**Equipment Management Service Enterprise Edition**

Version:V1.5.2

2018-06-11

Document revision record

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Standard revision history record | | | | | | | | | |
| Version | Data | | Reviser | Revision page | | | Revise the content | | |
| 1.0 | 2017-03-28 | | LiuXinxin |  | | | New document | | |
| 1.0 | 2017-04-07 | | DongYuegang |  | | | The command issuing interface adds callbackUrl and usn | | |
| 1.1 | 2017-05-31 | | DongYuegang |  | | | Increase the operation response data structure description | | |
| 1.2 | 2017-07-10 | | LiuXinxin |  | | | Revised document, 3.3~3.5 PUT changed to POST | | |
| 1.3 | 2017-08-16 | | LiuXinxin |  | | | Increase the callback return value of the operation, resCode=0 is the device successfully executed | | |
| 1.3 | 2017-09-06 | | Li Daqin |  | | | Add Appendix 4.4 parameters and return values | | |
| 1.4 | 2017-11-01 | | JiaShanshan |  | | | Add 4 interfaces: user equipment operation - control channel, user read attribute - asynchronous interface, user write attribute - asynchronous interface, user equipment operation - asynchronous interface | | |
| 1.5 | 2017-12-11 | | JiaShanshan |  | | | Add control class resCode error code list | | |
| 1.5.1 | 2018-05-07 | | JiaShanshan |  | | | Add a feature description | | |
| 1.5.2 | 2018-06-11 | | JiaShanshan |  | | | Modify domain access requirements | | |
|  | | | | | | | | | |
| Business entity | | Signing the opinion | | | | Signature | | | Date |
|  | |  | | | |  | | |  |
|  | |  | | | |  | | |  |
|  | |  | | | |  | | |  |
|  | |  | | | |  | | |  |
| Establishment | |  | | | | Check | | |  |
| Audit | |  | | | | Approve | | |  |
|  | |  | | | |  | | |  |
|  | |  | | |  | | |  | |
|  | |  | | |  | | |  | |

Directory

[**Equipment Management Service Enterprise Edition** 1](#_Toc520706130)

[1 Introduction 6](#_Toc520706131)

[1.1 Executive Summary 6](#_Toc520706132)

[1.2 Purpose of writing 6](#_Toc520706133)

[1.3 Terms and definitions 6](#_Toc520706134)

[1.4 Reference material 6](#_Toc520706135)

[1.5 Special Symbol Format Description 6](#_Toc520706136)

[2 Public description 6](#_Toc520706137)

[2.1 Basic introduction 7](#_Toc520706138)

[2.2 Call process 7](#_Toc520706139)

[2.3 Access address 7](#_Toc520706140)

[2.4 Interface common part 8](#_Toc520706141)

[2.5 Signature authentication 9](#_Toc520706142)

[2.5.1 Instructions 9](#_Toc520706143)

[2.5.2 Introduction to parameters 9](#_Toc520706144)

[2.5.3 Algorithm 10](#_Toc520706145)

[2.6 Internationalization 10](#_Toc520706146)

[2.7 Data type qualification 10](#_Toc520706147)

[2.7.1 Field type description 10](#_Toc520706148)

[2.7.2 Null value description 10](#_Toc520706149)

[3 Equipment management enterpris edition 12](#_Toc520706150)

[3.1 Public description 12](#_Toc520706151)

[3.2 Public structure 12](#_Toc520706152)

[3.2.1 User 12](#_Toc520706153)

[3.2.2 OpPropertyValue 12](#_Toc520706154)

[3.2.3 OpResult 12](#_Toc520706155)

[OpResult 12](#_Toc520706156)

[3.3 Querying the User Bind to the Device 13](#_Toc520706157)

[3.3.1 Interface definition 13](#_Toc520706158)

[3.3.2 Return code 13](#_Toc520706159)

[3.3.3 Request example 14](#_Toc520706160)

[3.4 Non-standard equipment orders issued 15](#_Toc520706161)

[3.4.1 Interface definition 15](#_Toc520706162)

[3.4.2 Return code 16](#_Toc520706163)

