

中国电信物联网开放平台 API 参考 1.3.1

（V1）

中国电信股份有限公司物联网分公司

二〇一八年五月

编制单位：

编制单位	中国电信股份有限公司物联网分公司平台运营部

修订记录：

版本号	日期	描述
01	2017/11/06	中国电信物联网开放平台 API 参考 1.3.1

目 录

1 Platform API 参考(北向)..... 1
1.1 前言..... 1
1.2 接口列表..... 1
1.2.1 应用安全接入..... 2
1.2.1.1 Auth(鉴权)..... 2
1.2.1.2 Refresh Token(刷新 token)..... 3
1.2.1.3 注销..... 5
1.2.2 设备管理..... 6
1.2.2.1 注册直连设备..... 6
1.2.2.2 发现非直连设备..... 8
1.2.2.3 查询设备激活状态..... 11
1.2.2.4 删除直连设备..... 13
1.2.2.5 删除非直连设备..... 14
1.2.2.6 修改设备信息..... 17
1.2.2.7 刷新设备密钥..... 19
1.2.2.8 设置加密..... 21
1.2.3 数据采集..... 23
1.2.3.1 按条件批量查询设备信息列表..... 23
1.2.3.2 查询单个设备信息..... 27
1.2.3.3 Application 订阅平台数据..... 29
1.2.3.4 查询设备历史数据..... 30
1.2.3.5 查询设备能力..... 34
1.2.4 信令传送..... 38
1.2.4.1 创建设备命令 V4..... 38
1.2.4.2 查询设备命令 V4..... 41
1.2.4.3 修改设备命令 V4..... 45
1.2.4.4 创建设备命令撤销任务 V4..... 47
1.2.4.5 查询设备命令撤销任务 V4..... 50
1.2.5 设备服务调用..... 54
1.2.5.1 设备服务调用..... 54
1.2.6 批量处理..... 58

1.2.6.1 创建批量任务 58
1.2.6.2 查询单个任务信息 61
1.2.6.3 查询任务详情信息 63
1.2.7 规则 66
1.2.7.1 创建规则 66
1.2.7.2 更新规则 79
1.2.7.3 修改规则状态 82
1.2.7.4 删除规则 84
1.2.7.5 查找规则 85
1.2.7.6 批量修改规则状态 87
1.2.8 消息推送 90
1.2.8.1 注册直连设备的通知 90
1.2.8.2 发现非直连设备的通知 91
1.2.8.3 设备信息变化的通知 93
1.2.8.4 设备数据变化的通知 94
1.2.8.5 删除非直连设备的通知 95
1.2.8.6 消息确认通知 96
1.2.8.7 设备响应命令通知 98
1.2.8.8 设备事件通知 100
1.2.8.9 设备服务上报通知 101
1.2.8.10 规则事件上报通知 103
1.2.8.11 绑定设备通知 105
1.2.8.12 批量设备数据变化上报通知 106
1.3 常用数据结构体定义 108
1.3.1 DeviceInfo 结构体说明 108
1.3.2 DeviceService 结构体说明 109
1.3.3 QueryDeviceDTOCloudNA 结构体说明 110
1.3.4 ServiceCommand 结构体说明 110
1.3.5 ServiceProperty 结构体说明 111
1.3.6 DeviceCommandResp 结构体说明 112
1.3.7 CommandDTOV4 结构体说明 113
1.3.8 GetDeviceRspDTO 结构体说明 113
1.3.9 CommandNA2CloudHeader 结构体说明 114
1.3.10 ApplicationSetEncrtptDTO 结构体说明 114
1.3.11 PutCarInfoData 结构体说明 115

1 Platform API 参考(北向)

1.1 前言

1.2 接口列表

1.3 常用数据结构体定义

1.1 前言

本文档系统化描述 IoT Platform（下文简称平台）对外开放的能力全集、集成原理和集成参考样例等信息，帮助集成开发者快速而准确的掌握集成方法从而高效实现特定的业务需求。本文档主要包含如下几个部分：

一接口列表

主要描述能力开放的接口集合，详细介绍每个接口的功能、输入参数、输出参数、和消息样例等信息。

二常用数据结构定义

是对多个接口都会用到的数据结构做一个详细介绍，方便开发者集中查看。

1.2 接口列表

Application 侧接口除了鉴权接口 Auth，其他接口调用都需要在 request header 中携带参数 app_key 和 Authorization:Bearer {accessToken}。Authorization 中{accessToken}的值即为调用 Auth 接口获取到的 accessToken。



应用开发语言若是 JAVA 请使用 JDK1.8。

北向提供的 API 是平台与 Application 之间的 https 接口， 请使用安全传输协议 TLS 1.1/1.2。

1.2.1 应用安全接入

Application 携带在 IoT Platform（下文简称平台）产生的 appId 和 secret 过来，调用鉴权接口，获取鉴权 token。请参考本文档 2.1.1 鉴权章节进行开发。

1.2.1.1 Auth(鉴权)

接口功能

实现第三方系统在访问开放 API 之前的认证。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/sec/v1.1.0/login
```

注意事项

鉴权接口是调用其他 API 的前提，北向接口除了鉴权接口（Auth），其他接口调用都需要在 request header 中携带参数 app_key 和 Authorization:Bearer {accessToken}。app_key 为参数中的 appId，Authorization 中{accessToken}的值即为调用 Auth 接口获取到的 accessToken。

如果多次获取令牌，则之前的令牌失效，最后一次获取的令牌才有效。请勿并发获取令牌。

参数说明

参数	必选/可选	类型	描述
appId	必选	String	用户名，填写应用程序 ID
secret	必选	String	登录用户口令，填写应用程序密码

返回结果

字段	类型	描述
scope	String	范围，默认值 default
tokenType	String	鉴权 token 类型，默认值 bearer
expiresIn	Integer	平台生成并返回 accessToken 的有效时间，单位秒
accessToken	String	Oauth 2.0 鉴权参数
refreshToken	String	Oauth 2.0 鉴权参数，用来刷新 accessToken。（1 个月的有效期）

消息事例

Method：

```
POST
```

request：(非 JSON 格式)

```
https://server:port/iocm/app/sec/v1.1.0/login
Content-Type:application/x-www-form-urlencoded
Body:
appId={appId}&secret={secret}
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
 "scope":"default",
 "tokenType":"bearer",
 "expiresIn":"*******",
 "accessToken":"*******",
 "refreshToken":"*******"
}
```

异常返回码

Http Status Code	**error_code**	**error_desc**	**说明**
401	100208	AppId or secret is not right.	appId 或 secret 错误

1.2.1.2 Refresh Token(刷新 token)

接口功能

accessToken 快过期时，第三方系统通过调用此接口，重新获取可用 token。accessToken 有效时间参照 2.1.1 鉴权接口返回字段 expiresIn 的值。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/sec/v1.1.0/refreshToken
```

注意事项

Body 参数说明

参数	必选/可选	类型	描述
appId	必选	String	用户名，填写应用程序 ID

参数	必选/可选	类型	描述
secret	必选	String	登录用户口令，填写应用程序密码
refreshToken	必选	String	刷新令牌，用于获取一个新的accessToken

返回结果

字段	类型	描述
scope	String	范围
tokenType	String	鉴权 token 类型，默认值 bearer
expiresIn	String	平台生成并返回 accessToken 的有效时间，单位秒
accessToken	String	Oauth 2.0 鉴权参数
refreshToken	String	刷新令牌，用于获取一个新的 accessToken

消息事例

Method：

```
POST
```

request：

```
https://server:port/iocm/app/sec/v1.1.0/refreshToken
Content-Type:application/json
Body:
{
 "appId"="******",
 "secret"="******",
 "refreshToken":"******"
}
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
 "accessToken":"*******",
 "tokenType":"*******",
 "expiresIn":"*******",
 "scope":”*******”,
 "refreshToken":"*******"
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
400	100022	The input is invalid.	输入参数无效
401	100006	Refresh access token failed.	refreshToken 无效
401	100208	AppId or secret is not right.	appId 或 secret 错误

1.2.1.3 注销

接口功能

NA 注销鉴权信息。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/sec/v1.1.0/logout
```

注意事项

无

参数说明

参数	必选/可选	类型	描述
accessToken	必选	String(256)	调用鉴权接口获取到的 Oauth 2.0 鉴权参数

返回结果

Status Code: 204 No Content

消息事例

Method:

```
POST
```

Request:

```
https://server:port/iocm/app/sec/v1.1.0/logout
Header:
Content-Type:application/json
Body:
{
   "accessToken": "*******"
}
```

Response:

```
Status Code: 204 No Content
Content-Type: application/json
```

异常返回码

Http Status Code	error_code	error_desc	说明
400	100022	The input is invalid.	输入参数无效

1.2.2 设备管理

设备管理提供了 Application（下文简称 App）申请设备的增，删，改，查以及修改设备基本信息的接口。

App 向 IoT Platform 申请新的设备, IoT Platform 分配对应的设备验证码，待设备携带验证码请求接入平台后，分配其 deviceId 和 secret，允许其使用。平台提供了增，删，改，查接口，实现对新增设备接入灵活操作。对设备的管理请参考本文档进行开发。

1.2.2.1 注册直连设备

接口功能

应用程序添加设备，获取设备的验证码，并在设备访问南向接口时携带验证码，获取设备唯一标识和密码。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/reg/v1.2.0/devices?appId={appId}
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

备注：app_key 为参数中的 appId，Authorization 中{accessToken}的值即为调用 Auth 接口获取到的 accessToken。

参数说明

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识
verifyCode	可选	String	客户端给出 verifyCode 则返回的就是这个 verifyCode，即使用客户端给出的 verifyCode。如果不指定，系统自动生成。

字段	必选/可选	类型	描述
nodeId	必选	String	设备唯一标识，如：MAC 或 SIM 卡号或设备 esn 号等。与设备对接时，必须与设备上报的 nodeId 一致。 备注：nodeId 和 verifyCode 需要填写相同的值，如果南向连接 SoftRadio 的模拟环境，该值可自行定义，格式为 TESTS_XXXXXX。如果是现网环境，该值通常为设备的 IMEI 号。
endUserId	可选	String	终端用户 Id，如手机号码，email 地址
psk	可选	String	psk 码，用于生成设备鉴权参数；如不传入，系统自动生成
timeout	可选	Integer	单位秒，不填使用默认值(180s)，填写 0 则永不过期，非 0 表示在指定时间内设备进行绑定，超过时间验证码无效

返回结果

字段	类型	描述
deviceId	String	设备唯一标识，1-64 个字节
verifyCode	String	申请的临时验证码，设备可以通过验证码获取 id 和密码
timeout	Integer	验证码有效时间，单位秒，设备需要在有效时间内接入平台
psk	String	psk 码，用于生成设备鉴权参数

消息事例

Method：

```
POST
```

request：

```
https: //server:port/iocm/app/reg/v1.2.0/devices?appId={appId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json
Body:
{
    "verifyCode":"AE10-12424-12414",
    "nodeId":"AE10-12424-12414",
    "timeout":300
}
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "deviceId":"*******",
    "verifyCode":"*******",
    "timeout":300,
    "psk":"******"
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
401	100025	AppId for auth not exist.	获取不到 appId 鉴权信息
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未被授权
400	100007	Bad request message.	参数不合法
200	100412	The amount of device has reached the limit.	当前应用下设备数量达到上限
400	100003	Invalid verify code.	验证码无效
200	100416	The device has already been binded.	设备已经绑定
500	100001	Internal server error.	服务内部处理错误
200	103026	The license is not exist.	License 不存在
200	103028	The license pool resources.	License 无资源
200	103027	The license sales is not exist.	License 的销售项不存在

1.2.2.2 发现非直连设备

接口功能

Application 发送 DISCOVERY 命令给网关下的设备。一般用于通过网关添加传感器，若没有网关，该接口请勿使用。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/{serviceId}/sendCommand?appId={appId}
```

注意事项

```
http 携带头域信息：
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

URL 参数说明：

字段	必选/可选	类型	描述
deviceId	必选	String	网关设备唯一标识，1-64 个字节
serviceId	必选	String	取值"Discovery"

http 消息体说明：

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识
message	必选	CommandDTONA2Cloud	CommandDTONA2Cloud 消息体说明参见下表

CommandDTONA2Cloud 消息体说明：

字段	必选/可选	类型	描述
header	必选	CommandNA2CloudHeader	见附录 1.3.9 CommandNA2CloudHeader 结构体说明字段说明
body	可选	ObjectNode	取值长度无要求，消息的消息体

返回结果：

字段	类型	描述
status	String(128)	命令状态：sent --- 已发送、delivered --- 已送达 executed --- 已执行

字段	类型	描述
timestamp	String(128)	发送命令的时间戳，1-32 字节 时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z
requestId	String(128)	1-32 个字节，平台分配的序列号，标识一个命令 需要 requestId 关联对应命令执行结果

消息事例

Method：

```
POST
```

request：

```
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/Discove
ry/sendCommand?appId={appId}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;

Body:
{
  "header": {
    "mode": "ACK",
    "from": "************",
    "to": null,
    "toType": null,
    "method": "DISCOVERY",
    "requestId": "**************************************",
    "callbackURL": "https://server:port/na/iocm/message/confirm"
  },
  "body": {
     "from": "************",
     "sessionID": "**********",
     "sdp": "**********"
  }
}
```

response：

```
Status Code: 202 Accepted
Content-Type: application/json
Body:
{
   "requestId":"**************************************",
   "status":"sent",
   "timestamp":"1446202014815"
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未授权
200	100418	The deviceData is not existed.	设备数据不存在
500	100023	The data in dataBase is abnomal.	数据库中数据异常
400	102203	CommandName is invalid.	命令名称无效
200	100431	The serviceType is not exist.	服务类型不存在
200	100428	The device is not online.	设备不在线
200	100432	The device command is muted.	设备已被 muted

1.2.2.3 查询设备激活状态

接口功能

给定网关设备的设备唯一标识查询网关设备激活状态。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/reg/v1.1.0/devices/{deviceId}?appId={appId}
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
deviceId	必选	String	设备唯一标识，1-64 个字节

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识

返回结果

字段	必选/可选	类型	描述
deviceId	必选	String	设备唯一标识，1-64 个字节
activated	必选	boolean	是否通过验证码获取密码的状态标识 true:已激活;false:未激活
name	必选	String	设备名称

消息事例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/reg/v1.1.0/devices/{deviceId}?appId={appId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
"deviceId":*******",
"activated":*******,
"name":"***"
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用没有权限

Http Status Code	error_code	error_desc	说明
200	100403	The device is not existed.	设备不存在

1.2.2.4 删除直连设备

接口功能

用来删除直连设备.

