device data list. Value length: a maximum of 50 characters

Table 3-3 DeviceProperty

Parameter	Mandatory	Type	Description
device_id	Yes	String	ID of the child device, which is unique and is allocated by IoTDA during device registration. The value is a string of no more than 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
services	Yes	List<**ServiceProperty**>	Device service data list.

Table 3-4 ServiceProperty

Parameter	Mandatory	Type	Description
service_id	Yes	String	Service ID of the device.
properties	Yes	Object	Service properties, which are defined in the product model of the device.
event_time	No	String	UTC time when the device collects data. The format is yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example, 2021-08-13T10:10:10.555Z. If the data reported by a device does not contain this parameter or the parameter format is incorrect, the data reporting time is the platform time. The format is yyyyMMdd'T'HHmmss'Z', for example, 20230523T014506Z.

Example Request

```
POST https://{endpoint}/v5/devices/{device_id}/sys/gateway/sub-devices/properties/report
Content-Type: application/json
access_token: ******

{
  "devices" : [ {
    "device_id" : "deviceId_0001",
    "services" : [ {
      "service_id" : "serviceId",
      "properties" : {
        "Height" : 124,
        "Speed" : 23.24
      },
```



```
    "event_time" : "2021-08-13T10:10:10.555Z"
  } ]
 }, {
  "device_id" : "deviceId_0002",
  "services" : [ {
   "service_id" : "serviceId",
   "properties" : {
    "Height" : 124,
    "Speed" : 23.24
   },
   "event_time" : "2021-08-13T10:10:10.555Z"
  } ]
 } ]
}
```

Example Response

If the status code is 200, reporting is successful.

Error Code

HTTP Status Code	HTTP Status Code Description	Error Code	Error Message	Error Description
400	Bad Request	IOTDA.000006	Invalid input data.	Invalid request parameters.
		IOTDA.021104	Subdevices in the request does not exist or does not belong to the gateway.	Some child devices in the request do not exist or do not belong to the gateway.
403	Forbidden	IOTDA.000004	Invalid access token.	Invalid token.
		IOTDA.021101	Request reached the maximum rate limit.	The request frequency has reached the upper limit.
		IOTDA.021102	The request rate has reached the upper limit of the tenant, limit %s.	The request frequency has reached the upper limit of the tenant.
		IOTDA.021103	The request batch properties number has reached the upper limit, limit %s.	The number of child devices in the request reaches the upper limit.
		IOTDA.021105	The content reported in a single request cannot exceed 1 MB.	The content reported in a single request cannot exceed 1 MB.

4 LwM2M API Reference on the Device Side

4.1 Using LwM2M For Access

Overview

LwM2M, proposed by the Open Mobile Alliance (OMA), is a lightweight, standard, and universal IoT device management protocol that can be used to quickly deploy IoT services in client/server mode. LwM2M establishes a set of standards for IoT device management and application. It provides lightweight, compact, and secure communication interfaces and efficient data models for M2M device management and service support. The IoT platform supports encrypted and non-encrypted access. Port 5684 and CoAP over DTLS are used for encrypted service data exchange and access. Port 5683 and CoAP are used for non-encrypted access.

NOTE

For details about LwM2M syntax and APIs, see **specifications**.

The IoT platform supports the plain text, opaque, Core Link, TLV, and JSON encoding formats specified in the protocol. In the multi-field operation (for example, writing multiple resources), the TLV format is used by default.

Limitations

Table 4-1 Limitations

Item	Limit
Supported LwM2M version	1.1
Supported DTLS version	DTLS 1.2
Supported cryptographic algorithm suite	TLS_PSK_WITH_AES_128_CCM_8 and TLS_PSK_WITH_AES_128_CBC_SHA256
Maximum body length	1 KB
API specifications	See **Specifications**.

Description

For details about the platform endpoint, see **Platform Connection Information**.

NOTE

Use the endpoint corresponding to CoAP (5683) or CoAPS (5684) and port 5683 (non-encrypted) or 5684 (encrypted) for device access.