[3.4.3 Request example 16](#_Toc520706164)

[3.5 Query the latest status of the device 17](#_Toc520706165)

[3.5.1 The interface definition 17](#_Toc520706166)

[3.5.2 Rturn code 18](#_Toc520706167)

[3.5.3 Request example 18](#_Toc520706168)

[4 Appendix 19](#_Toc520706169)

[4.1 Common Error Codes 19](#_Toc520706170)

[4.2 International language code table 21](#_Toc520706171)

[4.3 Signature algorithm example 23](#_Toc520706172)

[4.4 About parameter filling and return values 24](#_Toc520706173)

[4.4.1 Non-standard equipment single command important parameters 24](#_Toc520706174)

[4.4.2 Non-standard device group command important parameters 24](#_Toc520706175)

[4.4.3 sn and usn 24](#_Toc520706176)

[4.4.4 retCode and resCode 24](#_Toc520706177)

[4.4.5 Control the list of resCode error codes for the class resCode 25](#_Toc520706178)

1. Introduction
   1. Executive Summary

The Haier U+ cloud platform provides instructions for users to perform command-issuing, read-write operations, and other related device management enterprise services in a server-side manner.

As the basis and input for third-party application development.

* 1. Purpose of writing

Provide the foundation and guidance for third-party application development.

* 1. Terms and definitions

The Haier U+ cloud platform(IoT cloud platform) provides access services and data analysis services for devices.USDK, APP development package, in line with E++ protocol, with the functions of discovering devices, configuring devices to access the Internet, and communicating with devices.

* 1. Reference material

The relevant documents and standards referred to in this design are as follows:

* Signature specification for cloud platform UWS
* Cloud platform Rest interface design specification
* Cloud platform user class interface specification
* Cloud platform composite class interface design
* Cloud platform capability class interface design
* Overseas good air interface manual
  1. Special Symbol Format Description

This document does not use other special document marks or formats.

The interface examples in this article use sample data that is not real data.

1. Public description
   1. Basic introduction

uws provides interfaces for applications to access the U+ cloud platform, and these interfaces and apis are described in the documentation. Readers of this document need to have some technical knowledge of HTTP, HTTPS, REST, RPC, and other technologies.

The uws interface uses the REST-RPC style, and the supported data format is JSON.

The uws interface uses HTTPS one-way certificate authentication for communication to enhance security.

uws will strongly check the parameters of the sent request. If there is any inconsistency with the interface specification, the interface call will be wrong.

uws may increase the number of interface request and response parameters when expanding interface functions in the future. The interface consumer should be able to do this.

* 1. Call process

To use the interface in the document for the App or App Server, follow the process below.

This service is called by APPSERVER. When invoked, APPSERVER needs to follow the following process:

1. Apply for APPID and APPKEY used by APPSERVER (send email to Wang Guanshui f);

2. The type of equipment that APPSERVER can control (send email to Wang Guanshui);

3. Submit the IP white list of APPSERVER (send email to Wu Xiaobo).

* 1. Access address

All interfaces provided in this document only support https protocol requests.

In development and joint debugging, the application developer should connect to the developer environment and connect to different development environments by configuring the application developer to access the router of the external network (setting the router's dns).

Deploy the production environment:

If APPSERVER is deployed on the external network, it is accessed through :

|  |  |  |
| --- | --- | --- |
| Interface classification | Access address(North American environment domain name) | Access address(European environment domain name) |
| Equipment management enterprise edition | [https://uws-gea-us.haieriot.net](https://uws-gea-us.haieriot.net/) | [https://uws-gea-euro.haieriot.net](https://uws-gea-euro.haieriot.net/) |