调用方法

```
DELETE
```

接口路径

```
https://server:port/iocm/app/dm/v1.1.0/devices/{deviceId}?appId={appId}&cascade={cascade}
```

注意事项

```
携带头域信息：
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

本接口只能用于删除直连设备；

参数说明

字段	必选/可选	类型	描述
deviceId	必选	String	设备唯一标识，1-64 个字节
appId	可选	String	应用唯一标识
cascade	可选	Boolean	仅在设备是网关，且连接了传感器时有效。取值 null 或者 true 删除网关及其下属设备；取值 false 时删除网关，并将下属传感器属性变为网关属性

返回结果

Status Code: 204 No Content.

消息事例

Method:

```
DELETE
```

Request:

```
https://server:port/iocm/app/dm/v1.1.0/devices/{deviceId}?appId={appId}&cascade={cascade}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

Response:

```
Status Code: 204 No Content
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未授权
200	100403	The device is not existed.	设备不存在
200	100418	The deviceData is not existed.	设备数据不存在
500	100023	The data in dataBase is abnomal.	数据中数据异常
400	102203	CommandName is invalid.	命令名称无效
200	100431	The serviceType is not exist.	服务类型不存在
200	100428	The device is not online.	设备不在线
200	100432	The device command is muted.	设备命令已经 muted

1.2.2.5 删除非直连设备

接口功能

Application 发送 REMOVE 命令给网关下的设备。用于通过网关删除传感器，若没有网关，该接口请勿使用。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/{serviceId}/sendCommand?appId={appId}
```

注意事项

```
http 携带头域信息：
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

http 消息体说明：

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识
message	必选	CommandDTONA2Cloud	NA 发送给设备的命令内容

CommandDTONA2Cloud 消息体说明：

字段	必选/可选	类型	描述
header	必选	CommandNA2CloudHeader	见 1.3.9 CommandNA2CloudHeader 结构体说明字段说明
body	可选	ObjectNode	取值长度无要求，消息的消息体

返回结果：

字段	类型	描述
status	String	命令状态：1-32 字节 sent --- 已发送 delivered --- 已送达 executed --- 已执行
timestamp	String	发送命令的时间戳，1-32 字节 时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z
requestId	String	1-32 个字节，平台分配的序列号，标识一个命令 需要 requestId 关联对应命令执行结果

消息事例

Method：

```
POST
```

request：

```
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/Remover
/sendCommand?appId={appId}

Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;

Body:
{
  "header": {
    "mode": "ACK",
    "from": "************",
    "to": null,
    "toType": null,
    "method": "REMOVE",
    "requestId": "****************************************",
    "callbackURL": "https://server:port/callbackUri"
  },
  "body": {
    "timeout": 90,
    "deviceId": "********-0fda-4a56-bed6-4d9019891493",
    "mode": "ON"
  }
}
```

response：

```
Status Code: 202 Accepted
Content-Type: application/json
Body:
{
   "requestId":"****************************************",
   "status":"sent",
   "timestamp":"1446202014815"
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未授权
200	100418	The deviceData is not existed.	设备数据不存在

Http Status Code	error_code	error_desc	说明
500	100023	The data in dataBase is abnomal.	数据库中数据异常
400	102203	CommandName is invalid.	命令名称无效
200	100431	The serviceType is not exist.	服务类型不存在
200	100428	The device is not online.	设备不在线
200	100432	The device command is muted.	设备已被 muted

1.2.2.6 修改设备信息

接口功能

修改设备信息的接口，目前用于设备改名，预留修改设备信息的能力。

调用方法

```
PUT
```

接口路径

```
https://server:port/iocm/app/dm/v1.2.0/devices/{deviceId}?appId={appId}
```

注意事项

```
携带头域信息：
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
deviceId	必选	String	设备唯一标识，1-64 个字节
appId	可选	String	应用唯一标识
request	可选	UpdateDeviceInfoReqDTO	自定义结构体（见下表）

UpdateDeviceInfoReqDTO 结构体

字段	必选/可选	类型	描述

字段	必选/可选	类型	描述
name	可选	String	设备名称
endUser	可选	String	终端用户，若设备为网关，则 endUser 可选；若设备是通过网关接入的，则 endUser 必须为 null。
mute	可选	Enum	表示设备是否处于冻结状态，即是否上报数据（处于冻结状态，则不上报数据），取值有“TRUE”,“FALSE”
manufacturerId	可选	String	厂商 ID
manufacturerName	可选	String	厂商名
location	可选	String	设备的位置
deviceType	可选	String	设备类型: 大驼峰命名方式 MultiSensor ContactSensor Camera Gateway
model	可选	String	型号 z-wave: ProductType + ProductId 16 进制： XXXX-XXXX 补 0 对齐 如：001A-0A12 其他协议再定
protocolType	可选	String	设备协议类型
deviceConfig	可选	DeviceConfigDTO	设备配置信息，自定义结构体（见下表）
region	可选	String	设备所在地区
organization	可选	String	设备所属组织
timezone	可选	String	设备所在时区

DeviceConfigDTO

字段	必选/可选	类型	描述
dataConfig	可选	DataConfigDTO	数据配置信息，自定义结构体（见下表）

DataConfigDTO

字段	必选/可选	类型	描述
dataAgingTime	可选	Integer[0,90]	数据老化时长配置，取值：090，单位：天

返回结果

```
Status Code: 204 No Content
```

消息事例

Method：

```
PUT
```

request：

```
https://server:port/iocm/app/dm/v1.1.0/devices/{deviceId}?appId={appId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
Body:
{
"name":"****",
"endUser":"****"
}
```

response

```
Status Code: 204 No Content
```

1.2.2.7 刷新设备密钥

接口功能

支持刷新注册到本应用的设备和授予权限的其它应用的设备

密钥刷新后，deviceId 不变，nodeId 值发生变化，设备需要重新绑定。

调用方法

```
PUT
```

接口路径

```
https://server:port/iocm/app/reg/v1.1.0/devices/{deviceId}?appId={appId}
```

注意事项

```
携带头域信息
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
appId	可选	String	query	授权应用 Id
deviceId	必选	String(256)	path	设备 Id
verifyCode	可选	String(256)	body	验证码：null 平台生成验证码，并返回；非 null 使用，且返回该验证码。（大小写不敏感）
nodeId	可选	String(256)	body	null:nodeId 不变；非 null：更新；网关的唯一 ID，平台用于判重
timeout	可选	Integer	body	验证码超时时间，单位秒。null：默认值(180s)；0：永不过期；非 0：指定时间

返回结果

字段	类型	描述
verifyCode	String(256）	申请的临时验证码，网关可以通过验证码获取 id 和密码
timeout	Integer	验证码有效时间，单位秒，网关需要在有效时间内接入平台

消息事例

Method:

```
PUT
```

Request:

```
https://server:port/iocm/app/reg/v1.1.0/devices/{deviceId}?appId={appId}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;

Body:
{
   "verifyCode": "AE10-12424-12414",
   "nodeId": "rewqrewqrewqre",
   "timeout": 300
}
```

Response:

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
   "verifyCode": "AE10-12424-12414",
   "timeout": 300
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
401	100025	AppId for auth not exist.	应用 ID 不存在
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未被授权
400	100007	Bad request message.	参数不合法
200	100403	The device is not existed.	设备不存在
200	100610	Device is not active.	设备未激活
200	100611	Device is online.	设备在线
200	100418	The deviceData is not existed.	设备信息不存在
200	100426	The nodeId is duplicated.	nodeId 重复
400	100003	Invalid verify code.	验证码无效

1.2.2.8 设置加密

接口功能

NA 调用此接口将车机的加密信息发送给平台

调用方法

```
PUT
```

接口路径

```
https://server:port/iocm/app/dm/v1.1.0/devices/{deviceId}/services/{serviceType}
```

注意事项

```
携带头域信息
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
deviceId	必选	String	path	设备 id
serviceType	必选	String	path	设备服务类型，可选值 carEncrypt，carInfo，vehicleProperty
data	可选	JSONObject	body	服务数据： serviceType 取值 carEncrypt 时，data 结构体参考 1.3.10 ApplicationSetEncrtptDTO 结构体说明； serviceType 取值 carInfo、vehicleProperty 时，data 结构体参考 1.3.11 PutCarInfoData 结构体说明

返回结果

设置加密成功：Status Code: 200 OK

服务类型找不到：Status Code: 400 Not Found

消息事例

Method:

```
PUT
```

Request:

```
https://server:port/iocm/app/dm/v1.1.0/devices/{deviceId}/services/{serviceType}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;

Body:
{
  "data":{
        *****
  }
}
```

Response:

```
Status Code: 200 OK
Content-Type: application/json
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100403	The device is not existed.	设备不存在
400	100019	Illegal request.	非法请求，request 为空
400	103008	The encrtpt iv is not exist.	加密 iv 值不存在
400	103009	The encrtpt key is not exist.	加密 key 不存在
400	103007	The encrtpt flag is not exist.	加密 flag 不存在
200	103004	Add new car info, simNumber or carModel cannot be null.	新的 carInfo, simNumber 或者 carModel 不能为 null
200	103001	The simNumber is already exist.	simNumber 已经存在
200	103002	The plateNumber is already exist.	plateNumber 已经存在
200	103010	The vin is already exist.	vin 已经存在
404	200001	The resource is not found.	资源不存在

1.2.3 数据采集

平台支持对网关事件的订阅，同时还支持查看传感器上报的历史数据，能按时、天、月等维度查看传感器上报的历史数据。数据采集实现了对网关数据收集。

1.2.3.1 按条件批量查询设备信息列表

接口功能

按条件批量查询设备信息列表，如查询指定网关下的所有设备信息列表。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/dm/v1.3.0/devices?appId={appId}&gatewayId={gatewayId}&nodeType={nodeType}&deviceType={deviceType}&protocolType={protocolType}&pageNo={pageNo}&pageSize={pageSize}&startTime={startTime}&endTime={endTime}&status={status}&sort={sort}
```

注意事项

```
携带头域信息
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识
gatewayId	条件可选	String	设备唯一标识（含网关的设备唯一标识）。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。 gatewayId 和 pageNo 不能同时为 null，二者必选其一。
nodeType	可选	String	指定节点的类：ENDPOINT: 传感器. GATEWAY: 网关.
deviceType	可选	String	设备类型
protocolType	可选	String	协议类型（v1.3.0url 携带）
pageNo	条件可选	Integer	分页查询参数， pageNo=null 时查询内容不分页；取值大于等于 0 的整数时分页查询,等于 0 时查询第一页。 gatewayId 和 pageNo 不能同时为 null，二者必选其一。
pageSize	可选	Integer	分页查询参数，取值大于等于 1 的整数，缺省：1
status	可选	String	设备状态,取值:ONLINE 在线;OFFLINE 不在线;ABNORMAL 异常状态
startTime	可选	String	查询注册设备信息时间在 startTime 之后的记录。时间格式：yyyyMMdd’T’HHmmss’Z’ 如： 20151212T121212Z
endTime	可选	String	查询注册设备信息时间在 endTime 之前的记录。时间格式：yyyyMMdd’T’HHmmss’Z’ 如： 20151212T121212Z
sort	可选	String	指定返回记录的排序，可取值 ASC 按注册设备

字段	必选/可选	类型	描述
			的时间升序排列； DESC 按注册设备的时间降序排列 缺省：DESC

注：gatewayId 和 pageNo 不能同时为空。

结果说明

字段	类型	描述
totalCount	Long	查询的记录数量
pageNo	Long	返回的页码序号(大于等于 0 的整数,0 代表第一页)
pageSize	Long	页码大小,当不分页查询时,取值等于 totalCount
devices	List<GetDeviceRspDTO>	设备分页列表信息，自定义结构，1.3.8 GetDeviceRspDTO 结构体说明

消息事例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/dm/v1.3.0/devices?appId={appId}&gatewayId={gatewayId}&nodeType={nodeType}&deviceType={deviceType}&protocolType={protocolType}&pageNo={pageNo}&pageSize={pageSize}&startTime={startTime}&endTime={endTime}&status={status}&sort={sort}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
   "totalCount":1,
   "pageNo":0,
   "pageSize":10,
   "devices":[
      {
         "deviceId":"************************",
         "gatewayId":"************************",
```

```
            "nodeType":"GATEWAY",
            "createTime":"20170426T022604Z",
            "lastModifiedTime":"20170426T022604Z",
            "deviceInfo":{
               "nodeId":null,
               "name":null,
               "description":null,
               "manufacturerId":null,
               "manufacturerName":null,
               "mac":null,
               "location":null,
               "deviceType":null,
               "model":null,
               "swVersion":null,
               "fwVersion":null,
               "hwVersion":null,
               "protocolType":null,
               "bridgeId":null,
               "status":"OFFLINE",
               "statusDetail":"NOT_ACTIVE",
               "mute":null,
               "supportedSecurity":null,
               "isSecurity":null,
               "signalStrength":null,
               "sigVersion":null,
               "serialNumber":null,
               "batteryLevel":null
            },
            "services":null,
            "connectionInfo":{
               "protocolType":null
            },
            "location":null,
            "devGroupIds":[]
         }
      ]
}
```


异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100216	The application input is invalid.	应用输入无效
200	100218	The gatewayId and pageNo can't be both null.	网关 id 和 pageNo 不能同时为 null
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未被授权

1.2.3.2 查询单个设备信息

接口功能

查询指定条件的设备信息。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/dm/v1.3.0/devices/{deviceId}?appId={appId}
```

注意事项

```
携带头域信息
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识
deviceId	必选	String	设备唯一标识（含网关的设备唯一标识），1-64 个字节

返回结果：1.3.8 GetDeviceRspDTO 结构体说明**结构**

消息事例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/dm/v1.3.0/devices/{deviceId}?appId={appId}
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
          "deviceId":"************************",
          "gatewayId":"************************",
```

```
        "nodeType":"GATEWAY",
        "createTime":"20170426T022604Z",
        "lastModifiedTime":"20170426T022604Z",
        "deviceInfo":{
            "nodeId":null,
            "name":null,
            "description":null,
            "manufacturerId":null,
            "manufacturerName":null,
            "mac":null,
            "location":null,
            "deviceType":null,
            "model":null,
            "swVersion":null,
            "fwVersion":null,
            "hwVersion":null,
            "protocolType":null,
            "bridgeId":null,
            "status":"OFFLINE",
            "statusDetail":"NOT_ACTIVE",
            "mute":null,
            "supportedSecurity":null,
            "isSecurity":null,
            "signalStrength":null,
            "sigVersion":null,
            "serialNumber":null,
            "batteryLevel":null
        },
        "services":null,
        "connectionInfo":{
            "protocolType":null
        },
        "location":null,
        "devGroupIds":[]
}
```


