# API接口对接必读

1:保证接口提供方的java服务正常运行；

2:明确设置java服务的具体ip以及java服务运行的端口；

3:接口对接实例:

如创建产品接口 ,接口的地址为/iot/rest/api/product/createProduct，则实际调用为：[http://193.168.3.183:8099/iot/rest/api/product/createProduct](http://192.168.2.183:8099/iot/rest/api/product/createProduct)

其中193.168.3.183 为Java服务的具体ip ,8099 为java服务运行的端口。

4:调用 API 时，除了本文介绍的该 API 的特有请求参数，还需传入公共请求参数。公共请求参数说明，请参照[公共参数](#_5:公共参数)。

5:调用接口会根据请求的参数响应内容，code详细意义见[备注](#_4.备注)。

6:该文档中commodityKey等同于ProductKey, commoditySecret等同于ProductSecret ;

facilityKey用于区分各个设备,facilitySecret等同于DeviceSecret;

commodityKey & facilityKey 作为连接mqtt服务器时的用户名, facilitySecret 作为连接mqtt服务器时的密码

# 1.产品管理

## 概述

产品是设备的集合，通常是一组具有相同功能定义的设备集合。例如，产品指同一个型号的产品，设备就是该型号下的某个设备。

在将设备接入到物联网平台之前，您需要通过接口创建产品。

产品创建成功后，您可通过接口对创建的产品进行管理或者对产品的物模型进行开发，同时可以对产品下的设备进行开发。

## 1.1 创建产品

### **1.1.1 接口描述**

调用此接口在物联网平台的子系统下创建一个产品，将产品的详细信息插入到数据库，用以完整定义新产品的基础属性和类型信息；

调用接口的时候需要明确填写产品所属子系统、产品代码、产品名称、产品键参数，产品描述为选填内容，可为空；产品类型和设备类型按照产品功能进行填写，也为必填选项，将所需参数填写后提交接口，便可成功创建产品；当产品代码等参数在数据库已存在时会返回错误。

备注:

\*只有产品接入方式为网关时,该产品下的设备才会具有子设备管理功能

\*只有产品接入方式为子设备时,该产品下的设备可以被加入到接入方式为网关的设备下

\*产品接入方式为网关子设备时,该产品下的设备也可以用直连接入方式接入

### **1.1.2 服务接入**

| **接口地址** | **/iot/rest/api/product/createProduct** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **1.1.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| childSysId | 子系统编号 | 必填 | integer(int64) | 2 电梯系统3 照明系统5 环控子系统7 站台门系统8 自动售检票9 列车系统10 闭路电视13 sys14 不间断电源 |
| commodityCode | 产品代码 | 必填 | string |  |
| commodityDesc | 产品描述 | 可选 | string |  |
| commodityName | 产品名称 | 必填 | string |  |
| commodityKey | 产品键 | 必填 | string |  |
| nodeType | 产品接入方式 | 必填 | integer(int32) | 1直连2网关3网关子设备 |
| commodityType | 产品类型 | 必填 | integer(int32) | 28垂直电梯29开关30蓄电池60门禁门82广播区域83广播控制盒201车站209隧道 |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 1.1.4 对接示例

#### 请求示例

{

 "childSysId": 6,

 "commodityCode": "api\_ceshi",

 "commodityDesc": "api测试",

 "commodityKey": "api\_commodityKey",

 "commodityName": "api\_commodityName",

 "commodityType": 6,

 "nodeType": 1

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 09:38:46",

 "success": true

}

## 1.2 更新产品

### **1.2.1 接口描述**

可以调用此接口更新物联网平台一个产品的名称和描述，并将改动更新到数据库，调用接口的时候需要填写产品的代码，确保正确更改的前提是产品代码对应的产品是存在的，若不存在，则会返回错误；产品的名称为必填内容，产品描述选填，可为空。

### **1.2.2 服务接入**

| **接口地址** | **/iot/rest/api/product/updateProduct** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **1.2.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| commodityCode | 产品代码 | 必填 | string |  |
| commodityDesc | 产品描述 | 可选 | string |  |
| commodityName | 产品名称 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 1.2.4 对接示例

#### **请求示例**

{

 "commodityCode": "api\_ceshi",

 "commodityDesc": "api测试",

 "commodityName": "api\_commodityName"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 09:40:33",

 "success": true

}

## 1.3 查询产品

### **1.3.1 接口描述**

可以调用此接口查询物联网平台的产品，从数据库中查询产品代码对应的产品的详细信息；调用接口的时候需要填写要查询产品的代码，当产品存在时返回产品的详情，否则返回错误。

### **1.3.2 服务接入**

| **接口地址** | **/iot/rest/api/product/queryProduct** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **1.3.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| commodityCode | 产品代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| appSysDesc | 系统描述 | string |
| appSysId | 系统Id | integer(int32) |
| childSysDesc | 子系统描述 | string |
| childSysId | 子系统Id | integer(int32) |
| commodityCode | 产品代码 | string |
| commodityDesc | 产品描述 | string |
| commodityId | 产品编号 | integer(int64) |
| commodityKey | 产品Key | string |
| commodityName | 产品名称 | string |
| commoditySecret | 产品密钥 | string |
| nodeType | 产品接入方式 | integer(int32) |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| commodityType | 产品类型 | integer(int32) |
| state | 状态 | integer(int32) |

### 1.3.4 对接示例

#### **请求示例**：

{

 "commodityCode": "api\_ceshi"