* 1. Interface common part

In the application interaction with uws, the application needs to pass some fixed parameters in each request header; each response of uws also contains a fixed response code, as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input parameters that need to be passed in the request | | | | | |
| Parameter names | | Type | Location | Required | Instructions |
| appId | | String | Header | Yes | Application ID, within 40 characters,Haier U+ cloud platform globally unique. |
| appVersion | | String | Header | Yes | Apply version up to 32 characters, apply version identification. |
| clientId | | String | Header | Yes | The client ID, which is primarily used to uniquely identify the client (for example, a mobile phone). You can call usdk to get the value of the client ID. |
| sequenceId | | String | Header | Yes | Message flow (Client unique) client transaction serial number. 6-32 bit. It is defined by the client and generated by itself. It is recommended to use date + sequential numbering. |
| accessToken | | String | Header | Yes | Security token, 30 bit character. Users log on Haier U+ cloud platform, which is created by the system. Users exit Haier U+ cloud platform and the system will destroy it. When you are not logged in, access the platform interface that does not require login, and you still need to pass in this parameter. The parameter value can be null or any value (no more than 30 characters). |
| sign | | String | Header | Yes | The signature generated by signing the request is shown in chapter 2.5. |
| timestamp | | long | Header | Yes | Unix timestamp.Accurate to milliseconds. |
| language | | String | Header | Yes | An international logo that represents the language used by the client. The specific identification code is shown in the appendix. |
| timezone | | int | Header | Yes | On behalf of the time zone used by the client, uws does not support internationalization at the moment, fill in 8. |
| Content-Type | | String | Header | Yes | Must be application/json;charset=UTF-8 |
| Output parameters, which are included in the uws response | | | | | |
| Parameter names | Type | | Location | Must return | Instructions |
| retCode | String | | Body | Yes | Return code (where 00000 means the request is successful, others represent errors, error codes and descriptions are listed in the Appendix Error Code Table) |
| retInfo | String | | Body | Yes | Return information for debugging, does not support internationalization, and cannot be directly displayed on the UI. |

* 1. Signature authentication
     1. Instructions

The caller needs to sign the request sent to uws, and the signature value of the signature calculation needs to be assigned to the sign attribute in the Header header (see the public section) for the server to perform signature verification.

* + 1. Introduction to parameters

**The string to be signed is:** url + Body +appId+appKey +timestamp；

**url ：**Refers to the path part of the requested interface address after removing https://domain name+port;

**Body：**Refers to the JSON string after the body part of the application sends the request to remove all whitespace characters. If there is no body, it is an empty string (not null);

**appId：**Properties in Header headers (see the public section description);

**appKey：**The appKey applied to the application on the Haigeek network cannot be sent in clear text;

**timestamp：**Properties in Header headers (see the public section description);

* + 1. Algorithm

The signature algorithm is to calculate the 32-bit lowercase SHA-256 value for the signature string. See the appendix for an example of the algorithm.

* 1. Internationalization

The retCode and retInfo returned by the interface response are not internationalized and are processed by the interface caller.

The internationalization of the interface involving business data is defined by passing the language parameter in the header. The specific international language code is shown in the appendix.

* 1. Data type qualification
     1. Field type description

|  |  |  |  |
| --- | --- | --- | --- |
| Qualified type | instructions | format | Json example |
| DateTime | Date time type string | yyyy-MM-dd hh:mm:ss | {“lgTime”:“2013-10-08 08:00:00”} |
| Date | Date type string | yyyy-MM-dd | {“lgDate”:“2013-10-08”} |
| String | String |  | {“address”:“street 123”} |
| int | int |  | {“age”:1234} |
| long | long |  | {“oid”:1234567890123} |
| double | double |  | {“price”:12.35} |
| boolean | boolean（true or false） |  | {“idOld”:true}。 |

* + 1. Null value description

To avoid parsing errors, the return parameters of each interface of uws do not return a null value.

Required parameters, whether input or output, must have a value and cannot be null.

Non-required parameters are as follows:

Numerical type data（int、long、double）Only numbers are returned, including positive numbers, zeros, and negative numbers.

Boolean type data（boolean）Only return true or false

The above basic types do not themselves contain null values.

DateTime, Date, String, and structure type data. If it is null, the corresponding attribute will not be returned.