异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
200	100203	The application is not existed.	应用不存在
200	100403	The device is not existed.	设备不存在
200	100418	The deviceData is not existed.	设备数据不存在
200	100217	The application hasn't been authorized.	应用未被授权

1.2.3.3 Application 订阅平台数据

接口功能

Application 订阅设备变更通知，当设备发生变更时平台会推送给 Application。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/sub/v1.2.0/subscribe
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，如： 1、deviceAdded（添加新设备） 2、deviceInfoChanged（设备信息变化） 3、deviceDataChanged（设备数据变化） 4、deviceDeleted（删除设备） 5、deviceEvent（设备事件） 6、messageConfirm（消息确认） 7、commandRsp（响应命令） 8、serviceInfoChanged（设备信息） 9、ruleEvent（规则事件） 10、bindDevice（设备绑定激活） 11、deviceDatasChanged（设备数据批量变化）
callbackurl	必选	String	回调的 url 地址。必须使用 HTTPS 信道回调地址，同时回调地址中必须指定回调地址的端口。(说明：HTTP 信道只可用于调测)

返回结果

```
Status Code: 201 Created
```

消息事例

Method：

```
POST
```

request：

```
https://server:port/iocm/app/sub/v1.2.0/subscribe
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
Body:
{
 "notifyType":"deviceInfoChanged”,
 "callbackurl":"https://server:port/callbackUri"
}
```

response：

```
Status Code: 201 Created
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100222	The request callbackurl is illegal.	回调地址非法
200	100203	The application is not existed.	鉴权应用不存在
500	101017	Get newCallbackUrl from oss failed.	从 oss 获取新的回调地址失败
500	101016	Get iotws address failed.	获取 iotws 地址失败

1.2.3.4 查询设备历史数据

接口功能

Application 查询设备历史数据。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/data/v1.1.0/deviceDataHistory?deviceId={deviceId}&gatewayId={gatewayId}&serviceId ={serviceId}&pageNo={pageNo}&pageSize ={pageSize}&startTime={startTime}&endTime={endTime}&property={property}&appId={appId}
```

注意事项

```
携带头域信息
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
deviceId	必选	String	设备唯一标识，1-64 个字节
gatewayId	必选	String	查询参数，网关的设备唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
serviceId	可选	String	服务 Id
property	可选	String	服务属性数据
appId	可选	String	应用唯一标识
pageNo	可选	Integer	分页查询参数， pageNo=null 时查询内容不分页；取值大于等于 0 的整数时分页查询，等于 0 时查询第一页
pageSize	可选	Integer	分页查询参数，取值大于等于 1 的整数，缺省值为 1 设备历史数据最多保存 10 万条，这个数值有可能被运营商修改
startTime	可选	String	查询参数，根据时间段查询的起始时间; 时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z 设备历史数据最多保存 90 天，这个数值有可能被运营商修改
endTime	可选	String	查询参数，根据时间段查询的结束时间; 时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z

返回结果

字段	必选/可选	类型	描述
totalCount	可选	Long	查询的记录数量
pageNo	必选	Long	返回的页码序号(大于等于 0 的整数,0 代表第一页)
pageSize	必选	Long	页码大小,当不分页查询时,取值等于 totalCount

字段	必选/可选	类型	描述
list	必选	List<DeviceDataHistoryDTO>	历史记录列表

DeviceDataHistoryDTO

字段	必选/可选	类型	描述
serviceId	必选	String	服务 ID
deviceId	必选	String	设备唯一标识，1-64 个字节
gatewayId	必选	String	网关的设备唯一标识，1-64 个字节。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
appId	必选	String	应用程序 ID
data	必选	JsonObject	设备上报的数据，数据是键值对，取值请参见 Profile 文件中服务能力表里面的 propertyties。
timestamp	必选	String	上报数据的时间戳 时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z

消息事例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/data/v1.1.0/deviceDataHistory?deviceId={deviceId}&
gatewayId={gatewayId}&serviceId ={serviceId}&pageNo={pageNo}&pageSize
={pageSize}&startTime={startTime}&endTime={endTime}&property={property}&appId={appI
d}
Content-Type:application/json
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type:application/json
Body:
```

```
{
   "totalCount":"****",
   "pageNo":"*****",
   "pageSize":"*****",
   "deviceDataHistoryDTOs":["*****"]
}
```

注：实例说明调用查询设备历史数据接口的返回结果，如下：

```
{
   "totalCount": 2,
   "pageNo": 0,
   "pageSize": 10,
   "deviceDataHistoryDTOs": [
      {
         "deviceId": "0d34bb9d-73e1-4780-a12f-9243345a4567",
         "gatewayId": "f6ee5aff-53f1-4459-be2f-4fc05ccd7004",
         "appId": "c8855f40-d7a3-4d51-8b46-9f7747839ee2",
         "serviceId": "Battery",
         "data": {
            "batteryLevel": 100
         },
         "timestamp": "20160708T025828Z"
      },
      {
         "deviceId": "0d34bb9d-73e1-4780-a12f-9243345a4567",
         "gatewayId": "f6ee5aff-53f1-4459-be2f-4fc05ccd7004",
         "appId": "c8855f40-d7a3-4d51-8b46-9f7747839ee2",
         "serviceId": "Temperature",
         "data": {
            "temperature": "33.5"
         },
         "timestamp": "20160708T025827Z"
      }
   ]
}
```

上述实例涉及的 Profile 文件服务能力表如下：

1. **servicetype-capability.json（Battery）**

```
{
    "services": [
       {
          "serviceType": "Battery",
          "description": "Battery",
          "commands": null,
          "properties": [
             {
                "propertyName": "batteryLevel",
                "dataType": "int",
                "required": true,
                "min": 0,
                "max": 100,
                "step": 1,
                "maxLength": null,
                "method": "RE",
```

```
                "unit": null,
                "enumList": null
            },
        ]
    }
  ]
}
```

2. **servicetype-capability.json（Temperature）**

```
{
   "services": [
      {
         "serviceType": "Temperature",
         "description": "Temperature",
         "commands": null,
         "properties": [
            {
               "propertyName": "temperature",
               "dataType": "decimal",
               "required": true,
               "min": 0,
               "max": 100,
               "step": null,
               "maxLength": null,
               "method": "R",
               "unit": "C",
               "enumList": null
            }
         ]
      }
   ]
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
200	100203	The application is not existed.	应用不存在
400	100419	The deviceId and gatewayId can't be both null.	deviceId 和 gatewayId 不能同时为空
200	100217	The application hasn't been authorized.	应用未被授权

1.2.3.5 查询设备能力

接口功能

查询设备能力

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/data/v1.1.0/deviceCapabilities?appId={appId}&gatewayId={gatewayId}&deviceId={deviceId}
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	描述
appId	可选	String	应用唯一标识，1-64 个字节
deviceId	必选	String	设备唯一标识，1-64 个字节
gatewayId	可选	String	设备的网关 id。当设备是直连设备时，gatewayId 务必与 deviceId 保持一致。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。

返回结果

字段	类型	描述
deviceCapabilities	List<DeviceCapabilityDTO>	自定义结构体，见下表

DeviceCapabilityDTO

字段	类型	描述
deviceId	String	设备唯一标识，1-64 个字节
serviceCapabilities	List<ServiceCapabilityDTO >	设备的服务能力列表

ServiceCapabilityDTO

字段	类型	描述
serviceId	String	服务标识
serviceType	String	服务类型
description	String	描述
commands	List<ServiceCommand>	支持的命令名称列表，ServiceCommand 结构体
properties	List<ServiceProperty>	支持的属性名称列表，ServiceProperty 结构体

ServiceCommand 结构体说明

字段	类型	描述
commandName	String(256)	命令名称
paras	ServiceCommandPara	属性列表
responses	ServiceCommandResponse	响应列表

ServiceCommandPara 结构体说明

字段	类型	描述
paraName	String(256)	参数名称
dataType	String(256)	数据类型
required	boolean	是否必选
min	String	最小
max	String	最大
step	Double	步长
maxLength	Int	最大长度
unit	String	单位（符号）
enumList	List<String>	枚举类型列表

ServiceCommandResponse 结构体说明

字段	类型	描述

responseName	String(256)	响应名称
paras	List<ServiceCommandPara>	属性列表

ServiceProperty 结构体说明

字段	类型	描述
propertyName	String(256)	属性名称
dataType	String(256)	数据类型
required	boolean	是否必选
min	String	最小
max	String	最大
step	double	步长
maxLength	int	最大长度
method	String(256)	访问方法：RWE 可读 R，可写 W，可观察 E
unit	String	单位（符号）
enumList	List<String>	枚举类型列表

消息事例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/data/v1.1.0/deviceCapabilities?appId={appId}&gatewayId
={gatewayId}&deviceId={deviceId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type: application/json
Body: {
   "deviceId":*****
   "serviceCapabilities" : [{
     "serviceId" : "",
     "serviceType" : "",
```

```
    "option" : "",
    "description" : null,
    "commands" : null,
    "properties" : [{
       "propertyName" : "",
       "desc" : "",
       "dataType" : "",
       "enumList" : ["", ""],
       "min" : 0,
       "max" : 0,
       "step" : 0.0,
       "maxLength" : 0.0,
       "unit" : null,
       "method" : "",
       "required" : true
     }],
    "events" : null
  }
]
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未被授权

1.2.4 信令传送

应用场景：NA 给设备发送消息

开发指导：平台分别提供了下发至设备的接口，下发消息的具体格式需要与设备约定。

1.2.4.1 创建设备命令 V4

接口功能

应用创建设备命令。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?appId={appId}
```

注意事项

携带头域信息：

```
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明：

字段	必选/可选	类型	位置	描述
body	必选	PostDeviceCommandReq	body	设备命令的内容,详见请求结构体说明
appId	可选	String(1-64)	query	应用 ID，该应用 ID 要与当前访问应用有绑定关系方可访问

PostDeviceCommandReq 请求结构体说明

字段	必选/可选	类型	描述
deviceId	必选	String(64)	下发命令的设备 ID
command	必选	CommandDTOV4	下发命令的信息
callbackUrl	可选	String(1024)	命令状态变化通知地址，必须使用 HTTPS 信道回调地址 (说明：HTTP 信道只可用于调测，不可用于商用环境)； 当命令状态变化时(执行失败，执行成功，超时，发送，已送达)会通知 NA，平台会以 POST 方式发送 HTTP 消息给应用，请求 Body 为 json 字符串，格式形如：{"deviceId":"deviceId","commandId":"commandId","result":{"status":"SUCCESS","result":{…}}}
expireTime	可选	Integer(>=0)	下发命令有效的超期时间，单位为秒，表示设备命令在创建后 expireTime 秒内有效，超过这个时间范围后命令将不再下发，如果未设置则默认为 48 小时

CommandDTOV4 结构体说明

字段	必选/可选	类型	描述
serviceId	必选	String(1-64)	命令对应的服务ID，不能为空字符串
method	必选	String(1-128)	命令服务下具体的命令名称，服务属性等，不能为空字符串
paras	可选	ObjectNode	命令参数的jsonString，具体格式需要应用和设备约定。

返回结果

投递成功：Status Code: 201 Created

返回参数参见附录 1.3.6 DeviceCommandResp 结构体说明 结构体说明

消息示例

Method：

```
POST
```

request：

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?appId={appId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
Body:
{
      "deviceId": "********",
      "command": {
            "serviceId": "********",
            "method": "********",
            "paras": {
                  "paraName1": "paraValue1",
                  "paraName2": "paraValue2"
       }
   },
      "callbackUrl": "https://server:port/callbackUri"
}
```

response：

```
Status Code: 201 Created
Content-Type:application/json

Body:
{
      "commandId": "********",
      "appId": "********",
      "deviceId": "********",
      "command": {
            "serviceId": "********",
            "method": "********",
            "paras": {
                  "paraName1": "paraValue1",
                  "paraName2": "paraValue2"
       }
   },
   "callbackUrl": "https://server:port/callbackUri",
      "expireTime": null,
      "status": "PENDDING",
      "result": null,
      "creationTime": "20170222T164000Z",
      "executeTime": null,
      "platformIssuedTime": null,
      "deliveredTime": null,
      "issuedTimes": null
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
400	100022	The input is invalid.	输入参数无效
500	100220	Get AppKey from header failed.	获取 appKey 失败
500	100001	Internal server error.	立即下发命令发送超时
404	100203	The application is not existed.	应用不存在
404	100418	The deviceData is not existed.	设备数据不存在
404	100431	The serviceType is not exist.	服务类型不存在

1.2.4.2 查询设备命令 V4

接口功能

应用查询设备命令。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?pageNo={pageNo}&pageSize={pageSize}&deviceId={deviceId}&startTime={startTime}&endTime={endTime}&appId={appId}
```