}

#### **响应示例**:

{

 "code": 200,

 "msg": "成功",

 "data": {

 "appSysId": 3,

 "appSysDesc": "BAS",

 "childSysId": 6,

 "childSysDesc": "给排水系统",

 "commodityId": 22,

 "nodeType": 1,

 "commodityCode": "api\_ceshi",

 "commodityName": "api\_commodityName",

 "commodityDesc": "api测试",

 "commodityKey": "api\_commodityKey",

 "commoditySecret": "4s7fbcb0keoy",

 "commodityType": 6,

 "createTime": "2024-04-02 09:38:46",

 "createBy": 1,

 "state": 0

 },

 "time": "2024-04-02 10:04:14",

 "success": true

}

## 1.4 获取产品列表

### **1.4.1 接口描述**

可以调用此接口查询物联网平台所有的产品列表，在数据库中获取所有的产品及详细信息；此接口为直接调取，不需要任何参数，数据库存在产品则返回产品的列表，否则返回错误。

### **1.4.2 服务接入**

| **接口地址** | **/iot/rest/api/product/queryProductList** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 50次/秒 |

### **1.4.3 数据模型**

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| appSysDesc | 系统描述 | string |
| appSysId | 系统Id | integer(int32) |
| childSysDesc | 子系统描述 | string |
| childSysId | 子系统Id | integer(int32) |
| commodityCode | 产品代码 | string |
| commodityDesc | 产品描述 | string |
| commodityId | 产品编号 | integer(int64) |
| commodityKey | 产品Key | string |
| commodityName | 产品名称 | string |
| commoditySecret | 产品密钥 | string |
| nodeType | 产品接入方式 | integer(int32) |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| commodityType | 产品类型 | integer(int32) |
| state | 状态 | integer(int32) |

### 1.4.4 对接示例

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": [

 {

 "appSysId": 3,

 "appSysDesc": "BAS",

 "childSysId": 6,

 "childSysDesc": "给排水系统",

 "commodityId": 22,

 "nodeType": 1,

 "commodityCode": "api\_ceshi",

 "commodityName": "api\_commodityName",

 "commodityDesc": "api测试",

 "commodityKey": "api\_commodityKey",

 "commoditySecret": "4s7fbcb0keoy",

 "commodityType": 6,

 "createTime": "2024-04-02 09:38:46",

 "createBy": 1,

 "state": 0

 }

 ],

 "time": "2024-04-02 09:41:34",

 "success": true

}

## 1.5 删除产品

### **1.5.1 接口描述**

可以调用此接口删除物联网平台的一个产品，从数据库中删除对应的产品，调用接口的时候需要填写产品的代码，当产品代码存在的时候删除成功，产品代码不存在则响应错误。

### **1.5.2 服务接入**

| **接口地址** | **/iot/rest/api/product/deleteProduct** |
| --- | --- |
| 请求方式 | HTTP / DELETE |
| 调用频率限制 | 100次/秒 |

### **1.5.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| commodityCode | 产品代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 1.5.4 对接示例

#### **请求示例**

{

 "commodityCode": " api\_ceshi "

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 09:43:35",

 "success": true

}

# 2.物模型管理

## 概述

物模型是物理空间中的实体在云端的数字化表示。物模型存在的基础依赖于一个产品，因此必须在产品存在的情况下才能创建成功。

物模型创建成功后，您可通过接口对创建的物模型进行管理或者进行后续的设备开发。

## 2.1 创建物模型

### **2.1.1 接口描述**

可以调用此接口在物联网平台的产品下创建物模型，接受物模型的基本信息并存储到数据库中，产品代码、产品属性代码、产品属性名称、数据类型和点类型为必填项，其它参数为可选值，创建物模型成功的前提是产品代码必须存在，若信息填写错误则返回错误。

### **2.1.2 服务接入**

| **接口地址** | **/iot/rest/api/thingmodel/createThingModel** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **2.1.3 数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| commodityCode | 产品代码 | 必填 | string |  |
| commodityPropertyList | 产品属性集合 | 必填 | array |  |

commodityPropertyList参数说明:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| almpriority | 报警优先级 | 可选 | string | 1 1级紧急报警2 2级严重报警3 3级一般报警4 提示报警9 正常报警10 事项 |
| authorizationType | 读写类型 | 可选 | integer(int32) | 1 读写2只读 |
| commodityPropertyCode | 产品属性代码 | 必填 | string |  |
| commodityPropertyDesc | 产品属性描述 | 可选 | string |  |
| commodityPropertyName | 产品属性名称 | 必填 | string |  |
| ctrlBody | 控制状态相关josn体 | 可选 | string |  |
| dataBody | 数据类型相关json体 | 可选 | string |  |
| dataType | 数据类型 | 必填 | integer(int32) | 0 int32(整数型)1 float(单精度浮点型)2 double(双精度浮点型)3 enum(枚举型)4 bool(布尔型)5 text(字符串)6 Long(长整型) |
| measType | 量测类型 | 可选 | string | 13 AB相有功功率 300 电力开关状态304 电力合位 360 运行状态 361 停止状态 400 信号状态403 网络状态 602 事故总 700 远方/就地 704 工频/变频 712 正常/故障 786 未打开/打开 787 恢复/报警 788 恢复/激活 791 生效/失效 794 自动/屏蔽 813 就地/环控 814 环控/联动 850 球机状态 851 枪机状态 893 合闸/分闸 902 正常/火警 903 正常/激活 904 正常/启动 907 信号消失/复归912 信号消失/分914 储能/未储能 |
| normalState | 正常状态 | 可选 | string | 无0123 |
| pointType | 点类型 | 必填 | integer(int32) | 1 ai2 ao3 di4 do5 acc |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 2.1.4 对接示例