E.g:

***DateTime type***

birthday is a DateTime type, not null:

{"name":"Tom","age":23,"birthday":"2013-10-08 08:00:00","address":{"city":"beijing","street":"haidian"} }

When birthday is null, the birthday attribute does not return:

{"name":"Tom","age":23, "address":{"city":"beijing","street":"haidian"} }

***String type***

Name is a String type，not null：

{"name":"Tom","age":23,"birthday":"2013-10-08 08:00:00","address":{"city":"beijing","street":"haidian"} }

When name is null, the name attribute is not returned:

{"age":23,"birthday":"2013-10-08 08:00:00","address":{"city":"beijing","street":"haidian"} }

***Structure type***

address is a struct type, not null:

{"name":"Tom","age":23,"birthday":"2013-10-08 08:00:00","address":{"city":"beijing","street":"haidian"} }

When address is null, the address attribute does not return:

{"name":"Tom","age":23,"birthday":"2013-10-08 08:00:00" }

1. Equipment management enterpris edition
   1. Public description

The enterprise interface header and signature specifications basically follow the specifications in Chapter 2. The difference is that this part of the interface does not require Token check, the accessToken parameter in the header can be passed.

* 1. Public structure
     1. User

|  |  |  |  |
| --- | --- | --- | --- |
| Name | User Info | | User |
| Field name | Type | Instructions | Remarks |
| loginId | String | User name (email address) |  |
| userId | String | User id |  |
| userProfile | Map | User extension attribute |  |

* + 1. OpPropertyValue

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Operation property value | | OpPropertyValue |
| Field name | Type | Instructions | Remarks |
| name | String | Attribute |  |
| value | String | Value |  |

* + 1. OpResult

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Operation result | | OpResult |
| Field name | Type | instructions | Remarks |
| usn | String | Operation serial number |  |
| deviceId | String | Operating device Id |  |
| result | String | Operation response result | Is a base64 code, the result of the decryption of the standard model device is: {"extData":{},"args":[]}, where the data in [] is a plurality of key-value pairs consisting of name and value; The result of the decryption of the standard model device is: {"extData":{},"statuses":[]}, where the data in [] is a plurality of key-value pairs consisting of name and value. |

* 1. Querying the User Bind to the Device
     1. Interface definition

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Interface name | | | Query the user bound to the device | | | | | |
| Access address | | | /udse/v1/devBindUsers | | | | HTTP Method | PUT |
| Functional description | | | User information bound according to device MAC query | | | | | |
| Input parameters | | | | | | | | |
| Parameter name | | Type | | Location | Required | Description | | |
| deviceId | | String | | Body | Yes | The device id | | |
| Output parameters | | | | | | | | |
| Parameter name | Type | | | Location | Required | Description | | |
| users | User[] | | | Body | Yes | user list | | |

* + 1. Return code

|  |  |  |
| --- | --- | --- |
| Error code | Info | Scenario |
| 00000 | Success | Successful operation |
| B00001 | Missing required parameters | appId is empty |
| C00002 | appserver has no access authorization | appserver has no access authorization |
| C00006 | The product configuration information is empty | The product configuration information is empty |
| D00001 | sign signature error | Digital signature error |
| G20202 | The current user does not match the device | The current system does not match the device  The current system does not have permission to operate the device |

* + 1. Request example

|  |  |
| --- | --- |
| Request address | /udse/v1/devBindUsers |
| User request | Header：  appId: SV-GEHWHKQ-0000  appVersion: 99.99.99.99990  clientId: 123  sequenceId: 20161020153428000015  accessToken: TGT3RFHEN0534U172OOYRA0GKCHKI0  sign: 139854d169436e6d91c7b11701b0e2a4bd9152c2005a1fab95dcd60639c3c17d  timestamp: 1490253051551  language: zh-cn  timezone: +8  appKey: 961c447171c19efd78beaef9abc72e7d  Content-Encoding: utf-8  Content-type: application/json  Body  {  "deviceId":"0007A8947D05"  } |
| Request response | {  "retCode": "00000",  "retInfo": "成功!",  "users": [  {  "loginId": "mnxyxxxh2@163.com",  "userId": "838670064340172800"  },  {  "loginId": "mnxyxxxh1@163.com",  "userId": "844794014807883776"  "userProfile": {  "country": "italy",  "nickName": "名字",  "avatar": "123"  }  }  ]  } |
|  |  |