注意事项

http 携带头域信息：

```
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	位置	描述
pageNo	可选	Integer(>=0)	query	查询的页码，大于等于 0，默认 0
pageSize	可选	Integer(>=1&&<=1000)	query	查询每页的数量，大于等于 1，最大值 1000，默认 1000
deviceId	可选	String(64)	query	指定查询设备命令的设备 ID
startTime	可选	String	query	查询开始时间，查询下发命令时间在 startTime 之后的记录。时间格式：yyyyMMdd'T'HHmmss'Z' 如： 20151212T121212Z
endTime	可选	String	query	查询结束时间，查询下发命令时间在 endTime 之前的记录。时间格式：yyyyMMdd'T'HHmmss'Z' 如： 20151212T121212Z
appId	可选	String	query	应用 ID，该应用 ID 要与当前访问应用有绑定关系方可访问

返回结果

查询成功：Status Code: 200 OK

DeviceCommandQueryResp 结构体说明

字段	必选/可选	类型	描述
pagination	必选	Pagination	页码信息
data	可选	List<DeviceCommandResp>	设备命令列表，1.3.6 DeviceCommandResp 结构体说明

Pagination 结构体说明

字段	必选/可选	类型	描述
pageNo	必选	long	页码
pageSize	必选	long	分页大小
totalSize	必选	long	记录总数

返回结果

查询成功：Status Code: 200 OK

消息示例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands?pageNo=0&pageSize=10&deviceId=******&startTime=20151112T101012Z&endTime=20151212T121212Z
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type:application/json

Body:
{
   "pagination": {
         "pageNo": 0,
         "pageSize": 20,
         "totalSize": 100
   },
     "data": [
            {
            "commandId": "********",
            "appId": "********",
            "deviceId": "********",
            "command": {
               "serviceId": "********",
               "method": "********",
               "paras": {
                         "paraName1": "paraValue1",
                         "paraName2": "paraValue2"
             }
          },
```

```
            "callbackUrl": "https://server:port/callbackUri",
            "expireTime": null,
            "status": "PENDDING",
            "result": null,
            "creationTime": "20170222T164000Z",
            "executeTime": null,
            "platformIssuedTime": null,
            "deliveredTime": null,
            "issuedTimes": null
      },
            {
            "commandId": "********",
            "appId": "********",
            "deviceId": "********",
            "command": {
               "serviceId": "********",
               "method": "********",
               "paras": {
                         "paraName1": "paraValue1",
                         "paraName2": "paraValue2"
                }
          },
            "callbackUrl": "https://server:port/callbackUri",
            "expireTime": null,
            "status": "PENDDING",
            "result": null,
            "creationTime": "20170222T164000Z",
            "executeTime": null,
            "platformIssuedTime": null,
            "deliveredTime": null,
            "issuedTimes": null
      }
   ]
}
```


异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	无效的输入
500	100220	Get AppKey from header failed.	获取 appKey 失败
200	100203	The application is not existed.	应用不存在
200	100418	The deviceData is not existed.	设备数据不存在
200	100403	The device is not existed.	设备不存在
200	100434	The device command is not existed.	命令不存在

1.2.4.3 修改设备命令 V4

接口功能

应用修改命令信息，当前仅支持修改设备命令状态为撤销，即为将设备命令撤销。

调用方法

```
PUT
```

接口路径**

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands/{deviceCommandId}?appId={appId}
```

注意事项

http 携带头域信息：

```
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	位置	描述
deviceCommandId	必选	String	path	待修改的设备命令 ID
appId	可选	String	query	应用 ID，该应用 ID 要与当前访问应用有绑定关系方可访问
body	必选	UpdateDeviceCommandReq	body	修改设备命令的内容，详见请求结构体说明

UpdateDeviceCommandReq 结构体说明

字段	必选/可选	类型	描述
status	必选	String	命令执行结果，可选值：CANCELED

返回结果

```
Status Code: 200 OK
```

返回参数：参见附录 1.3.6 DeviceCommandResp 结构体说明 结构体说明

消息示例

Method：

```
PUT
```

request：

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommands/{deviceCommandId}?appId={app
Id}
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
Body:
{
   "status":"CANCELED"
}
```

response：

```
StatusCode: 200 OK
{
      "commandId": "********",
      "appId": "********",
      "deviceId": "********",
      "command": {
         "serviceId": "********",
         "method": "********",
         "paras": {
                   "paraName1": "paraValue1",
                   "paraName2": "paraValue2"
         }
      },
      "callbackUrl": "https://server:port/callbackUri",
      "expireTime": null,
      "status": "PENDDING",
      "result": null,
      "creationTime": "20170222T164000Z",
      "executeTime": null,
      "platformIssuedTime": null,
      "deliveredTime": null,
      "issuedTimes": null
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
400	100022	The input is invalid.	输入参数无效
500	100220	Get AppKey from header failed.	获取 appKey 失败
200	100217	The application hasn't been authorized.	应用没有操作权限

200	100203	The application is not existed.	应用不存在
200	100418	The deviceData is not existed.	设备数据不存在
200	100434	The device command is not existed.	命令不存在
200	100435	The device command already canceled, expired or executed, Cannot cancel.	命令已撤销、过期或执行，无法撤销

1.2.4.4 创建设备命令撤销任务 V4

接口功能

应用创建设备命令撤销任务，该任务用于撤销指定设备 ID 下的所有设备命令。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?appId={appId}
```

注意事项

http 携带头域信息：

```
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	位置	描述
appId	可选	String	query	应用 ID，该应用 ID 要与当前访问应用有绑定关系方可访问
body	必选	CreateDeviceCommandCancelTaskReq	body	设备命令撤销任务参数，详见请求结构体说明

CreateDeviceCommandCancelTaskReq 请求结构体说明

字段	必选/可选	类型	描述

deviceId	必选	String(1-64)	待撤销设备命令的设备 ID，撤销任务将会撤销所有给该设备下发的设备命令

返回结果

```
Status Code: 201 Created
```

DeviceCommandCancelTaskResp 响应结构体说明

字段	必选/可选	类型	描述
taskId	必选	String(1-64)	设备命令撤销任务的任务 ID
appId	必选	String(1-64)	设备命令撤销任务所属的应用 ID
deviceId	必选	String(1-64)	设备命令撤销任务指定撤销命令的设备 ID
status	必选	String	撤销任务的任务状态，WAITTING 表示等待执行中，RUNNING 表示撤销任务正在执行，SUCCESS 表示撤销任务执行成功，FAILED 表示撤销任务执行失败，PART_SUCCESS 表示推销任务部分执行成功。
totalCount	必选	Integer	撤销的设备命令总数
deviceCommands	可选	List<DeviceCommandResp>	设备命令撤销任务的任务 ID，参见附录 1.3.6 DeviceCommandResp 结构体说明 结构体说明

消息示例

Method：

```
POST
```

request：

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?appId={appId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
```

```
Content-Type:application/json;
Body:

{
   "deviceId": "********"
}
```

response：

```
Status Code: 201 Created
{
   "taskId": "********",
   "appId": "********",
   "deviceId": "********",
   "status": "WAITTING",
   "totalCount": 1,
   "deviceCommands": [
      {
            "commandId": "********",
            "appId": "********",
            "deviceId": "********",
            "command": {
               "serviceId": "********",
               "method": "********",
               "paras": {
                         "paraName1": "paraValue1",
                         "paraName2": "paraValue2"
             }
          },
            "callbackUrl": "https://server:port/callbackUri",
            "expireTime": null,
            "status": "PENDDING",
            "result": null,
            "creationTime": "20170222T164000Z",
            "executeTime": null,
            "platformIssuedTime": null,
            "deliveredTime": null,
            "issuedTimes": null
      }
   ]
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	无效的输入
400	100220	Get AppKey from header failed.	获取 appKey 失败
401	100217	The application hasn't been authorized.	应用不存在

404	100203	The application is not existed.	设备数据不存在
404	100418	The deviceData is not existed.	设备不存在

1.2.4.5 查询设备命令撤销任务 V4

接口功能

应用查询设备命令撤销任务。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?pageNo={pageNo}&pageSize={pageSize}&taskId={taskId}&deviceId={deviceId}&status={status}&startTime={startTime}&endTime={endTime}&appId={appId}
```

注意事项

http 携带头域信息：

```
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

字段	必选/可选	类型	位置	描述
pageNo	可选	Integer(>=0)	query	查询的页码，大于等于 0，默认 0
pageSize	可选	Integer(>=1&&<=1000)	query	查询每页的数量，大于等于 1，最大值 1000，默认 1000
taskId	可选	String	query	指定任务 ID 查询设备命令撤销任务
deviceId	可选	String	query	指定设备 ID 查询设备命令撤销任务
status	可选	String	query	查询指定任务状态的设备命令撤销任务

startTime	可选	String	query	查询开始时间，查询创建撤销设备命令任务时间在 startTime 之后的记录。时间格式：yyyyMMdd'T'HHmmss'Z' 如：20151212T121212Z
endTime	可选	String	query	查询结束时间，查询创建撤销设备命令任务时间在 endTime 之前的记录。时间格式：yyyyMMdd'T'HHmmss'Z' 如：20151212T121212Z
appId	可选	String	query	应用 ID，该应用 ID 要与当前访问应用有绑定关系方可访问

返回结果

查询成功：Status Code: 200 OK

字段	必选/可选	类型	描述
pagination	必选	Pagination	页码信息
data	可选	List<DeviceCommandCancelTaskResp>	设备命令列表

Pagination 响应结构体说明

字段	必选/可选	类型	描述
pageNo	必选	long	页码
pageSize	必选	long	分页大小

totalSize	必选	long	记录总数

DeviceCommandCancelTaskResp 响应结构体说明

字段	必选/可选	类型	描述
taskId	必选	String(1-64)	设备命令撤销任务的任务 ID
appId	必选	String(1-64)	设备命令撤销任务所属的应用 ID
deviceId	必选	String(1-64)	设备命令撤销任务指定撤销命令的设备 ID
status	必选	String	撤销任务的任务状态，WAITTING 表示等待执行中，RUNNING 表示撤销任务正在执行，SUCCESS 表示撤销任务执行成功，FAILED 表示撤销任务执行失败，PART_SUCCESS 表示推销任务部分执行成功。
totalCount	必选	Integer	撤销的设备命令总数
deviceCommands	可选	List<DeviceCommandResp>	设备命令撤销任务的任务 ID，参见附录 1.3.6 DeviceCommandResp 结构体说明 结构体说明

消息示例

Method：

```
GET
```

request：

```
https://server:port/iocm/app/cmd/v1.4.0/deviceCommandCancelTasks?pageNo={pageNo}&pageSize={pageSize}&taskId={taskId}&deviceId={deviceId}&status={status}&startTime={startTime}&endTime={endTime}&appId={appId}
```

```
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

response：

```
Status Code: 200 OK
Content-Type:application/json

Body:
{
   "pagination": {
         "pageNo": 0,
         "pageSize": 20,
         "totalSize": 100
   },
      "data": [
            {
          "taskId": "********",
          "appId": "********",
          "deviceId": "********",
          "status": "WAITTING",
          "totalCount": 1,
          "deviceCommands": [
              {
                    "commandId": "********",
                    "appId": "********",
                    "deviceId": "********",
                    "command": {
                       "serviceId": "********",
                       "method": "********",
                       "paras": {
                                 "paraName1": "paraValue1",
                                 "paraName2": "paraValue2"
                     }
                  },
                    "callbackUrl": "https://server:port/callbackUri",
                    "expireTime": null,
                    "status": "PENDDING",
                    "result": null,
                    "creationTime": "20170222T164000Z",
                    "executeTime": null,
                    "platformIssuedTime": null,
                    "deliveredTime": null,
                    "issuedTimes": null
              }
          ]
       },
            {
          "taskId": "********",
          "appId": "********",
          "deviceId": "********",
          "status": "WAITTING",
          "totalCount": 1,
          "deviceCommands": [
```

```
            {
                  "commandId": "********",
                  "appId": "********",
                  "deviceId": "********",
                  "command": {
                     "serviceId": "********",
                     "method": "********",
                     "paras": {
                               "paraName1": "paraValue1",
                               "paraName2": "paraValue2"
                      }
                   },
                     "callbackUrl": "https://server:port/callbackUri",
                     "expireTime": null,
                     "status": "PENDDING",
                     "result": null,
                     "creationTime": "20170222T164000Z",
                     "executeTime": null,
                     "platformIssuedTime": null,
                     "deliveredTime": null,
                     "issuedTimes": null
            }
         ]
      }
   ]
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	无效的输入
400	100220	Get AppKey from header failed.	获取 appKey 失败
401	100217	The application hasn't been authorized.	应用不存在
404	100203	The application is not existed.	设备数据不存在
404	100418	The deviceData is not existed.	设备不存在

1.2.5 设备服务调用

1.2.5.1 设备服务调用

接口功能

Application 给设备发送命令消息，实现对传感器的实时控制。平台分别提供了下发至设备或者具体某传感器的控制命令接口，下发消息的具体格式需要 Application 与设备自定义，平台在接口中进行封装，透传。

Application 发送命令给网关（或者网关下的设备）指定的服务下发命令/事件。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/{serviceId}/sendCommand?appId={appId}
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
appId	可选	String	query	授权应用 Id
deviceId	必选	String	path	设备唯一标识，1-64 个字节
serviceId	必选	String	path	服务 id，1-64 个字节 唯一标识一个服务，设备的服务类型不冲突，服务类型就是服务 id。
header	必选	CommandNA2CloudHeader	Body	见下表 CommandNA2CloudHeader 字段说明
body	必选	JsonObject	Body	命令的具体内容，取值长度无要求。 JsonObject 里面是一个个健值对。每个健都是 profile 中命令的参数名（paraName）。

CommandNA2CloudHeader **字段说明：**

字段	必选/可选	类型	描述
requestId	可选	String(128)	如果填写该 requestId，就是用填写的 requestId 标识一个命令，这里的 requestId 不能重复，由发起端 APP 按照下面的规则保证其唯一性：Generate

字段	必选/可选	类型	描述
			seqNum：APP 生成序列号 seqNum，在 POST 请求 BODY 中携带，序列号规则采用：UUID_XXXX，UUID 会根据 JAVA 自带算法生成，每次 APP 启动后生成一个新的值，后续使用时不变，直到下次 APP 重新启动再生成一个新的 UUID 值，XXXX 的取值范围：0001-9999，达到 9999 后重新从 0001 开始，如此循环
mode	必选	Enum	是否要确认消息，不需要确认消息：NOACK ，需要确认消息：ACK，其它值无效
from	可选	String(128)	表示消息发布者的地址： App 发起的请求：/users/{userId} NA 发起的请求：/{serviceName} 平台发起的请求：/cloud/{serviceName}
toType	可选	Enum	CLOUD、GATEWAY
to	可选	String(128)	接收者地址
method	必选	String(1-32)	命令名称，profile 命令中定义的命令名（commandName），如："DISCOVERY"
callback URL	可选	String(1024)	回调的 url 地址。必须使用 HTTPS 信道回调地址，同时回调地址中必须指定回调地址的端口。(说明：HTTP 信道只可用于调测)

返回结果

命令已成功投递: Status Code: 200 OK

命令已接收处理：StatusCode: 202 Accepted

字段	类型	描述
status	String(128)	命令状态： sent --- 已发送 deliveried ---toType 为 CLOUD 的请求才有该返回值 failed--- toType 为 CLOUD 的请求才有该返回值
timestamp	String(128)	发送命令的时间戳 时间格式：yyyyMMdd’T’HHmmss’Z’ 如： 20151212T121212Z
requestId	String(128)	toType 为 GATEWAY，如果请求有 requestId 则同请求的 requestId 一致，如果请求无 requestId 则平台分配一个序列号 toType 为 CLOUD，则为 null;