#### **请求示例**

{

 "commodityCode": "api\_ceshi",

 "commodityPropertyList": [

 {

 "almpriority": "3",

 "authorizationType": 1,

 "commodityPropertyCode": "KZMBZT",

 "commodityPropertyDesc": "控制面板状态",

 "commodityPropertyName": "控制面板状态",

 "dataType": 4,

 "dataBody": "{\"0\":\"关\",\"1\":\"开\"}",

 "pointType": 1,

 "ctrlBody": "[{\"ctrlValue\":\"\",\"descValue\":\"\"}]",

 "measType": "400",

 "normalState": ""

 }

 ]

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 09:52:12",

 "success": true

}

## 2.2 更新物模型

### **2.2.1 接口描述**

可以调用此接口更新物联网平台产品下的物模型，将新的物模型的基本信息存储到数据库，产品代码、原产品属性代码、现产品属性代码、产品属性名称、数据类型和点类型为必填项，其它参数为可选值，修改成功的前提是原产品代码必须存在，不存在则返回错误。

### **2.2.2 服务接入**

| **接口地址** | **/iot/rest/api/thingmodel/updateThingModel** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **2.2.3 数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| commodityCode | 产品代码 | 必填 | string |  |
| commodityPropertyList | 产品属性集合 | 必填 | array |  |

commodityPropertyList参数说明:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| almpriority | 报警优先级 | 可选 | string | 1 1级紧急报警2 2级严重报警3 3级一般报警4 提示报警9 正常报警10 事项 |
| authorizationType | 读写类型 | 可选 | integer(int32) | 1 读写2只读 |
| commodityPropertyCode | 产品属性代码 | 必填 | string |  |
| commodityPropertyDesc | 产品属性描述 | 可选 | string |  |
| commodityPropertyName | 产品属性名称 | 必填 | string |  |
| ctrlBody | 控制状态相关josn体 | 可选 | string |  |
| dataBody | 数据类型相关json体 | 可选 | string |  |
| dataType | 数据类型 | 必填 | integer(int32) | 0 int32(整数型)1 float(单精度浮点型)2 double(双精度浮点型)3 enum(枚举型)4 bool(布尔型)5 text(字符串)6 Long(长整型) |
| measType | 量测类型 | 可选 | string | 13 AB相有功功率 300 电力开关状态304 电力合位 360 运行状态 361 停止状态 400 信号状态403 网络状态 602 事故总 700 远方/就地 704 工频/变频 712 正常/故障 786 未打开/打开 787 恢复/报警 788 恢复/激活 791 生效/失效 794 自动/屏蔽 813 就地/环控 814 环控/联动 850 球机状态 851 枪机状态 893 合闸/分闸 902 正常/火警 903 正常/激活 904 正常/启动 907 信号消失/复归912 信号消失/分914 储能/未储能 |
| normalState | 正常状态 | 可选 | string | 无0123 |
| pointType | 点类型 | 必填 | integer(int32) | 1 ai2 ao3 di4 do5 acc |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 2.2.4 对接示例

#### **请求示例**

{

 "commodityCode": "api\_ceshi",

 "commodityPropertyList": [

 {

 "almpriority": "3",

 "authorizationType": 1,

 "commodityPropertyCode": "KZMBZT",

 "commodityPropertyDesc": "控制面板状态",

 "commodityPropertyName": "控制面板状态",

 "dataType": 4,

 "dataBody": "{\"0\":\"关\",\"1\":\"开\"}",

 "pointType": 1,

 "ctrlBody": "[{\"ctrlValue\":\"\",\"descValue\":\"\"}]",

 "measType": "400",

 "normalState": ""

 }

 ]

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 09:54:00",

 "success": true

}

## 2.3 删除物模型

### **2.3.1 接口描述**

可以调用此接口删除物联网平台的产品下的物模型，从数据库中删除物模型的对应内容，产品代码、产品属性代码是必须填写的信息，删除成功的前提是产品代码存在并且产品属性代码也存在，根据信息找到对应的物模型并删除，否则返回错误。

### **2.3.2 服务接入**

| **接口地址** | **/iot/rest/api/thingmodel/deleteThingModel** |
| --- | --- |
| 请求方式 | HTTP / DELETE |
| 调用频率限制 | 100次/秒 |

### **2.3.3 数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| commodityCode | 产品代码 | 必填 | string |  |
| commodityPropertyDeleteList | 产品属性集合 | 必填 | array |  |

commodityPropertyDeleteList参数说明:

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| commodityPropertyCode | 产品属性代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 2.3.4 对接示例

#### **请求示例**

{

 "commodityCode": "api\_ceshi",

 "commodityPropertyDeleteList": [

 {

 "commodityPropertyCode": " KZMBZT "

 }

 ]

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:00:29",

 "success": true

}

## 2.4 查看物模型

### **2.4.1 接口描述**

可以调用此接口查看物联网平台的某个产品下的物模型，从数据库中查询到相应的内容并返回；因此产品代码、产品属性代码是必填项，当数据库中产品代码和属性代码存在，便会返回对应物模型的基本信息，若不存在则会返回错误信息。

### **2.4.2 服务接入**

| **接口地址** | **/iot/rest/api/thingmodel/queryThingModel** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 300次/秒 |