* 1. Non-standard equipment orders issued
     1. Interface definition

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Interface name | | | Non-standard equipment orders issued | | | | | | |
| Access address | | | /udse/v1/devOp | | | | HTTP Method | | PUT |
| Functional description | | | Non-standard equipment command (single command, group command) issued | | | | | | |
| Input parameters | | | | | | | | | |
| Parameter name | | Type | | Location | Required | | Description | | |
| deviceId | | String | | Body | yes | | The device id | | |
| sn | | String | | Body | yes | | Operation serial number. Must be unique | | |
| category | | String | | Body | yes | | Classification of operations.  Single command："AttrOp"  Group command："GroupOp" | | |
| name | | String | | Body | yes | | The name of the operation | | |
| operateCodes | | String | | Body | yes | | Operation command Base64 encrypted value | | |
| Output parameters | | | | | | | | | |
| Parameter name | Type | | | Location | | Required | | Description | |
|  |  | | |  | |  | |  | |

* + 1. Return code

|  |  |  |
| --- | --- | --- |
| Error code | Info | Scenario |
| C00002 | appserver has no access authorization | appserver has no access authorization |
| C00004 | Insufficient operation permission | Size format error |
| C00006 | The product configuration information is empty | The product configuration information is empty |
| D00008 | User is not legal | accessToken error |
| G20202 | Current user does not match the device | The current system does not match the device  The current system does not have permission to operate the device. |
| G03002 | Device out of line | The device is not online and cannot issue commands |

* + 1. Request example

|  |  |
| --- | --- |
| Request address | /udse/v1/devOp |
| User request | Header：  appId:MB-ABC-0000  appVersion:2015110401  clientId:356877020056553-08002700DC94  sequenceId:08002700DC94-15110519074300001  sign:bd4495183b97e8133aeab2f1916fed41  timestamp: 1436236880183  language:zh-cn  timezone:8  Content-type: application/json  Body  {  "deviceId": "0007A893C119",  "sn": "FJIJ2L3-FSFRFGRTWT-HYRH",  "category": "AttrOp",  "name": "221001",  "operateCodes": "eyJ2YWx1ZSI6IjIyMTAwMSJ9"  } |
| Request response | {  "retCode": "00000",  "retInfo": "成功!"  } |
| Operational response data | {  "retCode": "00000",  "retInfo": "成功!",  "usn": "600ce95da3e14fc7a68f483dd14db864",  "deviceId": "0007A893C119",  "result": "ewogICAgImV4dERhdGEiOiB7fSwKICAgICJzdGF0dXNlcyI6IFsKICAgICAgICB7CiAgICAgICAgICAgICJuYW1lIjogIioqKiIsCiAgICAgICAgICAgICJ2YWx1ZSI6ICIqKioiCiAgICAgICAgfSwKICAgICAgICB7CiAgICAgICAgICAgICJuYW1lIjogIioqKiIsCiAgICAgICAgICAgICJ2YWx1ZSI6ICIqKioiCiAgICAgICAgfQogICAgXQp9"，  "resCode":0  } |

* 1. Query the latest status of the device
     1. The interface definition

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Interface name | | | Query the latest status of the device | | | | | | | |
| Access address | | | /udse/v1/devOpStatus | | | | | | HTTP Method | PUT |
| Functional description | | | Support for standard models and non-standard model devices | | | | | | | |
| Input parameters | | | | | | | | | | |
| Parameter name | | Type | | Location | | Required | | Description | | |
| deviceId | | String | | Body | | yes | | Device id | | |
| Output parameters | | | | | | | | | | |
| Parameter name | Type | | | | Location | | Required | Description | | |
| statuses | Map<String,String> | | | | Body | | yes | Device status information, which is uploaded by the module to the cloud platform, and the cloud platform is stored and provided externally. | | |
| timestamp | long | | | | Body | | yes | Operation timestamp | | |