消息示例

Method:

```
POST
```

Request:

```
https://server:port/iocm/app/signaltrans/v1.1.0/devices/{deviceId}/services/{servic
eId}/sendCommand?appId={appId}
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;

Body:
{
   "header": {
      "mode": "ACK",
      "from": "/users/23212121",
      "method": "INVITE-INIT",
      "callbackURL": "https://186.25.1.49:8043/na/iocm/message/confirm"
   },
   "body": {
      "from": "************",
      "sessionID": "**********",
      "sdp": "**********"
   }
}
```

Response:

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
   "requestId": "************",
   "status": "sent",
   "timestamp": "**********"
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未授权
200	100418	The deviceData is not existed.	设备数据不存在
500	100023	The data in dataBase is	数据库中数据异常

Http Status Code	error_code	error_desc	说明
		abnomal.	
400	102203	CommandName is invalid.	命令名称无效
200	100431	The serviceType is not exist.	服务类型不存在
200	100428	The device is not online.	设备不在线
200	100432	The device command is muted.	设备已被 muted

1.2.6 批量处理

1.2.6.1 创建批量任务

接口功能

NA 创建批量任务。支持给本应用和授予权限的其它应用创建批量任务。

调用方法

```
POST
```

接口路径

```
https://server:port/iocm/app/batchtask/v1.1.0/tasks
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
appId	必选	String(64)	body	应用唯一标识
timeout	必选	Integer	body	任务超时时间
taskName	必选	String	body	任务名称
taskType	必选	String	body	任务类型 DeviceCmd
para	必选	Objec	body	根据 taskType 任务类型的对应不同类型参

参数	必选/可选	类型	位置	描述
m		tNode		数，详见下表 param 参数结构说明

taskType 参数取值 DeviceCmd（批量下发命令任务），param 参数：

字段	必选/可选	类型	描述
type	必选	String	批量命令类型： DeviceList / DeviceType / DeviceArea / GroupList / Broadcast
deviceList	可选	List<String>	deviceId 列表，当命令类型为 DeviceList 时，需要填写
deviceType	可选	String	设备类型，当命令类型为 DeviceType 时需要填写。
manufacturerId	可选	String	厂商 ID，当命令类型为 DeviceType 时可填写
model	可选	String	设备型号，当命令类型为 DeviceType 时可填写。
deviceLocation	可选	String	设备位置，当命令类型为 DeviceArea 时，需要填写
groupList	可选	List<String>	组名字列表，当命令类型为 GroupList 时需要填写。
command	必选	CommandDTOV1	命令，查看 CommandDTOV1 结构定义

CommandDTOV1 结构体说明

字段	必选/可选	类型	描述
serviceId	必选	String(64)	服务 id
method	必选	String(32)	命令名称，服务属性等
paras	可选	ObjectNode	命令参数的 jsonString

返回结果

Status Code: 200 OK

字段	类型	描述
taskID	String	任务 ID

消息示例

Method: POST

Request:

```
https://server:port/iocm/app/batchtask/v1.1.0/tasks
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
Body:
{
 "appId" : "******",
 "timeout" : 1000,
 "taskName" : "******",
    "taskType" : "DeviceReg",
    "param" : {
       "fileId" : "******"
     }
}
```

Response:

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
    "taskID":"******"
}
```

异常返回码

Http Status Code	**error_code**	**error_desc**	**说明**
401	100002	Invalid access token.	错误的 token 信息
200	105001	The batchTask count has reached the limit.	未完成的任务数大于等于 10 个时，返回任务达到数量限制
200	105002	The batchTask name has exist.	任务名字已存在
400	100007	Bad request message.	错误的请求消息
401	100028	The user has no right.	用户没有操作权限

1.2.6.2 查询单个任务信息

接口功能

NA 查询单个任务信息。支持查询本应用和授予权限的其它应用创建的批量任务信息。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/batchtask/v1.1.0/tasks/{taskId}?appId={appId}
```

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
taskId	必选	String	path	任务 Id
appId	可选	String	query	授权应用 Id

返回结果

Status Code: 200 OK

字段	类型	描述
taskId	String	任务 ID
taskName	String	任务名字
appId	String	任务的归属 appId
operator	String	下发任务的操作员
taskFrom	String	Portal/Northbound
taskType	String	任务类型 DeviceCmd
status	String	任务状态 Pending/Running/Complete/Timeout
startTime	String	任务的创建时间
timeout	Int	任务的超时时间
progress	Int	任务进度，千分比 0-1000，向下取整
totalCnt	Int	任务详情总数

字段	类型	描述
successCnt	Int	成功的任务详情数
failCnt	Int	失败的任务详情数
timeoutCnt	Int	超时的任务详情数
expiredCnt	Int	未执行的失效任务详情数
completeCnt	Int	完成的任务详情数，完成=成功+失败+超时
successRate	Int	任务成功率，千分比 0-1000，向下取整
param	ObjectNode	不同的任务类型的具体参数

param 参数结构说明：

taskType 参数取值 DeviceCmd（批量下发命令任务），param 参数：

字段	必选/可选	类型	描述
commandType	必选	String	批量命令类型： DeviceList / DeviceType / DeviceArea / GroupList / Broadcast
deviceType	可选	String	设备类型
manufacturerId	可选	String	厂商 Id
model	可选	String	设备类型
serviceId	必选	String	设备中的服务标识
method	必选	String	命令名称，如："DISCOVERY"

消息示例

Method: GET

Request:

```
https://server:port/iocm/app/batchtask/v1.1.0/tasks/{taskId}?appId={appId}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

Response:

```
Status Code: 200 OK
Content-Type: application/json
Body:
```

```
{
   "taskId" : "********",
   "taskName" : "********",
   "appId" : "********",
   "operator" : "********",
   "taskFrom" : "********",
   "taskType" : "DeviceCmd",
   "status" : "********",
   "startTime" : "********",
   "timeout" : 1000,
   "progress" : 100,
   "totalCnt" : 100,
   "successCnt" : 70,
   "failCnt" : 10,
   "timeoutCnt" : 10,
   "expiredCnt" : 10,
   "completeCnt" : 100,
   "successRate" : 70,
   "param" : {
     "fileName" : "******"
    }
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未被授权

1.2.6.3 查询任务详情信息

接口功能

NA 查询单个任务的任务详情信息。支持查询本应用和授予权限的其它应用创建的批量任务信息。

调用方法

```
GET
```

接口路径

```
https://server:port/iocm/app/batchtask/v1.1.0/taskDetails?taskId={taskId}&appId={appId}&status={status}&index={index}&nodeId={nodeId}&deviceId={deviceId}&commandId={commandId}&pageNo={pageNo}&pageSize={pageSize}
```

注意事项

```
携带头域信息
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
appId	可选	String	query	授权应用 Id
taskId	必选	String	query	任务 Id
status	可选	String	query	任务详情状态 Pending/WaitResult/Success/Fail/ Timeout
pageNo	可选	Integer	query	分页查询参数，pageNo=null 时查询内容不分页；取值大于等于 0 的整数时分页查询,等于 0 时查询第一页
pageSize	可选	Integer	query	分页查询参数，取值大于等于 1 的整数,缺省：1
index	可选	Integer	query	文件里第几行,查询批量注册设备使用
nodeId	可选	String	query	设备 nodeId，查询批量注册设备使用
deviceId	可选	String	query	设备 Id，查询批量命令使用
commandId	可选	String	query	命令 Id，查询批量命令使用

返回结果

Status Code: 200 OK

字段	类型	描述
pageNo	Integer	页码
pageSize	Integer	每页数据记录数量
totalCount	Integer	总数据记录数量
taskDetails	LIST<QueryTaskDetailDTOCloud2NA>	设备详情信息列表

QueryTaskDetailDTOCloud2NA 结构

字段	类型	描述
status	String	Pending/WaitResult/Success/Fail/Timeout，任务详情状态
output	ObjectNode	批量注册设备或批量下发命令的输出，详见下表
error	String	错误原因，格式为： {"error_code": "", "error_desc": ""}
param	ObjectNode	不同的任务类型的具体参数，详见下表

批量下发命令任务，output 为：

字段	类型	描述
requestId	String(64)	请求 id
commandResult	CommandResult	命令的响应

批量下发命令任务，param 为：

字段	类型	描述
deviceId	String	设备 Id
commandId	String	命令 Id

消息示例

Method: GET

Request:

```
https://server:port/iocm/app/batchtask/v1.1.0/taskDetails?taskId={taskId}&appId={appId}&status={status}&index={index}&nodeId={nodeId}&deviceId={deviceId}&commandId={commandId}&pageNo={pageNo}&pageSize={pageSize}
Header:
"app_key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

Response:

```
Status Code: 200 OK
Content-Type: application/json
Body:
{
   "totalCount": 1,
   "pageNo": 0,
   "pageSize": 1,
   "taskDetails": [
      {
         "status": "Fail",
```