4.2 API Overview

LwM2M Object Resources Supported by the IoT Platform

Table 4-2

Path	Object	Resource	Function
/rd?ep={nodeId}	Device	Register	Connecting devices to the platform
/3/0/3	Device	Firmware Version	Querying the firmware version
/4/0/0	Connectivity monitoring	Network Bearer	Identifying the type of the network to which the device is connected
/4/0	Connectivity monitoring	Network Signal Strength	Querying the signal level
/4/0/8	Connectivity monitoring	Cell ID	Identifying a cell

Path	Object	Resource	Function
/5/0/1	Firmware Update	Package URI	Delivering the firmware download URL
/5/0/2	Firmware Update	Update	Pushing a firmware upgrade notification
/5/0/3	Firmware Update	State	Pushing a firmware upgrade status notification
/5/0/5	Firmware Update	Update Result	Querying the firmware upgrade result
/19/0/0	BinaryAppDataContainer	Data	Upstream service message transmission
/19/1/0	BinaryAppDataContainer	Data	Downstream service data transmission

NOTE

BinaryAppDataContainer object:

To simplify LwM2M object use for device manufacturers, Huawei defines this object, which is implemented by chips or modules. Devices only need to call AT commands or functions to send and receive service data, regardless of the LwM2M protocol. For details, see **Specifications**.

Authenticating a Device

API	Description
4.3 Authenticating a Device	When a device registers with the IoT platform, the platform authenticates the device. The platform subscribes to resources from the device. The device obtains the token delivered by the IoT platform for subscription. Token information must be carried when other APIs (for example, the API for property reporting) are called.

Reporting Device Properties

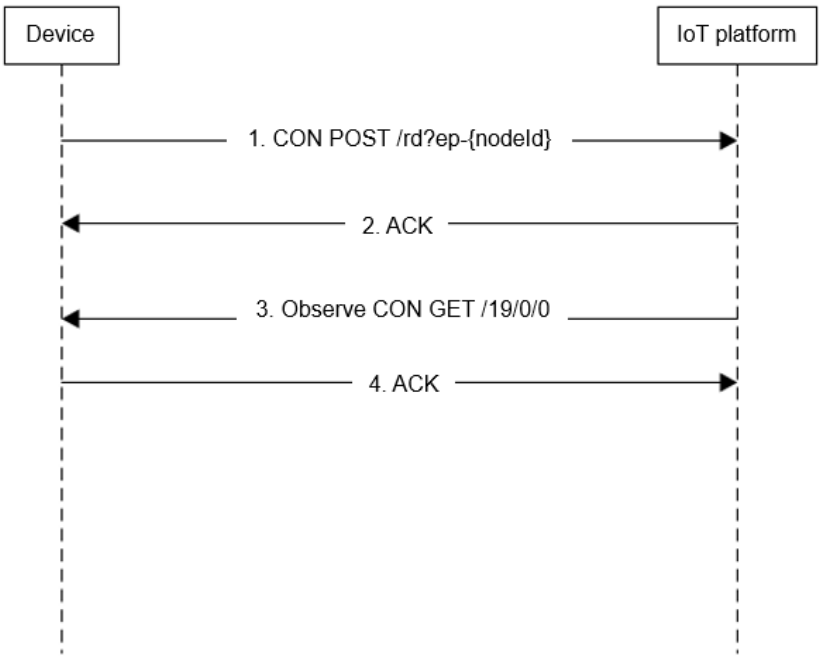
API	Description
4.4 Reporting Device Properties	A device reports properties to the IoT platform by carrying the token delivered by the IoT platform for /19/0/0 subscription.

Delivering a Command

API	Description
4.5 Delivering a Command	The IoT platform delivers a command to a device by packing the command into the payload of the LwM2M write message through the /19/1/0 resource object.

4.3 Authenticating a Device

Process Description



LwM2M Object Resource

When a device registers with the IoT platform, the platform authenticates the device.

Operation	CoAP Method	URI	Success	Failure
Register	POST	/rd?ep={nodeId}	2.01 Created	4.00 Bad Request, 4.03 Forbidden, 4.12 Precondition Failed

The platform subscribes to resources from the device. The device obtains the token delivered by the IoT platform for subscription. Token information must be carried when other APIs (for example, the API for property reporting) are called.

Operation	CoAP Method	URI	Success	Failure
Observe	GET with Observe option = 0	/19/0/0	2.04 Content with Observe option	4.00 Bad Request, 4.04 Not Found, 4.01 Unauthorized, 4.05 Method Not Allowed

Request Parameters

Parameter	Mandatory	Type	Location	Description
nodeId	Yes	String	Uri-Query	Node ID, used to uniquely identify a device. This parameter is specified during device registration on the IoT platform.