* + 1. Rturn code

|  |  |  |
| --- | --- | --- |
| Error code | Info | Scenario |
| 00000 | Success | Successful operation |
| B00001 | Missing required parameters | appId is empty |
| C00002 | appserver has no access authorization | appserver has no access authorization |
| C00006 | The product configuration information is empty | The product configuration information is empty |
| D00001 | sign signature error | Digital signature error |
| G20202 | The current user does not match the device | The current system does not match the device  The current system does not have permission to operate the device |

* + 1. Request example

|  |  |
| --- | --- |
| Request address | /udse/v1/devOpStatus |
| User request | Header：  appId:MB-ABC-0000  appVersion:2015110401  clientId:356877020056553-08002700DC94  sequenceId:08002700DC94-15110519074300001  sign:bd4495183b97e8133aeab2f1916fed41  timestamp: 1436236880183  language:zh-cn  timezone:8  Content-type: application/json  Body  {  "deviceId": "0007A893C119"  } |
| Request response | {  "retCode": "00000",  "retInfo": "成功!",  "statuses": {  "60200a": "302000",  "202008": "NULL",  "202009": "202009",  "202006": "NULL",  "202007": "202007",  "202004": "NULL",  "20200m": "NULL",  "20200j": "20200j",  "20200k": "NULL"  },  "timestamp": "1490250511728"  } |

1. Appendix
   1. Common Error Codes

|  |  |
| --- | --- |
| **Error code** | **Info** |
| A00001 | Service unavailable |
| A00002 | Network exception |
| A00003 | Access or operation timeout |
| A00004 | Internal system error |
| A00005 | Database access exception |
| A00006 | Unknown exception |
| A00007 | Mail service exception |
| A00008 | Mail delivery failed |
| A00009 | The number of emails sent exceeded |
| B00001 | Missing required parameters |
| B00002 | Parameter type error |
| B00003 | Parameter values are out of range or not enumerated |
| B00004 | The parameters do not meet the rule requirements |
| B00006 | Parameter length error |
| B00007 | The parameters do not match the interface definition |
| C00001 | AppId and appKey validation failed |
| C00002 | AppServer has no access authorization |
| C00003 | Insufficient access |
| C00004 | Insufficient operation permission |
| C00005 | Repeated requests |
| C00006 | Unknown device type |
| C00007 | The appId configuration information is empty |
| C00008 | The appKey is empty |
| D00001 | Sign signature error |
| D00003 | Token does not exist and is not authenticated by Token |
| D00004 | Token has expired and is not authenticated by Token. |
| D00005 | Token is not created from this application and is not authenticated by Token. |
| D00006 | Session invalidation |
| D00007 | Not internal users |
| D00008 | User illegal |