```
            "output": null,
            "error": "{\"error_code\":100007,\"error_desc\":\"Badrequest message.\"}",
            "param": {
                "index": 0,
                "nodeId": "1666"
            }
        }
    ]
}
```

异常返回码

Http Status Code	error_code	error_desc	说明
400	100022	The input is invalid.	输入参数无效
200	100203	The application is not existed.	应用不存在
200	100217	The application hasn't been authorized.	应用未被授权

1.2.7 规则

Application 能够根据具体的业务需要，制定规则（条件）和动作，当网关的状态达到指定规则后，以邮件或者短信的方式对用户进行实时提醒。根据业务需要，编排规则，指定场景触发。

1.2.7.1 创建规则

接口功能

创建规则

请求方法

POST

请求路径

https://server:port/iocm/app/rule/v1.2.0/rules

注意事项

携带头域信息

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id。
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的 accessToken。

消息体参数定义查看 RuleDTO1.2 定义说明。

RuleDTO1.2 结构体说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id。
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的 accessToken。
ruleId	可选	String	body	规则实例的 id，仅在规则更新时有效，其余情况应为空
appKey	可选	String	body	应用实例的 key 值(1-128)
name	必选	String	body	规则名（并且接受 1~128 个字符）
description	可选	String	body	规则描述(1-256)
author	必选	String	body	创建此规则的用户的 id（1-128）（不能为空）
conditions	可选	List < Json >	body	条件列表，自定义结构体（见下表），与 groupExpress 二选一必填
logic	可选	String	body	多条件之间的逻辑关系，支持 and 和 or，默认为 and
timeRange	可选	TimeRange	body	条件场景的时间段。TimeRange 为自定义结构体（见下表 Condition-CycleTimer）
actions	定时场景必选	List < Json >	body	规则的动作。Action 为自定义结构体。（见下面 Action-DeviceCMD、Action -SMS、Action -email、Action-DeviceAlarm）
matchNow	可选	String	body	表示是否立即触发，即是否理解进行条件判断，条件符合的话执行动作，取值有“yes”,“no”，默认为“no”

status	可选	String	body	规则的状态，“active”代表激活状态，“inactive”代表未激活，默认为“active”状态
groupExpress	可选	GroupExpress	body	复杂多条件表达式 。为自定义结构体 GroupExpress，与 conidtions 二选一必填
timezoneID	可选	String	body	国际时区，如 Asia/Shanghai、UTC
triggerSources	可选	List <Json	body	触发源列表，自定义结构体 TriggerSource，只与 GroupExpress 联用构造复杂多条件规则，目前有 DEVICE 类型和 TIMER 类型

Device-TriggerSource 结构体说明

字段	必选/可选	字段类型	描述
type	必选	Enum	设备触发源类型 DEVICE
deviceId	必选	String	设备 Id(1-36)
serviceId	必选	String	服务 id(1-128)

Timer-TriggerSource 结构体说明

字段	必选/可选	字段类型	描述
type	必选	Enum	设备触发源类型 TIMER

GroupExpress 结构体说明：

字段	必选/可选	字段类型	描述
operator	可选	Enum	操作符 and/or，默认为 and
groupElements	必选	List<GroupElement>	最外层操作符对应的元素，自定义结构体 GroupElement

GroupElement 结构体说明：

字段	必选/可选	字段类型	描述
operator	可选	Enum	操作符 and/or，默认为 and
elements	必选	List<Json>	内层表达式运算的元素,一个 element

			表示一个 condition,自定义结构 conditions

Condition-DeviceData 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	条件触发 condition 类型，取值为 "DEVICE_DATA"
id	可选	String	对 condition 的编号，在一条规则范围内唯一，不指定时由系统自动编号
deviceInfo	必选	DeviceInfo	设备数据的定位信息，自定义结构，根据定位信息读取的数据作为比较表达式的左值
operator	必选	String	数据比较的操作符数据比较的操作符(目前支持的操作符有：>,<,>=,<=,=,between,geo.circle.in, geo.circle.out，具体使用见举例说明)
value	可选	String	比较表达式的右值(与 between 操作符联用时，右值表示最小值和最大值用逗号隔开如“20,30”表示大于等于 20 小于 30，应用场景:温度在 20 到 30 之间触发规则)
transInfo	可选	Json	不需要平台理解的信息
duration	可选	Integer	规则执行延时时间，单位为分钟，默认为 0 不延时
strategy	必选	Strategy	配置条件处理策略，自定义结构体 Strategy

Strategy 结构体说明：

字段	必选/可选	字段类型	描述
trigger	可选	String	pulse：上报的数据满足条件则触发，不判断上一次的数据 reverse：上一次数据不满足条件，本次数据满足条件则触发 默认为 pulse

eValidTime	可选	Integer (>=-1)	事件时效性，单位为 s，表示收到的传感器数据多久有效，（以 eventTime 为基准

DeviceInfo 结构体说明：

字段	必选/可选	字段类型	描述
deviceId	必选	String	设备 id
path	必选	String	以"/"分割的信息路径，按照当前的数据模型，路径为 ServiceId/DataProperty，例如门磁状态为"DoorWindow/status"

Condition-DeviceGroupData 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	条件触发 condition 类型，取值为 "DEVICE_DATA_GROUP"
id	可选	String	对 condition 的编号，在一条规则范围内唯一，不指定时由系统自动编号
deviceGroupInfo	必选	DeviceGroupInfo	组设备数据的定位信息，自定义结构，根据定位信息读取的数据作为比较表达式的左值
operator	必选	String	数据比较的操作符数据比较的操作符(目前支持的操作符有：>,<,>=,<=,=,between,geo.circle.in,geo.circle.out，具体使用见举例说明)
value	可选	String	比较表达式的右值(与 between 操作符联用时，右值表示最小值和最大值用逗号隔开如“20,30”表示大于等于 20 小于 30，应用场景:温度在 20 到 30 之间触发规则)
transInfo	可选	Json	不需要平台理解的信息
duration	可选	Integer	规则执行延时时间，单位为分钟，默认为 0 不延时

strategy	必选	Strategy	配置条件处理策略，自定义结构体 Strategy

DeviceGroupInfo 结构体说明：

字段	必选/可选	字段类型	描述
groupId	必选	String	设备组 id
path	必选	String	以"/"分割的信息路径，按照当前的数据模型，路径为 ServiceId/DataProperty，例如门磁状态为"DoorWindow/status",属性支持多个,比如 OBD 需要两个属性,经度和纬度”，多个属性中间以逗号分隔

Condition-DeviceTypeData 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	条件触发 condition 类型，取值为"DEVICE_DATA_TYPE"
id	可选	String	对 condition 的编号，在一条规则范围内唯一，不指定时由系统自动编号
deviceTypeInfo	必选	DeviceTypeInfo	组设备数据的定位信息，自定义结构，根据定位信息读取的数据作为比较表达式的左值
operator	必选	String	数据比较的操作符数据比较的操作符(目前支持的操作符有：>,<,>=,<=,=,between,geo.circle.in,geo.circle.out，具体使用见举例说明)
value	可选	String	比较表达式的右值(与 between 操作符联用时，右值表示最小值和最大值用逗号隔开如“20,30”表示大于等于 20 小于 30，应用场景:温度在 20 到 30 之间触发规则)
transInfo	可选	Json	不需要平台理解的信息

duration	可选	Integer	规则执行延时时间，单位为分钟，默认为 0 不延时
strategy	必选	Strategy	配置条件处理策略，自定义结构体 Strategy

DeviceTypeInfo 结构体说明：

字段	必选/可选	字段类型	描述
manufacturerId	必选	String	生产商 Id
model	必选	String	设备型号
deviceType	必选	String	设备类型
protocolType	必选	String	协议类型
path	必选	String	以"/"分割的信息路径，按照当前的数据模型，路径为 ServiceId/DataProperty，例如门磁状态为"DoorWindow/status",属性支持多个,比如 OBD 需要两个属性,经度和纬度”，多个属性中间以逗号分隔

Condition-DailyTimer 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	定点触发 condition 类型，“DAILY_TIMER”
id	可选	String	对 condition 的编号，在一条规则范围内唯一，不指定时由系统自动编号
time	必选	String	指定执行时间点，格式为"HH:MM"
daysOfWeek	可选	String	星期列表，以逗号分隔

Condition-CycleTimer 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	周期性触发 condition 类型，取值为“CYCLE_TIMER”

id	可选	String	对 condition 的编号，在一条规则范围内唯一，不指定时由系统自动编号
timeRange	必选	timeRange	限定的时间范围
interval	必选	Integer	周期，单位为分钟

Action-DeviceCMD 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	DEVICE_CMD
id	可选	String	对 action 的编号，在一条规则范围内唯一，不指定时由系统自动编号
appKey	可选	String(64)	应用实例的 key 值
deviceId	可选	String(64)	设备 id(不填时，命令下发给触发规则的设备)
cmd	必选	CMD	命令内容
cmdVersion	可选	String	命令下发的版本
cmdMetaData	可选	ObjectNode	命令下发的参数
transInfo	可选	Json	不需要平台理解的信息

CMD 结构说明：

字段	必选/可选	字段类型	描述
messageType	必选	String(128)	命令名字
messageBody	可选	Json	下发命令的消息内容
serviceId	必选	String(64)	服务 id

当 cmdVersion="V1.3.0"时，OjbectNode(cmdMetaData)结构说明：

字段	必选/可选	字段类型	描述
requestId	可选	String(64)	命令请求 Id

callbackUrl	可选	String(1024)	命令状态变化通知地址，当命令状态变化时(执行失败，执行成功，超时，发送，已送达)会通知 NA
expireTime	可选	Integer	已废弃参数！异步命令 1.3 版本创建规则 Demo

Action -SMS 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	发送短信 action 类型，取值为“SMS”
id	可选	String	对 action 的编号，在一条规则范围内唯一，不指定时由系统自动编号
msisdn	必选	String	短信通知地址
content	可选	String	通知内容通知内容（短信 content 为空时，短信邮件由北向应用发送）
subject	可选	String	通知主题
title	可选	String	短信别名
accountId	可选	String	账户 Id

Action -email 结构说明：

字段	必选/必选	字段类型	描述
type	必选	String	发送邮件 action 类型，取值为“EMAIL”
id	必选	String	对 action 的编号，在一条规则范围内唯一，不指定时由系统自动编号
email	必选	String	邮箱地址
content	必选	String	通知内容通知内容（邮件 content 或者 subject 为空时，短信邮件由北向应用发送）
subject	必选	String	通知主题
title	必选	String	邮件别名
accountId	必选	String	账户 Id

Action-DeviceAlarm 结构说明：

字段	必选/可选	字段类型	描述
type	必选	String	发送邮件 action 类型，取值为“DEVICE_ALARM”
id	可选	String	对 action 的编号，在一条规则范围内唯一，不指定时由系统自动编号
name	必选	String	告警名
status	必选	String	告警状态 fault/recovery
severity	必选	String	告警级别 warning/minor/major/critical
description	可选	String	告警描述

成功结果

状态码 StatusCode：200 OK

内容类型 Content-Type：application/json 返回结构：

字段	字段类型	描述
ruleId	String	规则实例 id

异常返回码

HttpStatus	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
401	101409	Authentication failure.	请求头域中没有携带 appId
200	100203	The application is not existed.	请求头域中携带 appId 不存在
200	101403	The rule is not existed.	规则不存在
200	100217	The application hasn't been authorized.	应用没有权限
200	101402	The rule name already existed.	规则名称已存在
500	100023	The data in dataBase is abnormal.	获取配置参数数据失败
200	101404	The rule count of each app has reached the upper limit, the current up limit is 30 entries.	应用已有规则数量查过规定的最大值

400	100216	The application input is invalid.	规则属性传参不合法
500	101424	Internal server error happened when create rule.	创建规则异常
200	101401	The rule input is invalid.	接口传参不合法
400	101416	The action of rule is invalid.	Action 参数不合法
400	101418	Action id duplicated in same rule.	Action id 重复
400	101425	Rule api parameter is invalid.	接口传参不合法
500	50204	Convert json to class failed.	数据格式转换失败

消息示例

```
Curl
curl -X POST
--header "Content-Type: application/json"
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules"
-d "{
  "name": "rule1",
  "author": "wudashan",
  "description": "this is a test rule 2015.12.23",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
      "type": "DEVICE DATA",
      "deviceInfo": {
        "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
        "path": "DoorWindow/status"
      },
      "operator": "=",
      "value": "OPEN",
      "strategy" :{
          "trigger": "pulse",
           "eValidTime": "-1"
      }
    }
  ],
  "actions": [
    {
      "type": "DEVICE CMD",
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "cmd": {
        "serviceId": "DoorWindow",
        "messageType": "MUTE",
        "messageBody": {
```

```
        "force": "true"
       }
      }
    }
  ]
}"
```

异步命令 1.3 版本创建规则 Demo

```
curl -X POST
--header "Content-Type: application/json"
--header "Accept: application/json"
--header "app_key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules"
-d "{
  "name": "rule1",
  "author": "wudashan",
  "description": "this is a test rule 2015.12.23",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
      "type": "DEVICE DATA",
      "deviceInfo": {
        "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
        "path": "DoorWindow/status"
      },
      "operator": "=",
      "value": "OPEN",
      "strategy" :{
         "trigger": "pulse",
          "eValidTime": "-1"
      }
    }
  ],
  "actions": [
    {
      "type": "DEVICE CMD",
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "cmd": {
        "serviceId": "DoorWindow",
        "messageType": "MUTE",
        "messageBody": {
          "force": "true"
        }
      },
      "cmdVersion": "V1.3.0"
    }
  ]
}"
```

异步命令 1.3 版本创建 between 表达式规则 Demo

```
curl -X POST
--header "Content-Type: application/json"
--header "Accept: application/json"
```

```
--header "app_key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules"
-d "{
  "name": "rule1",
  "author": "wudashan",
  "description": "this is a test rule 2015.12.23",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
      "type": "DEVICE_DATA",
      "deviceInfo": {
        "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
        "path": "Temperature/temperature"
      },
      "operator": "between",
      "value": "20,30",
       "strategy":{
        "trigger":"pulse",
        "eValidTime":"-1"
      }
    }
  ],
  "actions": [
    {
      "type": "DEVICE CMD",
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "cmd": {
        "serviceId": "DoorWindow",
        "messageType": "MUTE",
        "messageBody": {
          "force": "true"
        }
      },
      "cmdVersion": "V1.3.0"
    }
  ]
}"
```

异步命令 1.3 版本创建 geo.circle.in/out 表达式规则 Demo

```
curl -X POST
--header "Content-Type: application/json"
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules"
-d "{
  "name": "rule1",
  "author": "wudashan",
  "description": "this is a test rule 2015.12.23",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
```

```
    "type": "DEVICE_DATA",
    "deviceInfo": {
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "path": "Location/longitude,latitude"
    },
    "operator": "geo.circle.in",
    "value": "114074792,22641851,100",
     "strategy":{
      "trigger":"pulse",
      "eValidTime":"-1"
    }
   }
 ],
 "actions": [
   {
    "type": "DEVICE_CMD",
    "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
    "cmd": {
      "serviceId": "DoorWindow",
      "messageType": "MUTE",
      "messageBody": {
        "force": "true"
      }
    },
    "cmdVersion": "V1.3.0"
   }
 ]
}"
```

返回结果

```
{
  "ruleId":"*******"
}
```

1.2.7.2 更新规则

接口功能

更新规则

请求方法

PUT

请求路径

https://server:port/iocm/app/rule/v1.2.0/rules

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id。
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的 accessToken。

消息体参数定义查看 RuleDTO1.2 定义说明。

成功结果

状态码 StatusCode：200 OK

内容类型 Content-Type：application/json

返回结构

字段	字段类型	描述
ruleId	String	规则实例 id

异常返回码

HttpStatus	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
400	101419	RuleId can not be null or empty when update rule.	ruleId 不能为空
401	101409	Authentication failure.	请求头域中没有携带 appId
200	100203	The application is not existed.	请求头域中携带 appId 不存在
200	101403	The rule is not existed.	规则不存在
200	100217	The application hasn't been authorized.	应用没有权限
200	101402	The rule name already existed.	规则名称已存在
500	100023	The data in dataBase is abnormal.	获取配置参数数据失败
200	101404	The rule count of each app has reached the upper limit, the current up limit is 30 entries.	应用已有规则数量查过规定的最大值
400	100216	The application input is invalid.	规则属性传参不合法

500	101424	Internal server error happened when create rule.	创建规则异常
200	101401	The rule input is invalid.	接口传参不合法
400	101416	The action of rule is invalid.	Action 参数不合法
400	101418	Action id duplicated in same rule.	Action id 重复
400	101425	Rule api parameter is invalid.	接口传参不合法
500	50204	Convert json to class failed.	数据格式转换失败

消息示例

```
Curl
curl -X PUT
--header "Content-Type: application/json"
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules"
-d "{
  "ruleId":"rule1"
  "name": "rule1",
  "author": "wudashan",
  "description": "this is a test rule 2015.12.23",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
      "type": "DEVICE DATA",
      "deviceInfo": {
        "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
        "path": "DoorWindow/status"
      },
      "operator": "=",
      "value": "OPEN"
    }
  ],
  "actions": [
    {
      "type": "DEVICE CMD",
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "cmd": {
        "serviceId": "DoorWindow",
        "messageType": "MUTE",
        "messageBody": {
          "force": "true"
        }
      }
    }
  ]
}"
```

异步命令 1.3 版本更新规则 Demo

```
curl -X POST
--header "Content-Type: application/json"
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules"
-d "{
  "ruleId":"rule1"
  "name": "rule1",
  "author": "wudashan",
  "description": "this is a test rule 2015.12.23",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
      "type": "DEVICE DATA",
      "deviceInfo": {
        "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
        "path": "DoorWindow/status"
      },
      "operator": "=",
      "value": "OPEN"
    }
  ],
  "actions": [
    {
      "type": "DEVICE CMD",
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "cmd": {
        "serviceId": "DoorWindow",
        "messageType": "MUTE",
        "messageBody": {
          "force": "true"
        }
      },
      "cmdVersion": "V1.3.0"
    }
  ]
}"
```

返回结果

```
{
  "ruleId": "********"
}
```

1.2.7.3 修改规则状态

接口功能

修改规则状态

请求方法

PUT

请求路径

https://server:port/iocm/app/rule/v1.2.0/rules/{ruleId}/status/{status}

注意事项

```
携带头域信息
Header：
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的 accessToken
ruleId	必选	String	path	规则的 ID
status	必选	String	path	规则的状态，“active”代表激活状态，“inactive”代表未激活

成功结果

状态码 StatusCode：200 OK

异常返回码

HttpStatus	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
401	101409	Authentication failure.	请求头域中没有携带 appId
200	100203	The application is not existed.	请求头域中携带 appId 不存在
200	101403	The rule is not existed.	规则不存在
200	100217	The application hasn't been authorized.	应用没有权限
200	101401	The rule input is invalid.	接口传参不合法

消息示例

```
Curl
curl -X POST
--header "Content-Type: application/json"
--header "Accept: application/json"
--header "app_key: string"
--header "Authorization: Bearer {accessToken}"
-d "{
  "status": "******"
}"
"http://server:port/iocm/app/rule/v1.1.0/rules/string"
```

返回结果

NA

1.2.7.4 删除规则

接口功能

删除规则

请求方法

DELETE

请求路径

https://server:port/iocm/app/rule/v1.2.0/rules/{ruleId}

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的 accessToken
ruleId	必选	String	path	规则实例的 id

成功结果

状态码 StatusCode：204 No Content

异常返回码

HttpStatus	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
401	101409	Authentication failure.	请求头域中没有携带 appId
200	100203	The application is not existed.	请求头域中携带 appId 不存在
200	101403	The rule is not existed.	规则不存在
200	100217	The application hasn't been authorized.	应用没有权限

消息示例

```
Curl
curl -X DELETE
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"http://server:port/iocm/app/rule/v1.1.0/rules/{ruleId}"
```