### **2.4.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| commodityCode | 产品代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 |  |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

| **参数名称** | **参数说明** | **类型** |
| --- | --- | --- |
| commodityEventList | 产品事件集合 | array |
| commodityPropertyList | 产品属性集合 | array |
| commodityServiceList | 产品服务集合 | array |

commodityEventList参数说明:

| **参数名称** | **参数说明** | **类型** |
| --- | --- | --- |
| commodityEventId | 产品事件编号 | integer(int64) |
| commodityId | 产品编号 | integer(int64) |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| eventDesc | 描述 | string |
| eventIdentifier | 产品事件标识 | string |
| eventName | 产品事件名称 | string |
| eventType | 事件类型 | integer(int32) |
| responseJsonBody | 输出参数 | string |
| state | 状态 | integer(int32) |

commodityPropertyList参数说明:

| **参数名称** | **参数说明** | **类型** |
| --- | --- | --- |
| almpriority | 报警优先级 | string |
| authorizationType | 读写类型 | integer(int32) |
| commodityPropertyCode | 产品属性代码 | string |
| commodityPropertyDesc | 产品属性描述 | string |
| commodityPropertyId | 产品属性编号 | integer(int64) |
| commodityPropertyName | 产品属性名称 | string |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| ctrlBody | 控制状态相关josn体 | string |
| dataBody | 数据类型相关json体 | string |
| dataType | 数据类型 | integer(int32) |
| measType | 量测类型 | string |
| normalState | 正常状态 | string |
| pointType | 点类型 | integer(int32) |
| state | 状态 | integer(int32) |

commodityServiceList参数说明:

| **参数名称** | **参数说明** | **类型** |
| --- | --- | --- |
| commodityId | 产品编号 | integer(int64) |
| commodityServiceId | 产品服务编号 | integer(int64) |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| invokingType | 调用方式 | integer(int32) |
| requestJsonBody | 输入参数 | string |
| responseJsonBody | 输出参数 | string |
| serviceDesc | 描述 | string |
| serviceIdentifier | 产品服务标识 | string |
| serviceName | 产品服务名称 | string |
| state | 状态 | integer(int32) |

### 2.4.4 对接示例

#### **请求示例**

{

 "commodityCode": "BAS\_ZM\_GATEWAY"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": {

 "commodityPropertyList": [

 {

 "commodityPropertyId": 40,

 "commodityPropertyCode": "ZT",

 "commodityPropertyName": "状态",

 "commodityPropertyDesc": "网关状态",

 "dataType": 4,

 "dataBody": "{\"0\":\"离线\",\"1\":\"在线\"}",

 "authorizationType": 1,

 "pointType": 1,

 "measType": "711",

 "almpriority": "1",

 "normalState": "",

 "ctrlBody": "[{\"ctrlValue\":\"\",\"descValue\":\"\"}]",

 "createTime": "2024-06-12 15:31:20",

 "createBy": 1,

 "state": 0

 }

 ],

 "commodityServiceList": [

 {

 "commodityServiceId": 9,

 "commodityId": 27,

 "serviceName": "服务1",

 "serviceIdentifier": "s1",

 "invokingType": 0,

 "requestJsonBody": "[{\"paramName\":\"入参1\",\"paramLogo\":\"inparam1\",\"paramDatatype\":0,\"minValue\":\"1\",\"maxValue\":\"50\",\"enum\":[{\"enumVal\":\"\",\"enumDesc\":\"\"}],\"bool0\":\"\",\"bool1\":\"\",\"len\":null,\"step\":null,\"unit\":0},{\"paramName\":\"入参2\",\"paramLogo\":\"inparam2\",\"paramDatatype\":0,\"minValue\":\"1\",\"maxValue\":\"100\",\"enum\":[{\"enumVal\":\"\",\"enumDesc\":\"\"}],\"bool0\":\"\",\"bool1\":\"\",\"len\":null,\"step\":null,\"unit\":0}]",

 "responseJsonBody": "[{\"paramName\":\"出参1\",\"paramLogo\":\"outparam1\",\"paramDatatype\":0,\"minValue\":\"1\",\"maxValue\":\"100\",\"enum\":[{\"enumVal\":\"\",\"enumDesc\":\"\"}],\"bool0\":\"\",\"bool1\":\"\",\"len\":null,\"step\":\"1\",\"unit\":\"count\"}]",

 "serviceDesc": "服务1描述",

 "createTime": "2024-06-24 15:37:21",

 "createBy": 1,

 "state": 0

 }

 ],

 "commodityEventList": [

 {

 "commodityEventId": 8,

 "commodityId": 27,

 "eventName": "事件1",

 "eventIdentifier": "e1",

 "eventType": 0,

 "responseJsonBody": "[{\"paramName\":\"参数1\",\"paramLogo\":\"cs1\",\"paramDatatype\":0,\"minValue\":\"1\",\"maxValue\":\"100\",\"enum\":[{\"enumVal\":\"\",\"enumDesc\":\"\"}],\"bool0\":\"\",\"bool1\":\"\",\"len\":null,\"step\":\"1\",\"unit\":\"count\"},{\"paramName\":\"参数2\",\"paramLogo\":\"cs2\",\"paramDatatype\":0,\"minValue\":\"1\",\"maxValue\":\"200\",\"enum\":[{\"enumVal\":\"\",\"enumDesc\":\"\"}],\"bool0\":\"\",\"bool1\":\"\",\"len\":null,\"step\":null,\"unit\":0}]",

 "eventDesc": "事件1描述",

 "createTime": "2024-06-26 14:21:25",

 "createBy": 1,

 "state": 0

 }

 ]

 },

 "time": "2024-07-31 16:53:49",

 "success": true

}

## 2.5 产品属性列表

### **2.5.1 接口描述**

可以调用此接口查看物联网平台的某个产品下的物模型，从数据库中查询到相应的内容并返回；因此产品代码、产品属性代码是必填项，当数据库中产品代码和属性代码存在，便会返回对应物模型的基本信息，若不存在则会返回错误信息。

### **2.5.2 服务接入**

| **接口地址** | **/iot/rest/api/thingmodel/queryProductPropertys** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 300次/秒 |