* 1. International language code table

|  |  |  |
| --- | --- | --- |
| Language code | English name | Whether to support |
| af | Afrikaans - South Africa | no |
| ar-ae | Arabic(U.A.E.) | no |
| ar-bh | Arabic(Bahrain) | no |
| ar-dz | Arabic(Algeria) | no |
| ar-eg | Arabic(Egypt) | no |
| ar-iq | Arabic(Iraq) | no |
| ar-jo | Arabic(Jordan) | no |
| ar-kw | Arabic(Kuwait) | no |
| ar-lb | Arabic(Lebanon) | no |
| ar-ly | Arabic(Libya) | no |
| ar-ma | Arabic(Morocco) | no |
| ar-om | Arabic(Oman) | no |
| ar-qa | Arabic(Qatar) | no |
| ar-sa | Arabic(Saudi Arabia) | no |
| ar-sy | Arabic(Syria) | no |
| ar-tn | Arabic(Tunisia) | no |
| ar-ye | Arabic(Yemen) | no |
| be | Belarusian | no |
| bg | Bulgarian | no |
| ca | Catalan | no |
| cs | Czech | no |
| da | Danish | no |
| de | German(Standard) | no |
| de-at | German(Austrian) | no |
| de-ch | German(Swiss) | no |
| de-li | German(Liechtenstein) | no |
| de-lu | German(Luxembourg) | no |
| el | Greek | no |
| en | English | no |
| en-au | English(Australian) | no |
| en-bz | English(Belize) | no |
| en-ca | English(Canadian) | no |
| en-gb | English(British) | no |
| en-ie | English(Ireland) | no |
| en-jm | English(Jamaica) | no |
| en-nz | English(New Zealand) | no |
| en-tt | English(Trinidad) | no |
| en-us | English(United States) | no |
| en-za | English(South Africa) | no |
| es | Spanish(Spain - Modern Sort) | no |
| es-ar | Spanish(Argentina) | 否 |
| es-bo | Spanish(Bolivia) | 否 |
| es-cl | Spanish(Chile) | 否 |
| es-co | Spanish(Colombia) | 否 |
| es-cr | Spanish(Costa Rica) | 否 |
| es-do | Spanish(Dominican Republic) | 否 |
| es-ec | Spanish(Ecuador) | 否 |
| es-gt | Spanish(Guatemala) | 否 |
| es-hn | Spanish(Honduras) | 否 |
| es-mx | Spanish(Mexican) | 否 |
| es-ni | Spanish(Nicaragua) | 否 |
| es-pa | Spanish(Panama) | 否 |
| es-pe | Spanish(Peru) | 否 |
| es-pr | Spanish(Puerto Rico) | 否 |
| es-py | Spanish(Paraguay) | 否 |
| es-sv | Spanish(El Salvador) | 否 |
| es-uy | Spanish(Uruguay) | 否 |
| es-ve | Spanish(Venezuela) | 否 |
| et | Estonian | 否 |
| eu | Basque | 否 |
| fa | Farsi | 否 |
| fi | Finnish | 否 |
| fo | Faeroese | 否 |
| fr | French(Standard) | 否 |
| fr-be | French(Belgian) | 否 |
| fr-ca | French(Canadian) | 否 |
| fr-ch | French(Swiss) | 否 |
| fr-lu | French(Luxembourg) | 否 |
| gd | Gaelic(Scots) | 否 |
| gd-ie | Gaelic (Ireland) | 否 |
| he | Hebrew | 否 |
| hi | Hindi | 否 |
| hr | Croatian | 否 |
| hu | Hungarian | 否 |
| in | Indonesian | 否 |
| is | Icelandic | 否 |
| it | Italian(Standard) | 是 |
| it-ch | Italian(Swiss) | 否 |
| ja | Japanese | 否 |
| ji | Yiddish | 否 |
| ko | Korean | 否 |
| lt | Lithuanian | 否 |
| lv | Latvian | 否 |
| mk | Macedonian | 否 |
| ms | Malaysian | 否 |
| mt | Maltese | 否 |
| nl | Dutch(Standard) | 否 |
| nl-be | Dutch(Belgian) | 否 |
| no | Norwegian(Bokmal) | 否 |
| pl | Polish | 否 |
| pt | Portuguese(Standard) | 否 |
| pt-br | Portuguese(Brazilian) | 否 |
| rm | Rhaeto-Romanic | 否 |
| ro | Romanian | 否 |
| ro-mo | Romanian(Moldavia) | 否 |
| ru | Russian | 是 |
| ru-mo | Russian(Moldavia) | 否 |
| sb | Sorbian | 否 |
| sk | Slovak | 否 |
| sl | Slovenian | 否 |
| sq | Albanian | 否 |
| sr | Serbian(Cyrillic) | 否 |
| sv | Swedish | 否 |
| sv-fi | Swedish(Finland) | 否 |
| sx | Sutu | 否 |
| sz | Sami (Lappish) | 否 |
| th | Thai | 否 |
| tn | Tswana | 否 |
| tr | Turkish | 否 |
| ts | Tsonga | 否 |
| uk | Ukrainian | 否 |
| ur | Urdu | 否 |
| ve | Venda | 否 |
| vi | Vietnamese | 否 |
| xh | Xhosa | 否 |
| zh-cn | Chinese(PRC) | 否 |
| zh-hk | Chinese(Hong Kong SAR, PRC) | 否 |
| zh-sg | Chinese(Singapore) | 否 |
| zh-tw | Chinese(Taiwan Region) | 否 |
| zu | Zulu | 否 |