Register Request Example

```
CON-POST    MID=25995, Token=514078a73366, OptionSet={"Uri-Path":"rd", "Content-Format":"application/octet-stream", "Uri-Query":["ep=test"]}
```

Register Response Example

```
ACK-2.01    MID=25995, Token=514078a73366, OptionSet={"Location-Path":["rd","test"]}, no payload:
```

Observe Request Example

```
CON-GET     MID=48590, Token=2cb6a673cba24c04, OptionSet={"Observe":0, "Uri-Path":["19","0","0"], "Accept":"application/octet-stream"}, no payload
```

Observe Response Example

```
ACK-2.04    MID=48590, Token=2cb6a673cba24c04, OptionSet={"Observe":0, "Content-Format":"application/octet-stream"}
```


4.4 Reporting Device Properties

LwM2M Object Resource

A device reports properties to the IoT platform by carrying the token delivered by the IoT platform for /19/0/0 subscription.

Operation	CoAP Method	URI	Success	Failure
Notify	Asynchronous Response	N/A	2.05 Content with {value}	N/A

NOTE

Data reported by devices using LwM2M is binary packet data and needs to be parsed using codecs. For details, see **Developing a Codec**.

Request Parameters

Parameter	Location	Mandatory	Description
value	Payload	Mandatory	Data reported by the device.

Request Example

If the value is **c4 0d 5a 6e 96 0b c3 0e 2b 30 37**, the report example is as follows:
NON-2.05 MID=48590, Token=2cb6a673cba24c04, OptionSet={"Observe":22, "Content-Format":"application/octet-stream"}, c4 0d 5a 6e 96 0b c3 0e 2b 30 37

4.5 Delivering a Command

LwM2M Object Resource

The IoT platform delivers an asynchronous command to a device by packing the command into the payload of the LwM2M write message through the /19/1/0 resource object.

Operation	CoAP Method	URI	Success	Failure
Write	PUT	/19/1/0	2.04 Changed	4.00 Bad Request, 4.04 Not Found, 4.01 Unauthorized, 4.05 Method Not Allowed, 4.06 Not Acceptable

NOTE

1. For devices connected using LwM2M, IoT platform supports only asynchronous command delivery.

2. After receiving the command, the IoT platform calls the codec you upload to the platform to encode the command and delivers the encoded command to the device. For details, see **Developing a Codec**.

Request Example

If the encoded value is **c4 0d 5a 6e 96 0b c3 0e 2b 30 37**, the delivery example is as follows:

```
CON-PUT    MID=48587, Token=2d82e7f54b, OptionSet={"Uri-Path":["19","1","0"], "Content-Format":"application/octet-stream"}, c4 0d 5a 6e 96 0b c3 0e 2b 30 37
```

Response Example

```
ACK-2.04   MID=48587, Token=2d82e7f54b, OptionSet={"Content-Format":"application/octet-stream"}, no payload
```


5 Module AT Command Reference

5.1 AT Command List

For modules that have passed the compatibility certification, its AT commands and format specifications are basically the same as Huawei general requirements. Some module vendors have their own AT channels and their implementation is slightly different. For details, see the special description of the module vendors.

AT Command	Description
AT+HMVER	Obtains the Huawei SDK version information.
AT+HMCON	Sets MQTT connection parameters.
AT+HMDIS	Disables the connection to the Huawei IoT platform.
AT+HMPUB	Sends MQTT data to a specific topic.
+HMREC	Transmits the data received by the module to external MCUs.

AT Command	Description
+HMSTS	Proactively transfers the module connection or disconnection status to external MCUs.
AT+HMSUB	Subscribes to a custom topic.
AT+HMUNS	Unsubscribes to a custom topic.
AT+HMPKS	Sets the server or client certificate.

5.2 AT+HMVER

This command is used to obtain the version information about the Huawei SDK.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT +HMVER	-	-	+HMVER:vx.x.x AT TIME ON DATE	AT+HMVER

5.3 AT+HMCON

This command is used to set MQTT connection parameters.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT +HMCON=bsmode,lifetime,serverip,serverport,deviceid,passwd,codec	**bsmode**: The value **0** indicates the non-BS mode.	Default value: **0**	-	AT +HMCON=0,10,"iot-mqtts.cn-north-4.myhuaweicloud.com","8883","testID","testPasswd",0
	lifetime: MQTT heartbeat keep-alive duration. You are advised to set it to a value greater than or equal to 30.	Default value: **300**	+HMCON OK (Connection successful)	
	serverip: IP address of the device access server or bootstrap server.	Default value: IP address of the bootstrap server	+HMCON ERR: code (Connection failed. **code** indicates the failure cause code.)	