### **2.5.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| commodityCode | 产品代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 |  |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

| **参数名称** | **参数说明** | **类型** |
| --- | --- | --- |
| almpriority | 报警优先级 | string |
| authorizationType | 读写类型 | integer(int32) |
| commodityPropertyCode | 产品属性代码 | string |
| commodityPropertyDesc | 产品属性描述 | string |
| commodityPropertyId | 产品属性编号 | integer(int64) |
| commodityPropertyName | 产品属性名称 | string |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| ctrlBody | 控制状态相关josn体 | string |
| dataBody | 数据类型相关json体 | string |
| dataType | 数据类型 | integer(int32) |
| measType | 量测类型 | string |
| normalState | 正常状态 | string |
| pointType | 点类型 | integer(int32) |
| state | 状态 | integer(int32) |

### 2.5.4 对接示例

#### **请求示例**

{

 "commodityCode": "BAS\_ZM\_GATEWAY"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": [

 {

 "commodityPropertyId": 40,

 "commodityPropertyCode": "ZT",

 "commodityPropertyName": "状态",

 "commodityPropertyDesc": "网关状态",

 "dataType": 4,

 "dataBody": "{\"0\":\"离线\",\"1\":\"在线\"}",

 "authorizationType": 1,

 "pointType": 1,

 "measType": "711",

 "almpriority": "1",

 "normalState": "",

 "ctrlBody": "[{\"ctrlValue\":\"\",\"descValue\":\"\"}]",

 "createTime": "2024-06-12 15:31:20",

 "createBy": 1,

 "state": 0

 }

 ],

 "time": "2024-07-31 18:11:40.945",

 "success": true

}

# 3.设备管理

## 概述

创建产品完成后，可以在产品下添加设备。您可通过接口，创建一个或多个设备。前提条件，已完成创建产品并在产品下创建物模型。

成功创建设备后，您可通过接口查看设备信息、进行管理设备。

禁用设备，即禁止设备接入物联网平台；启用设备，即重新启用已被禁用设备，允许设备重新接入物联网平台。

删除设备后，除已产生的云端运行日志外，与该设备关联的其他数据也会一并删除。

您可通过设置设备属性接口完成设备属性值的修改,然后通过查询设备影子数据接口获取设备每个属性当前的属性值。

## 3.1 创建设备

### **3.1.1 接口描述**

可以调用此接口在物联网平台的产品下创建设备，将设备信息添加到数据库中；在接口调用的过程中，产品代码、设备代码、设备键、设备名称是非空值，其他参数可为空，设备创建的成功需要保证产品代码存在，当产品不存在则提示需要创建产品。

### **3.1.2 服务接入**

| **接口地址** | **/iot/rest/api/device/createDevice** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **3.1.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| authorityArea | 权限区域 | 可选 | string | 2 电力3 环控－车站4 环控－区间9 消防11 售检票12 信号13 闭路电视16 屏蔽门17 时钟20 环控复示21 防淹门 |
| bayareaNo | 间隔区域号 | 可选 | string | 13 站厅41 防火分区142 防火分区243 防火分区344 防火分区445 防火分区546 防火分区647 防火分区748 防火分区849 防火分区950 防火分区1051 OCC防火分区152 OCC防火分区253 OCC防火分区354 OCC防火分区455 OCC防火分区558 OCC防火分区8101 电力间隔1 |
| commodityCode | 产品代码 | 必填 | string |  |
| facilityCode | 设备代码 | 必填 | string |  |
| facilityDesc | 设备描述 | 可选 | string |  |
| facilityKey | 设备键 | 必填 | string |  |
| facilityName | 设备名称 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| Code | 响应编码 | integer(int32) |
| Data | 响应实体 | object |
| Msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 3.1.4 对接示例

#### **请求示例**

{

 "authorityArea": "1",

 "bayareaNo": "13",

 "commodityCode": "api\_ceshi",

 "facilityCode": "CGQ",

 "facilityDesc": "传感器",

 "facilityKey": "4q1ohk78mbz3",

 "facilityName": "传感器"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:13:26",

 "success": true

}

## 3.2 修改设备

### **3.2.1 接口描述**

可以调用此接口在物联网平台的修改设备，将设备信息改动更新到数据库中；接口的调用请求需要明确指定设备代码，设置设备的名称，设备描述的内容是可选项，可以是空的，修改设备成功的前提是设备代码存在，如果对应设备不存在则返回错误内容。

### **3.2.2 服务接入**

| **接口地址** | **/iot/rest/api/device/updateDevice** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **3.2.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| facilityCode | 设备代码 | 必填 | string |  |
| facilityDesc | 设备描述 | 可选 | string |  |
| facilityName | 设备名称 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 3.2.4 对接示例

#### **请求示例**

{

 "facilityCode": "CGQ",

 "facilityDesc": "传感器",

 "facilityName": "传感器"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:14:39",

 "success": true

}

## 3.3 禁用设备

### **3.3.1 接口描述**

可以调用此接口在物联网平台禁用设备，禁止对应设备接入到物联网平台；接口调用的过程中，需要填写设备的代码，根据设备的代码禁用设备，可以根据示例填写一个设备的代码禁用单个设备，也可以在数组中添加多个设备代码的json，实现批量禁用设备的功能；设备代码存在则可以禁用成功，否则返回错误信息。

### **3.3.2 服务接入**

| **接口地址** | **/iot/rest/api/device/disableDevice** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 50次/秒 |

### **3.3.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| facilityCode | 设备代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 3.3.4 对接示例

#### **请求示例**

[

 {

 "facilityCode": "CGQ"