* 1. Signature algorithm example

String getSign(String appId, String appKey, String timestamp, String body,String url){：

URL urlObj = new URL(url);

url=urlObj.getPath();

appKey = appKey.trim();

appKey = appKey.replaceAll("\"", "");

if (body != null) {

body = body.trim();

}

if (!body.equals("")) {

body = body.replaceAll("", "");

body = body.replaceAll("\t", "");

body = body.replaceAll("\r", "");

body = body.replaceAll("\n", "");

}

log.info("body:"+body);

StringBuffer sb = new StringBuffer();

sb.append(url).append(body).append(appId).append(appKey).append(timestamp);

MessageDigest md = null;

byte[] bytes = null;

try {

md = MessageDigest.getInstance("SHA-256");

bytes = md.digest(sb.toString().getBytes("utf-8"));

} catch (Exception e) {

e.printStackTrace();

}

return BinaryToHexString(bytes);

}

String BinaryToHexString(byte[] bytes) {

StringBuilder hex = new StringBuilder();

String hexStr = "0123456789abcdef";

for (int i = 0; i < bytes.length; i++) {

hex.append(String.valueOf(hexStr.charAt((bytes[i] & 0xF0) >> 4)));

hex.append(String.valueOf(hexStr.charAt(bytes[i] & 0x0F)));

}

return hex.toString();

}

* 1. About parameter filling and return values
     1. Non-standard equipment single command important parameters

category: 'AttrOp',

name: '20g10o',

operateCodes:base64({"value":"30g106"})

* + 1. Non-standard device group command important parameters

category: 'GroupOp',

name: '000001',

operateCodes:base64({"value":[{"name":"20g10o","value":"30g106"},{"name":"20g10m","value":"12"},{"name":"20g10d","value":"30g101"}]})

In the ID document, name may be written as 0x0001, please use 000001 when passing parameters.

* + 1. sn and usn

Sn is generated by the caller and refers to the stream number of the request. When the cloud platform answers the request, it returns the usn, and if a callback is involved, the usn in the callback also means the answer to the request sn.

That is, the complete session is:

request：sn，callback ——> Corresponding to the "user request" in the request example

The cloud platform responds to requests：usn ——> Corresponding to the " Request response " in the request example

Caller callback receives：usn ——> Corresponding to the "Operation response data" in the request example

Sn and usn one-to-one correspondence, please ensure that sn is unique.

* + 1. retCode and resCode

retCode is the cloud platform response to the request, see the return value of each interface and the public return value

resCode is the response of the device to the execution of the command. It appears in the callback. When it is 0, it indicates that the device is successfully executed. Other values are errors. For details, see section 4.4.5 Error Code List or Query ID Document or Consultation. Equipment development engineer.

* + 1. Control the list of resCode error codes for the class resCode

|  |  |  |
| --- | --- | --- |
| resCode | instructions | Remarks |
| 0 | Success; |  |
| 1 | Generic error code that represents a temporarily undefined or undistinguished error. |  |
| 2 | Internal error, which indicates an error that occurred inside the device and was not open to the outside world. |  |
| 11 | Command execution timeout. |  |
| 12 | Invalid command. Mainly refers to command execution logic error. For example, turn it off when it is off. |  |
| 13 | Invalid values. It mainly means that the operation value of the equipment is illegal. For example, the temperature setting is out of bounds. |  |
| 14 | Invalid Token. That is, user Token does not exist in the cloud platform. |  |
| 15 | The user is not bound to the device. |  |
| 16 | The device is not online. |  |
| 17 | Unsupported command. This means that the device does not have this property when reading or writing property requests, or does not support this operation when operating. |  |
| 107 | Invalid command. The command issued is not supported on the device baseboard. | The E++ class device is unique, that is, the invalid command frame of the E++ protocol |
| 200 | An internal error in the cloud platform indicates an error that occurred within the cloud platform and was not open to the outside world. | The cloud platform returns directly, not from the device |

Note: all the above error codes except 200 are returned by the device directly. Please consult the device directly for the processing logic of the error codes returned.