Command	Parameter	Default Parameter Processing	AT Response	Example
	serverport: port number of the device access server or bootstrap server.	Default value: MQTTS port 8883	-	
	id: device ID, which contains a maximum of 256 characters.	Obtained after a device is registered.	-	
	pwd: device secret, which contains a maximum of 256 characters.	Obtained after a device is registered.	-	
	codec: payload encoding mode. The value **0** indicates the original encoding mode, and the value **1** indicates the hexadecimal encoding mode.	Default value: **0**	-	

5.4 AT+HMDIS

This command is used to disconnect the connection to the Huawei IoT platform.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT+HMDIS	-	-	+HMDIS OK (Disconnection successful) +HMDIS ERR: code (Disconnection failed. **code** indicates the failure cause.)	AT+HMDIS

5.5 AT+HMPUB

This command is used to send MQTT data to a specific topic.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT +HMPUB=qos,topic,payload_len,payload	**qos**: MQTT QoS (**0**, **1**, **2**)	Default value: **0**	+HMPUB OK (Publishing successful)	If the hexstring mode is used, the data in bytes must be converted into the hexadecimal value before being transmitted. For example, to send bytes 0x12 0x34 to **test**, invoke the command **AT +HMPUB=0,"test",2,1234**.
	topic: specific topic.	The default value is the topic of reporting properties. You are advised to set this parameter.	+HMCON ERR: code (Publishing failed. **code** indicates the failure cause code.)	
	len: length of the data to be sent, in bytes.	-	-	
	payload: data converted to ASCII code. The maximum length is 1,024 bytes.	The maximum length is 2 KB.	-	

5.6 +HMREC

This command is used to transmit the data received by the module to external MCUs.

Command	Parameter	Default Parameter Processing	AT Response	Example
+HMREC=topic,payload_len,payload	**topic**: specific topic.	-	-	+HMREC="$oc/devices/my_deviceid/user/my_subtopic",2,0102 (hexmode)
	len: length of the data received, in bytes.	-	-	
	payload: data converted to ASCII code.	-	-	

5.7 +HMSTS

This command is used to proactively transfer the module connection or disconnection status to external MCUs.

Command	Parameter	Default Parameter Processing	AT Response	Example
+HMSTS: status	**status**: module connection status. The value **0** indicates that the module is connected, and the value **1** indicates that the module is disconnected. If the communication is interrupted due to a network fault, the module attempts to reconnect to the network and subscribe to the topics that have been subscribed previously.	-	-	+HMSTS:0 indicates that the module has been connected to the IoT platform. +HMSTS:1 indicates that the module is disconnected from the IoT platform due to a network fault.

5.8 AT+HMSUB

This command is used to subscribe to a custom topic.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT+HMSUB=qos,topic	**qos**: topic QoS.	Default value: **0**	+HMSUB OK when success	AT+HMSUB=0,"$oc/devices/my_deviceid/user/my_subtopic"

Command	Parameter	Default Parameter Processing	AT Response	Example
	topic: custom topic.	Custom	+HMSUB ERR:code code:reference to en_oc_mqtt_err_code_t defines	

5.9 AT+HMUNS

This command is used to unsubscribe to a custom topic.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT+HMUNS=topic	topic: custom topic.	Custom	+HMUNS OK when success +HMUNS ERR:code code:reference to en_oc_mqtt_err_code_t defines	AT+HMUNS="$oc/devices/my_deviceid/user/my_subtopic"

5.10 AT+HMPKS

This command is used to set the server or client certificate.

Command	Parameter	Default Parameter Processing	AT Response	Example
AT +HMPKS =type, para1, [para2]	**type**: The value **0** indicates the platform certificate. The certificate content is specified by **para1**. **type**: The value **1** indicates the public key certificate of a device. The certificate content is specified by **para1**. **type**: The value **2** indicates the private key certificate of a device. The certificate content is specified by **para1**, and the password is specified by **para2**. **para1** is used to store a certificate. If this parameter is left blank, the certificate is to be deleted. **para2** is used to store the password of a private key certificate. It is valid only when a private key certificate is set and the certificate (a PEM file) is transmitted in the character string format.	Custom	+HMPKS OK when success +HMPKS ERR when failed	AT+HMPKS =0,"SERVER_CA" The certificate cannot contain ".