 }

]

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:16:11",

 "success": true

}

## 3.4 启用设备

### **3.4.1 接口描述**

可以调用此接口在物联网平台启用设备，允许对应设备接入物联网平台；接口调用的过程中，需要填写设备的代码，根据设备的代码启用设备，可以根据示例填写一个设备的代码启用单个设备，也可以在数组中添加多个设备代码的json，实现批量启用设备的功能；设备代码存在则可以启用成功，否则返回错误信息。

### **3.4.2 服务接入**

| **接口地址** | **/iot/rest/api/device/enableDevice** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 50次/秒 |

### **3.4.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| facilityCode | 设备代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 3.4.4 对接示例

#### **请求示例**

[

 {

 "facilityCode": "CGQ"

 }

]

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:16:55",

 "success": true

}

## 3.5 删除设备

### **3.5.1 接口描述**

可以调用此接口删除物联网平台下的设备，将设备信息从数据库中移除；接口调用需要传设备的代码，设备代码是必填项，不能为空，当设备代码存在可以删除成功并返回成功内容，否则返回失败信息。

### **3.5.2 服务接入**

| **接口地址** | **/iot/rest/api/device/deleteDevice** |
| --- | --- |
| 请求方式 | HTTP / DELETE |
| 调用频率限制 | 100次/秒 |

### **3.5.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| facilityCode | 设备代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 3.5.4 对接示例

#### **请求示例**

{

 "facilityCode": "CGQ"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:18:06",

 "success": true

}

## 3.6 查询设备列表

**3.6.1 接口描述**

可以调用此接口查询物联网平台产品下的设备列表，在数据库中根据产品代码找到对应的设备列表，同时可以根据设备关键字和标签关键字将查到的设备进行筛选，这两个字段是选填内容，可以为空；调用接口的时候必须指定产品的代码，当产品代码存在，根据设备关键字和标签关键字筛选后返回对应的设备信息否则返回错误。

**3.6.2 服务接入**

| **接口地址** | **/iot/rest/api/device/queryDeviceList** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 300次/秒 |

### **3.6.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| commodityCode | 产品代码 | 必填 | string |  |
| keyWord | 设备关键字 | 可选 | string |  |
| tagWord | 标签关键字 | 可选 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | array |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| appSysDesc | 系统描述 | string |
| appSysId | 系统Id | integer(int32) |
| authorityArea | 权限区域 | string |
| bayareaNo | 间隔区域号 | string |
| childSysDesc | 子系统描述 | string |
| childSysId | 子系统Id | integer(int32) |
| clientId | 客户端 | string |
| commodityCode | 产品代码 | string |
| commodityDesc | 产品描述 | string |
| commodityId | 产品编号 | integer(int64) |
| commodityKey | 产品Key | string |
| commodityName | 产品名称 | string |
| commoditySecret | 产品密钥 | string |
| nodeType | 产品接入方式 | integer(int32) |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| facilityCode | 设备代码 | string |
| facilityDesc | 设备描述 | string |
| facilityId | 设备编号 | integer(int64) |
| facilityKey | 设备Key | string |
| facilityName | 设备名称 | string |
| facilitySecret | 设备密钥 | string |
| lastOnlineTime | 最后一次上线时间 | string(date-time) |
| onlineState | 在线状态 | integer(int32) |
| commodityType | 产品类型 | integer(int32) |
| state | 状态 | integer(int32) |

### 3.6.4 对接示例

#### **请求示例**

{

 "commodityCode": "api\_ceshi",

 "keyWord": "",

 "tagWord": ""

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": [

 {

 "appSysId": 3,

 "appSysDesc": "BAS",

 "childSysId": 6,

 "childSysDesc": "给排水系统",

 "commodityId": 22,

 "nodeType": 1,

 "commodityCode": "api\_ceshi",

 "commodityName": "api\_commodityName",

 "commodityDesc": "api测试",

 "commodityKey": "api\_commodityKey",

 "commoditySecret": "4s7fbcb0keoy",

 "commodityType": 6,

 "facilityId": 24,

 "facilityCode": "CGQ",

 "facilityName": "传感器",

 "facilityDesc": "传感器",

 "facilityKey": "4q1ohk78mbz3",

 "facilitySecret": "4s7iwnmm2eea",

 "authorityArea": "1",

 "bayareaNo": "13",

 "createTime": "2024-04-02 10:19:02",

 "createBy": 1,

 "state": 0

 }

 ],

 "time": "2024-04-02 10:20:49",

 "success": true

}

## 3.7 设置设备属性

**3.7.1 接口描述**

可以调用此接口在物联网平台中设置设备的属性值，接受实时更改设备的属性值并存储到数据库；产品Key、产品属性代码、设备Key以及属性值都是必填项，需要根据产品Key和产品属性代码以及设备Key找到对应的设备，再对设备的属性值进行设置；设置成功的前提是产品下对应的设备必须存在，并且此设备有对应的属性，否则返回错误。

**3.7.2 服务接入**

| **接口地址** | **/iot/rest/api/device/setDeviceProperty** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 300次/秒 |

### **3.7.3 数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| commodityKey | 产品Key | 必填 | string |  |
| commodityPropertyCode | 产品属性代码 | 必填 | string |  |
| facilityKey | 设备Key | 必填 | string |  |
| value | 值 | 必填 | object |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 3.7.4 对接示例