返回接口

NA

1.2.7.5 查找规则

接口功能

查找规则

请求方法

GET

请求路径

https://server:port/iocm/app/rule/v1.2.0/rules?name={name}&author={author}

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的 accessToken
author	必选	String	query	用户 id
name	可选	String	query	规则名称

成功结果

状态码 StatusCode：200 OK

返回结果：Rule 列表，消息体参数定义查看 RuleDTO1.2 定义说明。

异常返回码

HttpStatus	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
200	100217	The application hasn't been authorized.	应用没有权限

消息示例

```
Curl
curl -X GET
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"http://server:port/iocm/app/rule/v1.2.0/rules?name={name}&author={author}
```

消返回结果

```
{
  "name": "****",
  "author": "*****",
  "description": "this is a test rule 2015.12.23",
  "type": "noType",
  "status": "active",
  "logic": "and",
  "conditions": [
    {
      "type": "DeviceData",
      "id": 0,
      "deviceInfo": {
```

```
      "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
      "path": "DoorWindow/status"
    },
    "operator": "=",
    "value": "OPEN"
  }
 ],
 "actions": [
  {
    "type": "DeviceCMD",
    "id": 0,
    "appId": "pnS6yhwLszJ8kZF7HhuH7cOeoGQa",
    "deviceId": "0285f8e6-f0f1-441a-bc27-30f02545658c",
    "cmd": {
      "serviceId": "DoorWindow",
      "messageType": "MUTE",
      "messageBody": {
        "force": "true"
      }
    }
  }
 ]
}
```

1.2.7.6 批量修改规则状态

接口功能

批量修改规则状态

请求方法

PUT

请求路径

https://server:port/iocm/app/rule/v1.2.0/rules/status

注意事项

```
携带头域信息
Header:
"app key:{appId}"
"Authorization:Bearer {accessToken}"
Content-Type:application/json;
```

参数说明

参数	必选/可选	类型	位置	描述
app_key	必选	String	header	鉴权用户名，填写创建应用时返回的 id
Authorization	必选	String	header	鉴权信息，填写鉴权接口返回的

				accessToken
requests	必选	List<RuleStatusUpdateReqDTO>	body	请求结构体列表，RuleUpdateRequestDTO 为自定义结构体，见下表

RuleStatusUpdateReqDTO 结构体说明：

参数	必选/可选	类型	描述
ruleId	必选	String	规则 Id
status	必选	String	状态，取值可为“active”和“inactive”。

成功结果

状态码

StatusCode：200 OK

内容类型

Content-Type：application/json

返回结果：

参数	类型	描述
result	RuleResultBodyDTO	总体执行结果，自定义结构体，具体结构见下表
responses	List<RuleResponseBodyDTO>	每个条目的执行结果，自定义结构体，具体结构见下表

RuleResultBodyDTO 结构体说明：

参数	类型	描述
result_code	String	总体执行结果代码，若执行成功则为“OK”，否则为“Partial OK”
result_desc	String	总体执行结果描述，若全

		部执行成功则为“Operation successful.”，否则为“Operation partial ok.”

RuleResponseBodyDTO 结构体说明：

参数	类型	描述
ruleId	String	规则 Id
response	ExceptionMsg	执行结果，如果成功则不带，否则包含异常信息

异常返回码

HttpStatus	error_code	error_desc	说明
401	100002	Invalid access token.	错误的 token 信息
401	101409	Authentication failure.	请求头域中没有携带 appId
200	100203	The application is not existed.	请求头域中携带 appId 不存在
200	101403	The rule is not existed.	规则不存在
200	101401	The application hasn't been authorized.	应用没有权限
200	101401	The rule input is invalid.	接口传参不合法

消息示例

```
Curl
curl -X GET
--header "Accept: application/json"
--header "app key: string"
--header "Authorization: Bearer {accessToken}"
"https://server:port/iocm/app/rule/v1.2.0/rules/status"
-d "{
   "requests":[{
 "ruleId":"******ruleid 1",
 "status":"active"
 },{
 "ruleId":"******ruleid 2",
 "status":"inactive"
 }
 ]
```

```
}"
```

返回结果

```
{
  "result": {
 "result_code":"Partial OK",
 "result_desc":"Operation partial ok"
  },
  "responses": [
    {
     "ruleId":"******ruleid_1",
      "response": {
        "error_code": "*********",
        "error_desc": "*********"
      }
    },{
     "ruleId":"******ruleid_2",
      "response": {
        "error code": "*********",
        "error desc": "*********"
      }
    }
  ]
}
```

1.2.8 消息推送

Application 向平台订阅设备变更通知（1.2.3.3 Application 订阅平台数据），当设备发生变更时，平台向 Application 推送此消息（Application 启动初始化订阅），Application 根据通知类型进行对消息分派对应的服务模块进行处理。

Application 订阅接口：参考 1.2.3.3 Application 订阅平台数据。

1.2.8.1 注册直连设备的通知

接口功能

NA 调用中国电信物联网开放平台的接口添加新设备，获取设备验证码（参见 1.2.2.1 注册直连设备），中国电信物联网开放平台使用该接口功能将该信息通知给 NA，平台通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceAdded”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

注意事项

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型：deviceAdded
deviceId	必选	string	设备唯一标识
resultCode	必选	string	取值：expired 过期，不成功； succeeded 成功
deviceInfo	可选	DeviceInfo	参见 1.3.1 DeviceInfo 结构体说明

消息示例

Body

{

"deviceId":"74febc35-805f-4c43-9189-649d5d7fff77",

"resultCode":"succeeded",

"deviceInfo":{

"name":"Sensor_12",

"manufacturer":"wulian",

"deviceType":90",

"model":"90",

"mac":"C7EA1904004B1204",

"swVersion":"…",

"fwVersion":"…",

"hwVersion":"…",

"protocolType":"zigbee",

"description":"smock detector",

"nodeType":"GATEWAY",

}

}

1.2.8.2 发现非直连设备的通知

接口功能

中国电信物联网开放平台接收到网关添加新设备，中国电信物联网开放平台使用该接口功能将该信息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceAdded”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明：

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，deviceAdded（添加设备）
deviceId	必选	String	设备唯一标识
gatewayId	必选	String	网关唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
nodeType	必选	String	指定节点的类型 ENDPOINT：传感器. GATEWAY：网关.
deviceInfo	必选	DeviceInfo	设备信息，设备信息变化、添加、删除设备时会带此参数，定义参见 1.3.1 DeviceInfo 结构体说明

添加新设备：

Body

{

"notifyType ": "deviceAdded",

"deviceId":"7b3979fc-b072-433b-b3f6-673072e1bc04",

"gatewayId":"**********",

"deviceInfo":{

"name":"Sensor_12",

"manufacturer":"wulian",

"type":90",

"model":"90",

"mac":"C7EA1904004B1204",

"swVersion":"…",

"fwVersion":"…",

"hwVersion":"…",

"protocolType":"zigbee",

"description":"smock detector"

}

}

1.2.8.3 设备信息变化的通知

接口功能

中国电信物联网开放平台接收到设备信息变化（静态信息，如设备名称、厂商 id），中国电信物联网开放平台使用该接口功能将该信息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceInfoChanged”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明：

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型， deviceInfoChanged（设备信息变化）
deviceId	必选	String	设备唯一标识
gatewayId	必选	String	网关唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
deviceInfo	必选	DeviceInfo	设备信息，设备信息变化、添加、删除设备时会带此参数，定义参见 1.3.1 DeviceInfo 结构体说明

设备信息变化：

Body

{

"notifyType ": "deviceInfoChanged",

"deviceId":"7b3979fc-b072-433b-b3f6-673072e1bc04",

"gatewayId":"**********",

"deviceInfo":{

}

}

1.2.8.4 设备数据变化的通知

接口功能

中国电信物联网开放平台接收到设备数据变化（动态变化，如，服务属性值的变化），中国电信物联网开放平台使用该接口功能将该信息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceDataChanged”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明：

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，deviceDataChanged（设备数据变化）
requestId	可选	String	关联消息的 Id
deviceId	必选	String	设备唯一标识
gatewayId	可选	String	网关唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
service	必选	DeviceServiceData	设备的 service 数据，只有设备数据变化时会带此参数，定义参见下表

DeviceServiceData

字段	必选/可选	类型	描述
serviceId	必选	String	服务标识
serviceType	可选	String	服务类型

字段	必选/可选	类型	描述
data	可选	ObjectNode	服务属性信息
eventTime	可选	String	事件产生时间,时间格式：yyyyMMdd’T’HHmmss’Z’ 如： 20151212T121212Z

消息事例

Body

{

" notifyType ": "deviceDataChanged",

"requestId": "{requestId}",

"timestamp":"{timestamp}",

"deviceId":"7b3979fc-b072-433b-b3f6-673072e1bc04",

"gatewayId":"**********",

"service": {

"serviceType": "",

"serviceId": "",

"data": {"battery_low": 1},

"eventTime":"{eventTime}"

}

}

1.2.8.5 删除非直连设备的通知

接口功能

中国电信物联网开放平台接收到删除非直连设备，中国电信物联网开放平台使用该接口功能将该信息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceDeleted”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明：

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，deviceDeleted（删除设备）
deviceId	必选	String	设备唯一标识
gatewayId	必选	String	网关唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。

删除设备：

Body

{

"notifyType ": "deviceDeleted",

"deviceId":"7b3979fc-b072-433b-b3f6-673072e1bc04",

"gatewayId":"**********",

}

1.2.8.6 消息确认通知

接口功能

中国电信物联网开放平台接收到网关的确认消息（如下发命令给网关，网关确认接收到该消息），将该消息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“messageConfirm”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，messageConfirm (消息确认)
header	必选	Header	见 header 字段说明
body	必选	ObjectNode	根据业务具体定义，确认消息可以携带状态变化等消息

Header **字段说明：**

字段	必选/可选	类型	描述
requestId	必选	String	消息的序列号，唯一标识该消息，1-128 字节
from	必选	String	表示消息发布者的地址： 网关或设备发起的请求：/devices/{deviceId} 设备服务发起的请求：/devices/{deviceId}/services/{serviceId} 1-128 字节
to	必选	String	表示消息接收者的地址，1-128 字节 To 就是请求中的 From，如 APP 的 userId
status	必选	String	命令状态：1-32 字节 sent --- 已发送 delivered --- 已送达 executed --- 已执行
timestamp	必选	String	时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z 1-32 字节

网关确认消息通知

Body

{

"notifyType"： "messageConfirm",

"header":

{

"requestId":"{ requestId }",

"from": "{from}",

"to": "{to}",

"status":"delivered",

"timestamp":"{timestamp}"

}

"body":

{

JSONBody

}

}

1.2.8.7 设备响应命令通知

接口功能

中国电信物联网开放平台接收到设备（包括网关设备和普通设备）的响应命令（如下发命令给设备，设备执行命令后，发送响应命令给平台，如视频呼叫、视频录制，抓图等），将该消息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“commandRsp”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

注意事项

commandRsp 对应的 command 是指使用设备服务接口下发的命令。请参考 1.2.5 设备服务调用。

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，commandRsp (响应命令)
header	必选	Header	见 Header 字段说明
body	必选	ObjectNode	响应命令的消息体 Body 中的内容

Header

字段	必选/可选	类型	描述
requestId	必选	String	消息的序列号，唯一标识该消息 ，1-128 个字节
from	必选	String	1-128 个字节，表示消息发布者的

字段	必选/可选	类型	描述
			地址： 网关或设备发起的请求：/devices/{deviceId} 设备服务发起的请求：/devices/{deviceId}/services/{serviceId}
to	必选	String	表示消息接收者的地址，1-128 个字节 To 就是请求中的 From，如 APP 的 userId
timestamp	可选	String	时间格式：yyyyMMdd’T’HHmmss’Z’ 如：20151212T121212Z，1-32 个字节
deviceId	必选	String	设备唯一标识（命令所属设备的 ID）
serviceType	必选	String	命令所属服务类型
method	必选	String	(1-32 个字节)命令名称，存在于设备服务能力下的命令名称字段，如："DISCOVERY" "REMOVE" 。详情参考 profile 全量模板

设备命令响应通知

Body

{

"notifyType":"commandRsp",

"header":{

"requestId":"{requestId}",

"from":"/devices/{deviceId}/services/{serviceId}",

"to":"/apps/{appId}/users/{userId}",

"deviceId":"{deviceId}",

"serviceType":"{serviceType}",

"method":"{commandRsp}",

},

"body":

{

...