6 Change History

Release Date	Description
2024-06-25	This issue is the twenty-first official release, which incorporates the following change: Newly added: • 1.4.25.1 Create a Device Policy • 1.4.25.2 Query the Device Policy List • 1.4.25.3 Delete a Device Policy • 1.4.25.4 Query Device Policy Details • 1.4.25.5 Update Device Policy Information • 1.4.25.6 Bind a Device Policy • 1.4.25.7 Unbind a Device Policy • 1.4.25.8 Query the List of the Targets Bound to a Device Policy • 1.4.26.1 Create a Pre-provisioning Template • 1.4.26.2 Query the Pre-provisioning Template List • 1.4.26.3 Delete a Pre-provisioning Template • 1.4.26.4 Query Pre-provisioning Template Details • 1.4.26.5 Update Information About a Pre-provisioning Template with a Specified ID • 1.4.27.1 Create a Custom Authenticator • 1.4.27.2 Query the Custom Authenticator List • 1.4.27.3 Delete a Custom Authenticator • 1.4.27.4 Query Custom Authenticator Details • 1.4.27.5 Update a Custom Authenticator with a Specified ID

Release Date	Description
2024-05-30	This issue is the twentieth official release, which incorporates the following change: Newly added: • **1.4.24.1 Create a Bridge** • **1.4.24.2 Query the Bridge List** • **1.4.24.3 Delete a Bridge** • **1.4.24.4 Reset the Bridge Secret** • **1.4.17.4 Update a CA Certificate** • **1.4.2.11 Query the List of Device Groups to Which a Specified Device Is Added**
2024-04-26	This issue is the nineteenth official release, which incorporates the following change: Newly added: • **1.4.23.1 Create a Device Proxy** • **1.4.23.2 Query Device Proxy List** • **1.4.23.3 Query Device Proxy Details** • **1.4.23.4 Modify a Device Proxy** • **1.4.23.5 Delete a Device Proxy**
2024-04-11	This issue is the eighteenth official release, which incorporates the following change: Newly added: • **1.4.21.1 Create a Data Transfer Stacking Policy** • **1.4.21.2 Query the Data Transfer Stacking Policy List** • **1.4.21.3 Modify a Data Transfer Stacking Policy** • **1.4.21.4 Query the Data Transfer Stacking Policy** • **1.4.21.5 Delete a Data Transfer Stacking Policy** • **1.4.22.1 Create a Data Forwarding Flow Control Policy** • **1.4.22.2 Query a List of Data Forwarding Flow Control Policies** • **1.4.22.3 Modify a Data Forwarding Flow Control Policy** • **1.4.22.4 Query a Specified Data Forwarding Flow Control Policy** • **1.4.22.5 Delete a Specified Data Forwarding Flow Control Policy**

Release Date	Description
2024-03-29	This issue is the seventeenth official release, which incorporates the following change: Newly added: • **Updating Resource Spaces** Modified: • Added the **task_mode** parameter to support the gateway mode for software/firmware upgrade tasks, and the **task_ext_info** parameter to support additional extended information for batch tasks. Involved APIs: **Creating a Batch Task**, **Querying the Batch Task List**, and **Querying a Batch Task**.
2024-03-13	This issue is the sixteenth official release, which incorporates the following changes: Modified: Added return parameters: **provision_enable** determines whether the self-registration capability is enabled for the CA certificate; **template_id** indicates the ID of the associated template after the self-registration capability is enabled. Involved APIs: **Uploading a Device CA Certificate** and **Obtaining the Device CA Certificate List**.
2024-02-27	This issue is the fifteenth official release, which incorporates the following changes: Modified: Changed the **message_content** parameter in the **ActionSmnForwarding** to an optional parameter, changed the value range of the **message_title** parameter, and added the **message_template_name** parameter to provide the capability of selecting an SMN template. Involved APIs: **Creating a Rule**, **Querying the Rule List**, **Modifying a Rule**, and **Querying a Rule**.
2023-01-05	This issue is the fourteenth official release, which incorporates the following changes: Modified: • Added the **force_disconnect** parameter to specify whether to forcibly disconnect the northbound MQTT/AMQP connection when an access credential is generated. Involved API: **Generating an Access Credential**.