#### **请求示例**

{

 "commodityKey": "api\_commodityKey",

 "commodityPropertyCode": "KZMBZT",

 "facilityKey": "4q1ohk78mbz3",

 "value": "1"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:22:31",

 "success": true

}

## 3.8 获取设备属性

**3.8.1 接口描述**

可以调用此接口查询物联网平台设备的全部属性，在数据库中查询设备的所有属性信息并返回；此接口的调用需要填写设备的代码，根据设备代码查询设备属性内容，当数据库存在该设备则返回设备的属性列表，不存在则返回错误信息。

**3.8.2 服务接入**

| **接口地址** | **/iot/rest/api/device/queryProperties** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **3.8.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| facilityCode | 设备代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | array |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| almpriority | 报警优先级 | string |
| authorizationType | 读写类型 | integer(int32) |
| commodityPropertyCode | 产品属性代码 | string |
| commodityPropertyDesc | 产品属性描述 | string |
| commodityPropertyId | 产品属性编号 | integer(int64) |
| commodityPropertyName | 产品属性名称 | string |
| createBy | 创建人 | integer(int64) |
| createTime | 创建时间 | string(date-time) |
| ctrlBody | 控制状态相关josn体 | string |
| currentValue | 设备属性当前值 | string |
| dataBody | 数据类型相关json体 | string |
| dataType | 数据类型 | integer(int32) |
| facilityPropertyCode | 设备属性代码 | string |
| facilityPropertyId | 设备属性编号 | integer(int64) |
| lastUpdateTime | 最后一次更新时间 | string |
| measType | 量测类型 | string |
| normalState | 正常状态 | string |
| pointType | 点类型 | integer(int32) |
| state | 状态 | integer(int32) |

### 3.8.4 对接示例

#### **请求示例**

{

 "facilityCode": "CGQ"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": [

 {

 "commodityPropertyId": 29,

 "commodityPropertyCode": "KZMBZT",

 "commodityPropertyName": "控制面板状态",

 "commodityPropertyDesc": "控制面板状态",

 "dataType": 4,

 "dataBody": "{\"0\":\"关\",\"1\":\"开\"}",

 "authorizationType": 1,

 "pointType": 1,

 "measType": "400",

 "almpriority": "3",

 "normalState": "",

 "ctrlBody": "[{\"ctrlValue\":\"\",\"descValue\":\"\"}]",

 "facilityPropertyId": 49,

 "facilityPropertyCode": "CGQ\_KZMBZT",

 "createTime": "2024-04-02 10:19:02",

 "createBy": 1,

 "state": 0,

 "currentValue": 1,

 "lastUpdateTime": 2024-04-02 10:22:31

 }

 ],

 "time": "2024-04-02 10:24:29",

 "success": true

}

## 3.9 查询设备影子数据

**3.9.1 接口描述**

可以调用此接口查询物联网平台设备属性的当前值与最后更新时间即影子数据，通过设备的代码在数据库中查询属性值和最后更新时间并返回；此接口的调用需要填写设备的代码，当数据库存在该设备则返回设备的影子数据，否则返回错误信息。

**3.9.2 服务接入**

| **接口地址** | **/iot/rest/api/device/queryDeviceShadow** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **3.9.3 数据模型**

#### **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| facilityCode | 设备代码 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | array |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| dataParams | 属性信息 | ThirdShadowResponse |

**ThirdShadowResponse**参数说明

| **参数名称** | **参数说明** | **类型** | **schema** |
| --- | --- | --- | --- |
| currentValue | 设备属性当前值 | string |  |
| lastUpdateTime | 最后一次更新时间 | string |  |

### 3.9.4 对接示例

#### **请求示例**

{

 "facilityCode": "CGQ"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": {

 "dataParams": {

 "KZQZT": {

 "currentValue": "",

 "lastUpdateTime": "2024-04-02 15:45:03"

 }

 }

 },

 "time": "2024-04-03 17:08:28",

 "success": true

}

## 4.0 批量查询设备影子数据

**4.0.1 接口描述**

可以调用此接口批量查询物联网平台设备属性的当前值与最后更新时间即影子数据

**4.0.2 服务接入**

| **接口地址** | **/iot/rest/api/device/batchQueryDeviceShadow** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 100次/秒 |

### **4.0.3 数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| appSysId | 系统Id | false | integer(int32) |  |
| commodityCode | 产品代码 | false | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | array |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

| **参数名称** | **参数说明** | **类型** | **实例说明** |
| --- | --- | --- | --- |
| commodityCode | 产品代码 | string |  |
| commodityName | 产品名称 | string |  |
| facilityCode | 设备代码 | string |  |
| facilityName | 设备名称 | string |  |
| dataParams | 属性信息 | ThirdShadowResponse | ThirdShadowResponse |

**ThirdShadowResponse**参数说明

| **参数名称** | **参数说明** | **类型** | **schema** |
| --- | --- | --- | --- |
| currentValue | 设备属性当前值 | string |  |
| lastUpdateTime | 最后一次更新时间 | string |  |

### 4.0.4 对接示例

#### **请求示例**

{

 "appSysId": 3,

 "commodityCode": ""

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": [

 {

 "commodityCode": "znzmwg",

 "commodityName": "智能照明网关",

 "facilityCode": "ZJS\_ZNZM\_WG",

 "facilityName": "SB\_智能照明网关",

 "dataParams": {

 "KZQZT": {

 "currentValue": "",

 "lastUpdateTime": "2024-04-02 15:45:03"

 },

 "TXZT": {

 "currentValue": "",

 "lastUpdateTime": "2024-04-02 15:45:03"

 },

 "KZMBZT": {

 "currentValue": "",

 "lastUpdateTime": "2024-04-02 15:45:03"