}

}

1.2.8.8 设备事件通知

接口功能

中国电信物联网开放平台接收到设备发出的事件（如设备 UDS 存储空间不足事件），将该消息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceEvent”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，deviceEvent (设备事件)
header	必选	Header	见 Header 字段说明
body	必选	ObjectNode	响应命令的消息体 Body 中的内容

Header

字段	必选/可选	类型	描述
eventType	必选	String	事件类型，1-32 个字节
from	必选	String	1-128 个字节，表示消息发布者的地址，设备服务发起的请求：/devices/{deviceId}/services/{serviceId}
to	可选	String	1-128 个字节，当 catogery 为“event”时请求地址： 上报设备服务事件：/event/v1.1.0/devices/{deviceId}/services/{serviceId}
timestamp	必选	String	时间格式：yyyyMMdd’T’HHmmss’Z’ 如：

字段	必选/可选	类型	描述
			20151212T121212Z，1-32 个字节
eventTime	必选	String	时间格式：yyyyMMdd’T’ HHmmss’Z’ 如：20151212T121212Z，1-32 个字节
deviceId	必选	String	设备唯一标识（命令所属设备的 ID）
serviceType	必选	String	命令所属服务类型

消息事例

Body

{

"notifyType":"deviceEvent",

"header":{

"eventType":"{eventType}",

"from":"/devices/{deviceId}/services/{serviceId}",

"to":"/event/v1.1.0/devices/{deviceId}/services/{serviceId}",

"deviceId":"{deviceId}",

"serviceType":"{serviceType}",

"timestamp":"{timestamp}",

eventTime":"{eventTime}",

},

"body":

{

usedPercent":80

}

}

1.2.8.9 设备服务上报通知

接口功能

中国电信物联网开放平台接收到设备服务信息变化，中国电信物联网开放平台使用该接口功能将该信息通知给 Application。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“serviceInfoChanged”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明:

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，serviceInfoChanged（设备信息变化）
deviceId	必选	String	设备唯一标识
gatewayId	必选	String	网关唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
serviceId	必选	String	设备服务标识
serviceType	必选	String	设备服务类型
serviceInfo	必选	ServiceInfo	设备服务信息（增量上报），见下表

ServiceInfo

字段	必选/可选	类型	描述
muteCmds	必选	List<String>	隐私模式的命令："VIDEO_RECORD", "INVITE-INIT", "INVITE", "IMAGE_FIXED_CAPTURE"

消息事例

Body

{

"notifyType":"serviceInfoChanged",

"deviceId":"7b3979fc-b072-433b-b3f6-673072e1bc04",

"gatewayId":"**********",

"serviceType":"{ serviceType}",

"serviceId": "{serviceId}",

"serviceInfo":

{

"muteCmds":"VIDEO_RECORD"

}

}

1.2.8.10 规则事件上报通知

接口功能

中国电信物联网开放平台规则触发后生成对应的规则事件通知 NA。

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“ruleEvent”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明:

字段	必选/可选	类型	描述
notifyType	必选	String	"ruleEvent"
author	必选	String	同规则接口中的 author
ruleId	必选	String	规则 id
ruleName	必选	String	规则名称
logic	可选	String	多条件逻辑关系
reasons	必选	List<Reason>	触发原因，对应 conditions
triggerTime	必选	String	规则触发时间
actionsResults	必选	List<ActionResult>	动作执行结果

Reason 字段说明：

字段	必选/可选	类型	描述
satisfactionTime	必选	String	条件满足时间
type	必选	String	条件类型，包括 DEVICE_DATA，

字段	必选/可选	类型	描述
			DailyTimer，CycleTimer 三种
id	可选	String	对应 condition 的 id
info	可选	Json	不同条件类型携带不同信息
transInfo	可选	Json	事件推送时的回填信息，对应规则的 condition 中的 transInfo

DEVICE_DATA 类型的条件携带的 info：

字段	必选/可选	类型	描述
realValue	可选	String	设备数据的实际值

ActionResult 字段说明：

字段	必选/可选	类型	描述
type	必选	String	action 类型
id	可选	String	对应 action 的 id
exception	与 result 二选一	String	规则引擎构造 action 对应的请求过程中遇到异常时携带的异常信息
result	与 exception 二选一	Json	动作的执行结果，对于 DEVICE_CMD/SMS/EMAIL 类型的动作，内容为 statusCode+body，详细信息参考接口文档中对应接口的说明
info	可选	Json	不同动作类型携带不同信息
transInfo	可选	Json	事件推送时的回填信息，对应规则的 action 中的 transInfo

DEVICE_CMD 类型的动作携带的 info

字段	必选/可选	类型	描述
requestId	必选	String	下发异步命令时，为请求消息分配的 UUID

消息事例

Body

{ "notifyType": "ruleEvent",

"author": "*****",

"ruleId": "**********",

"ruleName": "name",

"reasons": [{

"satisfactionTime": " yyyyMMddTHHmmssZ ",

"type": "******"

}

],

"triggerTime": "yyyyMMddTHHmmssZ",

"actionsResults": [{

"type": "******",

"exception": "******"

}

]

}

1.2.8.11 绑定设备通知

接口功能

即设备初始登陆通知，网关设备在平台注册绑定（网关设备在平台创建对应设备的信息并获取密码的过程），平台通知给 NA。

调用方法

POST

接口路径

{callbackUrl}

此处的接口路径是应用向平台订阅数据时注册的对应 notifyType 为“bindDevice”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	通知类型，取值：bindDevice
deviceId	必选	string	设备 Id

resultCode	必选	string	绑定结果，取值：expired/succeeded
deviceInfo	可选	DeviceInfo	设备信息，参见 1.3.1 DeviceInfo 结构体说明

返回结果

无

消息示例

Method：POST

request：

Body

{

"notifyType": "bindDevice",

"deviceId": "74febc35-805f-4c43-9189-649d5d7fff77",

"resultCode": "succeeded",

"deviceInfo": {

"name": "Sensor_12",

"manufacturer": "wulian",

"deviceType": 90,

"model": "90",

"mac": "C7EA1904004B1204",

"swVersion": "…",

"fwVersion": "…",

"hwVersion": "…",

"protocolType": "zigbee",

"description": "smockdetector",

"nodeType": "GATEWAY"

}

}

1.2.8.12 批量设备数据变化上报通知

接口功能

批量设备数据变化后消息通知

调用方法

POST

接口路径

{callbackUrl}

此处的**接口路径**是应用向平台订阅数据时注册的对应 notifyType 为“deviceDatasChanged”的回调通知地址，请参考 1.2.3.3 Application 订阅平台数据。

参数说明

字段	必选/可选	类型	描述
notifyType	必选	String	deviceDatasChanged
requestId	必选	String	标识命令消息
deviceId	必选	String	设备唯一标识
gatewayId	必选	String	网关唯一标识
services	必选	List<DeviceServiceData>	服务

DeviceServiceData 结构：

字段	必选/可选	类型	描述
serviceId	必选	String	服务 ID
serviceType	必选	String	服务类型
data	必选	ObjectNode	数据
eventTime	必选	String	时间格式 yyyymmddThhmissZ，例如, 20151212T121212Z.

返回结果

无

消息示例

Body

{

"notifyType": "deviceDatasChanged",

"requestId": "**********",

"deviceId": "7b3979fc-b072-433b-b3f6-673072e1bc04",

"gatewayId": "**********",

"service": {

"serviceId": "Battery",

"serviceType": "Battery",

"data": {},

"eventTime": "20170101120000"

}

}

1.3 常用数据结构体定义

1.3.1 DeviceInfo 结构体说明

字段	必选/可选	类型	描述
nodeId	必选	String	设备唯一标识
name	可选	String	设备名称
description	可选	String	设备描述
manufactureId	必选	String	供应商标识
manufacturerName	可选	String	供应商名字
mac	可选	String	物理地址
location	可选	String	设备位置
deviceType	必选	String	设备种类： MultiSensor ContactSensor Camera Gateway
model	必选	String	设备型号. 如果是 Z-Wave 协议，型号就是 ProductType + ProductId（用零填充，如果需要） 格式的十六进制数,

字段	必选/可选	类型	描述
			例如 001A-0A12. 如果是其他协议，型号需要另一种格式表示
swVersion	可选	String	使用 Z-Wave 协议时的软件版本号。其格式是 major version number.minor version number, 例如, 1.1.
fwVersion	可选	String	固件版本号
hwVersion	可选	String	硬件版本号
protocolType	必选	String	协议类型 (Z-Wave).
signalStrength	可选	String	信号强度
bridgeId	可选	String	网桥标识
supportedSecurity	可选	String	是否支持安全模式，其取值为 TRUE 或 FALSE
isSecurity	可选	String	设备是否安全，在安全模式为 TRUE 的情况下有效，其取值为 TRUE 或 FALSE
sigVersion	可选	String	信令版本
runningStatus	可选	String	设备运行状态： NORMAL：正常 ABNORMAL：异常 FAULT：故障
status	可选	String	设备状态. ONLINE: 在线 OFFLINE: 下线 INBOX: 停用.
statusDetail	可选	String	设备状态的详细情况
mute	可选	String	设备是否进行屏蔽 TRUE: 被屏蔽. FALSE: 没被屏蔽.

1.3.2 DeviceService 结构体说明

字段	必选/可选	类型	描述

字段	必选/可选	类型	描述
serviceType	必选	String	服务类型
serviceId	必选	String	服务 ID 标识
data	必选	JSON String	指定一个属性值对(AVP).
eventTime	必选	DateTime	时间格式 yyyymmddThhmissZ，例如, 20151212T121212Z.
serviceInfo	可选	ServiceInfo	业务信息，此字段用于内部控制功能，不用关心此字段

1.3.3 QueryDeviceDTOCloudNA 结构体说明

字段	必选/可选	类型	描述
deviceId	必选	String	设备唯一标识
gatewayId	必选	String	网关唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
nodeType	必选	String	指定节点的类型 ENDPOINT：传感器. GATEWAY：网关.
createTime	必选	String	设备注册时间
lastModifiedTime	必选	String	设备信息最后修改时间
deviceInfo	必选	DeviceInfo	详见"1.3.1 DeviceInfo 结构体说明"
services	可选	List<DeviceService>	详见 "1.3.2 DeviceService 结构体说明"

1.3.4 ServiceCommand 结构体说明

字段	必选/可选	类型	描述
commandName	可选	String	命令
paras	可选	List<ServiceCommandPara>	属性列表，参考下

字段	必选/可选	类型	描述
			表

ServiceCommandPara 结构体

字段	必选/可选	类型	描述
paraName	可选	String	参数名
dataType	可选	String	数据类型
required	可选	boolean	是否必选
min	可选	int	最小
max	可选	int	最大
step	可选	Double	步长
maxLength	可选	int	最大长度
unit	可选	String	单位（符号）
enumList	可选	List<String>	枚举类型列表

1.3.5 ServiceProperty 结构体说明

字段	必选/可选	类型	描述
propertyName	可选	String	属性名
dataType	可选	String	数据类型
required	可选	boolean	是否必选
min	可选	int	最小
max	可选	int	最大
step	可选	double	步长
maxLength	可选	int	最大长度
method	可选	String	访问方法：RW 可读 R，可写 W
unit	可选	String	单位（符号）
enumList	可选	List<String>	枚举类型列表

1.3.6 DeviceCommandResp 结构体说明

字段	必选/可选	类型	描述
commandId	必选	String(1-64)	设备命令 ID
appId	必选	String(1-64)	设备命令撤销任务所属的应用 ID
deviceId	必选	String(1-64)	设备命令指定的设备 ID
command	必选	CommandDTOV4	下发命令的信息，定义参见附录 1.3.7 CommandDTOV4 结构体说明
callbackUrl	可选	String(1024)	命令状态变化通知地址，必须使用 HTTPS 信道回调地址 (说明：HTTP 信道只可用于调测，不能用于商用环境)； 当命令状态变化时(执行失败，执行成功，超时，发送，已送达)会通知 NA，平台会以 POST 方式发送 HTTP 消息给应用，请求 Body 为 json 字符串，格式形如： {"deviceId":"deviceId","commandId":"commandId","result":{"status":"SUCCESS","result":{…}}}
expireTime	可选	Integer(>=0)	下发命令有效的超期时间，单位为秒，表示设备命令在创建后 expireTime 秒内有效，超过这个时间范围后命令将不再下发，如果未设置则默认为 48 小时
status	必选	String	设备命令的状态，DEFAULT 表示未下发，EXPIRED 表示命令已经过期，SUCCESSFUL 表示命令已经成功执行，FAILED 表示命令执行失败，TIMEOUT 表示命令下发执行超时，CANCELED 表示命令已经被撤销执行
result	可选	ObjectNode	设备命令执行的详细结果，格式 为 json 字符串
creationTime	必选	String(20)	设备命令的创建时间
executeTime	可选	String(20)	设备命令执行的时间
platformIssuedTime	可选	String(20)	平台发送设备命令的时间

deliveredTime	可选	String(20)	平台将设备命令送达到设备的时间
issuedTimes	可选	Integer(>=0)	平台发送设备命令的次数

1.3.7 CommandDTOV4 结构体说明

字段	必选/可选	类型	描述
serviceId	必选	String(1-64)	命令对应的服务 ID，不能为空字符串
method	必选	String(1-128)	命令服务下具体的命令名称，服务属性等，不能为空字符串
paras	可选	ObjectNode	命令参数的 jsonString，具体格式需要应用和设备约定。

1.3.8 GetDeviceRspDTO 结构体说明

字段	类型	描述
deviceId	String	设备唯一标识，1-64 个字节
gatewayId	String	设备唯一标识。当设备是直连设备时，gatewayId 为设备本身的 deviceId。当设备是非直连设备时，gatewayId 为设备所关联的直连设备（即网关）的 deviceId。
nodeType	String	指定节点的类型 ENDPOINT: 传感器. GATEWAY: 网关.
createTime	String	设备注册时间；时间格式：yyyyMMdd’T’HHmmss’Z’ 如： 20151212T121212Z
lastModifiedTime	String	最新修改时间；时间格式：yyyyMMdd’T’HHmmss’Z’ 如： 20151212T121212Z
deviceInfo	DeviceInfo	DeviceInfo 的定义参见 1.3.1 DeviceInfo 结构体说明
services	List<DeviceService>	DeviceService 的定义参见 1.3.2 DeviceService 结构体说明
connectionInfo	ConnectionInfo	连接信息
location	Location	位置信息

字段	类型	描述
devGroupIds	List<String>	设备组 Id 标签列表

1.3.9 CommandNA2CloudHeader 结构体说明

字段	必选/可选	类型	描述
requestId	可选	String	如果填写该 requestId，用填写的 requestId 标识一个命令，如果不填写就由平台生成。1-32 个字节
mode	必选	String	是否要确认消息，不需要确认消息：“NOACK” ，需要确认消息：“ACK”，其它值无效。
from	可选	String	(1-128 个字节)表示消息发布者的地址： Application 发起的请求：/{serviceName}
to	可选	String	接收者地址
toType	可选	String	取值范围：CLOUD|GATEWAY
method	必选	String	发现命令名称："DISCOVERY" 删除命令名称："REMOVE"
callbackURL	可选	String	回调的 url 地址。必须使用 HTTPS 信道回调地址，同时回调地址中必须指定回调地址的端口。(说明：HTTP 信道只可用于调测)

1.3.10 ApplicationSetEncrtptDTO 结构体说明

字段	必选/可选	类型	描述
flag	必选	Integer	加密标志，取值 0 表示不加密；取值 2 表示 AES 加密，key 和 iv 必填
iv	可选	String(16)	iv 和 key 共同决定加密方式
key	必选	String(16)	iv 和 key 共同决定加密方式

1.3.11 PutCarInfoData 结构体说明

字段	必选/可选	类型	描述
engineModel	可选	String(256)	发动机型号
vin	可选	String(256)	VIN
plateNumber	可选	String(256)	车牌号
simNumber	可选	String(256)	SIM 卡号(注：首次创建车辆信息时必须携带 simNumber；修改车辆信息时可选)
carModel	可选	String(256)	车辆型号(注：首次创建车辆信息时必须携带 carModel，修改车辆信息时可选)
terminalID	可选	String(256)	终端 ID
iccid	可选	String(256)	ICCID
imei	可选	String(256)	IMEI
encryptFlag	可选	Integer	加密标志，取值 0 表示不加密，取值 2 表示加密
encryptIv	可选	String(256)	encryptIv 和 encryptKey 共同决定加密方式
encryptKey	可选	String(256)	encryptIv 和 encryptKey 共同决定加密方式