Release Date	Description
2023-12-06	This issue is the thirteen official release, which incorporates the following changes: Modified: • Added the **secret_type** parameter to set or update a sub secret of the device. Involved API: **Resetting a Device Secret**. • Added the **fingerprint_type** parameter to set or update a sub fingerprint of the device. Involved API: **Resetting a Device Fingerprint**. • Added response parameters **secondary_secret** and **secondary_fingerprint** to query a sub fingerprint or sub secret of a device. Involved APIs: **Querying a Device**, **Creating a Device**, and **Modifying a Device**.
2023-11-15	This issue is the twelfth official release, which incorporates the following changes: Involved APIs: **Creating a Rule Action**, **Modifying a Rule Action**, and **Querying the Rule Action List**. Modified content: • Modified the enumerated values of the **mechanism** parameter in the **dms_kafka_forwarding** structure, and added the **security_protocol** parameter to support complete Kafka security authentication. • Added the **signature_enable** and **token** parameters to the **http_forwarding** structure to provide diversified security authentication capabilities for HTTP push.
2023-10-23	This issue is the eleventh official release, which incorporates the following changes: Modified: Added the input parameter **product_name** to support product name search. Involved API: **Querying the Product List**.
2023-07-21	This issue is the tenth official release, which incorporates the following changes: Newly added: • **Deleting a Batch Task** • **Creating a Device Tunnel** • **Querying All Tunnels of a Device** • **Querying Device Tunnels** • **Disable a Device Tunnel** • **Deleting a Device Tunnel**

Release Date	Description
2023-06-17	This issue is the ninth official release, which incorporates the following changes: Newly added: • **Retrying a Batch Task** • **Stopping a Batch Task** Modified: Added the response parameters **error_code** and **error_msg** of the abnormal device behavior during synchronization, for example, a response times out. Involved APIs: **Synchronous Device Command** and **Modifying Device Properties**.
2023-05-13	This issue is the eighth official release, which incorporates the following changes: Modified: Added the **group_type** and **dynamic_group_rule** parameters to support dynamic device groups. Involved APIs: **Adding a Device Group**, **Modifying a Device Group**, **Querying the Device Group List**, and **Querying a Device Group**.
2023-04-19	This issue is the seventh official release, which incorporates the following changes: Newly added: The API for **Delivering Broadcast Messages** Modified: Modified the **task_type** parameter and added the enumerated value **updateDevices** to support batch device update tasks. Involved APIs: **Creating a Batch Task**, **Querying the Batch Task List**, and **Querying a Batch Task**.
2023-03-29	This issue is the sixth official release, which incorporates the following changes: Modified: Added the **device_linkage_status_condition** parameter to support the device linkage rule condition of device status changes. Involved APIs: **Creating a Rule**, **Modifying a Rule**, **Querying the Rule List**, and **Querying a Rule**.

Release Date	Description
2023-02-24	This issue is the fifth official release, which incorporates the following changes: Newly added: • **Creating an OTA Upgrade Package** • **Querying the OTA Upgrade Package List** • **Querying OTA Upgrade Package Details** • **Deleting an OTA Upgrade Package** Modified: Added the response parameters **connection_status_update_time** and **active_time** to query the connection status change time and activation time of a device. Involved API: **Querying a Device**.
2023-02-07	This issue is the fourth official release, which incorporates the following changes: Modified: Added the **dms_rocketmq_forwarding**, **mrs_kafka_forwarding**, **roma_forwarding**, and **influxdb_forwarding** parameters to support message transfer of RocketMQ, Kafka, ROMA, and InfluxDB. Involved APIs: **Creating a Rule Action**, **Modifying a Rule Action**, and **Querying the Rule Action List**.
2022-12-14	This issue is the third official release, which incorporates the following changes: Newly added: The API for **Querying Device List Flexibly** Modified: Added the enumerated value **DEVICE_SIDE** for the **rule_type** parameter to support linkage rules on the device. Involved APIs: **Creating a Rule**, **Modifying a Rule**, **Querying the Rule List**, and **Querying a Rule**.
2022-09-20	This issue is the second official release, which incorporates the following changes: Added the **properties** parameter to support MQTT 5.0. Involved APIs: **Delivering a Device Message** and **Querying Device Messages**.
2022-06-30	This is the first official release. **1 API Reference on the Application Side** **2 MQTT or MQTTS API Reference on the Device Side** **3 HTTPS API Reference on the Device Side** **5 Module AT Command Reference**