 }

 }

 }

 ],

 "time": "2024-04-02 17:30:00",

 "success": true

}

## 4.1 设备服务调用

**4.1.1 接口描述**

允许用户通过API方式，向指定的设备发送服务调用请求，并获取设备的响应。

**4.1.2 服务接入**

| **接口地址** | **/iot/rest/api/device/deviceServiceInvoke** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 300次/秒 |

### **4.1.3 数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| commodityKey | 产品Key | 必填 | string |  |
| facilityKey | 设备Key | 必填 | string |  |
| params | 请求参数 | 必填 | object |  |
| serviceIdentifier | 产品服务标识 | 必填 | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | object |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

### 4.1.4 对接示例

#### **请求示例**

{

 "commodityKey": "4k6j0z97qbc7",

 "facilityKey": "4m6ivezw1akj",

 "params": {

 "value": {

 "bbbb": "1",

 "cccc": "2"

 }

 },

 "serviceIdentifier": "query"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": null,

 "time": "2024-04-02 10:22:31",

 "success": true

}

# 4.备注

接口请求响应码列表如下：

|  |  |
| --- | --- |
| **响应码** | **状态描述** |
| 200 | 服务调用正常 |
| 400 | 服务调用失败 |
| 500 | 参数非法 |
| 3001 | 产品名称已存在 |
| 3002 | 产品代码已存在 |
| 3003 | 产品键已存在 |
| 3004 | 产品已被使用 |
| 3005 | 最后一条产品属性无法删除 |
| 3006 | 产品属性名称已存在 |
| 3007 | 产品属性代码已存在 |
| 4001 | 产品属性未创建 |
| 4002 | 设备代码已存在 |
| 4003 | 设备键已存在 |

# 5:公共参数

公共请求参数是调用每个API时都需要使用的请求参数。

|  |  |
| --- | --- |
| **字段名** | **描述** |
| Authorization | 授权token令牌,可通过[第三方登录](#_5.1_第三方登录)接口获得,并加上Bearer前缀,以Header形式传入 |

## 5.1 第三方登录

### 5.1.1 接口描述

该接口用于用户登录IOT平台时进行身份验证，并获取公共参数（授权令牌Token）。该Token是后续调用其他API时必须携带的公共参数，用于验证用户身份和权限。Token的有效期默认30分钟，Token失效后用户可以使用刷新Token API或者重新进行第三方登录获取新的有效Token。

### 5.1.2 服务接入

| **接口地址** | **/iot/rest/api/login** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 50次/秒 |

### 5.1.3 **数据模型**

#### **请求参数**

| **参数名称** | **参数说明** | **是否必须** | **数据类型** | **实例说明** |
| --- | --- | --- | --- | --- |
| passWord | 密码 | false | string |  |
| userName | 用户名 | false | string |  |

#### **响应参数**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | array |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

| **参数名称** | **参数说明** | **类型** | **实例说明** |
| --- | --- | --- | --- |
| token | token | string |  |

### 5.1.4 对接示例

#### **请求示例**

{

 "passWord": "123456",

 "userName": "admin"

}

#### **响应示例**

{

 "code": 200,

 "msg": "成功",

 "data": {

 "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsImlhdCI6MTcxMjYzMjkzOCwidXNlciI6IntcInVzZXJOYW1lXCI6XCJhZG1pblwiLFwicGFzc1dvcmRcIjpcIjEyMzQ1NlwiLFwicm9sZVwiOlwiYWRtaW5cIixcImp1cmlzZGljdGlvblwiOlwiaW90X3ZpZXcuaW90X2VkaXRcIn0iLCJleHAiOjE3MTI2MzQ3Mzh9.mdCEkYlyOxJhfoRdkPLki6GRA3lsUTyX-Kne2ikDUNGw4jYlFsqq3q3oQ8CNTbxKUBit03xPHdSsBc4sGx-eKg"

 },

 "time": "2024-04-09 11:22:18",

 "success": true

}

## 5.2刷新Token

### 5.2.1 接口描述

该接口用于用户重新获取有效期30分钟的授权令牌Token。

### 5.2.2 服务接入

| **接口地址** | **/iot/rest/api/refreshToken** |
| --- | --- |
| 请求方式 | HTTP / POST |
| 调用频率限制 | 50次/秒 |

### 5.2.3 数据模型

#### 请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | **参数说明** | **in** | **是否必须** | **数据类型** | **实例说明** |
| Authorization | token令牌 | header | false | string |  |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数说明** | **类型** |
| code | 响应编码 | integer(int32) |
| data | 响应实体 | array |
| msg | 响应消息 | string |
| success | 成功标志 | boolean |
| time | 时间 | string(date-time) |

data响应实体说明:

| **参数名称** | **参数说明** | **类型** | **实例说明** |
| --- | --- | --- | --- |
| token | token | string |  |
| jurisdiction | 权限 | string |  |

### 5.2.4 对接示例

#### 响应示例

{

 "code": 200,

 "msg": "成功",

 "data": {

 "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0aGlyZCIsImlhdCI6MTczOTE0OTk5MSwidXNlciI6IntcInVzZXJOYW1lXCI6XCJ0aGlyZFwiLFwicGFzc1dvcmRcIjpcIjEyMzQ1NlwiLFwicm9sZVwiOlwidGhpcmR1c2VyXCIsXCJqdXJpc2RpY3Rpb25cIjpcImlvdF90aGlyZFwifSIsImV4cCI6MTczOTE1MTc5MX0.c00vzD3SPSqUJuggqi57BGLzQd4M4YKZXN2Te-wrFGxL6uDbjwqaHaIT34GnSvtyvMLBz47DrYIROhcsTMQbvA",

 "jurisdiction": "iot\_third"

 },

 "time": "2025-02-10 09:13:11.390",

 "success": true

}