

设备接入指南

目录

1	接入方式.....	2
1.1	一机一密.....	2
1.2	一型一密.....	2
2	MQTT 协议.....	3
2.1	事件上报.....	5
2.2	事件上报响应.....	5
2.3	属性上报.....	5
2.4	属性上报响应.....	6
2.5	属性设置.....	6
2.6	属性设置响应.....	6
2.7	服务调用.....	6
2.8	服务调用响应.....	7

前言

本文档主要介绍设备接入平台的基本步骤以及 MQTT 相关协议。

平台目前支持一机一密与一型一密两种接入方式，设备与平台的连接协议采用 MQTT，通过发布/订阅特定 Topic 的方式上报设备事件、接收用户传递的指令。

无论是一机一密还是一型一密，设备端最终都需要通过三元组 (product_key, device_id, password) 以 mqtt 协议接入平台，其中 product_key 为产品 key，可以通过 pms 查看产品详情获取，device_id 与 password 通过 pms 生产设备或者设备调用注册接口获取。

1 接入方式

1.1 一机一密

每个设备都有唯一的设备 ID 与密码，设备在生产过程中就需要预烧录这些信息。按照以下步骤实现一机一密接入：

- 1、 登录 PMS，在设备管理-设备生产，选择生产的产品与数量添加生产任务。
- 2、 生产成功后，点击生产任务的“管理”-“正式入库”。
- 3、 入库成功后，点击生产任务的-设备列表-下载 CSV，即可获取生产成功的设备 ID 与密码等数据。
- 4、 产线将此设备 ID 与密码烧录到设备中，编写相应程序实现功能。

1.2 一型一密

采用一型一密的方式不需要预先烧录设备密码与设备 ID，但是需要

预先烧录设备所属产品的 product_key 与 product_secret，具体步骤如下：

- 1、 进入 PMS 后台，产品管理-点击“产品名称”查看产品详情，获取到 product_key、product_secret。
- 2、 将获取到的 product_key、product_secret 预烧录到设备中。
- 3、 设备上电后，判断是否存在设备 ID 与密码，如果没有，则调用 apiservice 的设备注册接口，注册成功即获取设备 ID 与密码，设备端本地永久存储设备 ID 与密码。设备注册接口请见《ApiService 接口文档》的 2.1 章。

2 MQTT 协议

平台支持设备接入的唯一方式是 MQTT，同时遵循以下预定：

- 1、 mqtt server 的 ip 与端口依赖您安装 mqtt server 时的配置，一般端口默认为 1883。
- 2、 登录 mqtt server 的用户名为 device_id，密码为预先烧录或者注册获取到的密码。
- 3、 mqtt 的 clientId 必填，且以 `d-{{device_id}}` 这样的字符串组合传递到平台。例如 device_id： ebc59f15-b959-4cbf-87c0-1d7b56de214a，则 clientId 为 d- ebc59f15-b959-4cbf-87c0-1d7b56de214a。
- 4、 统一 mqtt payload 结构：

```
type MQTTPayload struct {
    Id      string `json:"id"`
    Version string `json:"version"`
}
```

```

Method string `json:"method"`
Params interface{} `json:"params"`
}

```

其中 method 字段定义如下:

场景	method	备注
消息推送	user.message.push	App 端消息推送
设备服务调用	thing.service.invoke.{{service name}}	其中,{{service name}}为用户自行约定的服务名。
设备属性设置	thing.property.set	用户设置属性
设备属性上报	thing.property.post	设备属性上报
设备事件上报	thing.event.post	设备事件上报

5、统一响应 payload

```

type MQTTResponse struct {
    Code    int    `json:"code"`
    Msg     string `json:"msg"`
    Id      string `json:"id"`
    Version string `json:"version"`
    Method  string `json:"method"`
    Data    interface{} `json:"data"`
}

```

6、文档中 topic 部分变量与通配符规则说明。

- 1、`{{product_key}}` 请用设备三元组中的产品 key 字符串代替。
- 2、`{{device_id}}` 请用设备的真实 device_id 代替。形如 524acc341-668541-ba9658-*** 的字符串。
- 3、`{{user_id}}` 为 pub 类型的 topic 变量, 请用订阅时的 topic 收到的用户 uuid 代

替。

4、+ 为 mqtt 中的通配符“+”，目前设备接口中用户通配用户的 uuid，在收到消息后，请通过 “/” 拆分 topic，即可获得用户的 uuid。

7、关于 QOS。一般情况下，pub/sub 的 qos 都请设置为 0。

2.1 事件上报

topic: xiots/thing/{{product_key}}/{{device_id}}/event/post

pub/sub: pub

示例 payload:

```
{
  "id": "1670663264",
  "version": "2.0",
  "method": "thing.event.post",
  "params": {"power_off": true}
}
```

2.2 事件上报响应

topic: xiots/thing/{{product_key}}/{{device_id}}/event/post_reply

pub/sub: sub

平台响应 payload:

```
{
  "id": "1670663264"
  "code": 200,
  "msg": "",
  "data": "5225113333",
  "version": "2.0",
  "method": "thing.event.post",
}
```

2.3 属性上报

topic: xiots/thing/{{product_key}}/{{device_id}}/property/post

pub/sub: pub

示例 payload:

```
{
  "id": "1670663264",
  "version": "2.0",
  "method": "thing.property.post",
  "params": {"power": "on"}
}
```

2.4 属性上报响应

topic: xiots/thing/{{product_key}}/{{device_id}}/property/post_reply

pub/sub: sub

平台响应 payload:

```
{
  "id": "1670663264"
  "code": 200,
  "msg": "",
  "data": "5225113333",
  "version": "2.0",
  "method": "thing.property.post",
}
```

2.5 属性设置

topic: xiots/thing/{{product_key}}/{{device_id}}/user+/property/set

pub/sub: sub

注意: 其中的 **+** 为 mqtt 的通配符, 不可写具体的值。

示例 payload:

```
{
  "id": "1670663264",
  "version": "2.0",
  "method": "thing.property.set",
  "params": {"power": "off"}
}
```

2.6 属性设置响应

topic: xiots/thing/{{product_key}}/{{device_id}}/user/{{user_id}}/property/set_reply

pub/sub: pub

平台响应 payload:

```
{
  "id": "1670663264"
  "code": 200,
  "msg": "ok",
  "data": "5225113333",
  "version": "2.0",
  "method": "thing.property.set",
}
```

2.7 服务调用

topic: xiots/thing/{{product_key}}/{{device_id}}/user+/service/invoke

pub/sub: sub

注意：其中的 **+** 为 mqtt 的通配符，不可写具体的值。

示例 payload:

```
{
  "id": "1670663264",
  "version": "2.0",
  "method": "thing.service.invoke.voice_set",
  "params": "{\"voice\": \"26.5\"}"
}
```

2.8 服务调用响应

topic: xiots/thing/{{product_key}}/{{device_id}}/user/{{user_id}}/service/invoke_reply

pub/sub: pub

平台响应 payload:

```
{
  "id": "1670663264"
  "code": 200,
  "msg": "ok",
  "data": "5225113333",
  "version": "2.0",
  "method": "thing.service.invoke.voice_set",
}
```