

App 接入指南

目录

1 接入方式.....	2
1.1 用户鉴权.....	2
2 MQTT 协议.....	2
2.1 设备属性设置.....	4
2.2 属性设置响应.....	4
2.3 设备服务调用.....	5
2.4 服务调用响应.....	5
2.5 消息通知.....	5

前言

本文档主要介绍用户端接入平台的基本步骤以及 MQTT 相关协议。平台根据用户名与密码进行鉴权，用户与平台的连接协议采用 MQTT，通过发布/订阅特定 Topic 的方式接收消息、设置设备属性、调用设备服务等。

1 接入方式

1.1 用户鉴权

平台采用用户名与密码的方式对用户进行鉴权，用户接入平台之前可以通过以下两种方式获取用户名与密码：

- 1、PMS 后台-用户管理-添加用户，由管理员手动添加用户与密码。
- 2、通过 ApiService 注册用户，可参考《ApiService 接口》文档 3.2 章节，提交用户名与密码注册用户，注册成功后即可登录 mqtt server。

2 MQTT 协议

平台支持用户长连接接入的唯一协议是 MQTT，同时遵循以下约定：

- 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、连接 mqtt server 之前需要调用 ApiService 的 **用户登录接口**，可参考《ApiService 接口》文档的 3.3 章节，传递注册时的用户名与密码，登录成功后，返回的 uuid 为 mqtt 的 ClientID。
- 3、mqtt 的 clientID 必填，且以 **u-{{uuid}}** 这样的字符串组合传递

到平台。例如 uuid: 75f11c22-71cf-49b1-8978-be3f3b2a7232, 则 clientID 为 u-75f11c22-71cf-49b1-8978-be3f3b2a7232。

4、 用户名与密码即为用户注册时用户名与密码。

5、 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

6、 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"`
}
```

7、 文档中 topic 部分变量与通配符规则说明

- 1、 `{{user_id}}` 表示用户登录接口返回的 uuid 字段
- 2、 `{{device_id}}` 为用户获取到绑定的设备列表后的 device_id， 为一个形如 5a4112442-ad85dvf-** 的字符串。
- 3、 `+` 为 mqtt 中的通配符“+”，目前用户的 mqtt 接口中通配设备的 device_id，在收到消息后， 请通过 “/” 拆分 topic， 即可获得设备的 device_id。

8、 关于 QOS。一般情况下， pub/sub 的 qos 都请设置为 0。

2.1 设备属性设置

topic: xiots/user/{{user_id}}/thing/{{device_id}}/property/set

pub/sub: pub

示例 payload:

```
{
  "id": "1670663264",
  "version": "2.0",
  "method": "thing.property.set",
  "params": "{\" temperature\": \"25.1\"}"
}
```

2.2 属性设置响应

topic: xiots/user/{{user_id}}/thing/+/property/set_reply

pub/sub: sub

注意：topic 中的 `+` 为 mqtt 中的通配符 + 号，此处用于匹配真实的设备 ID，客户端接收到订阅消息后，需要通过 “/” 拆分 topic，获取到此处的真实设备 ID。

平台响应 payload:

```
{
  "id": "1670663264"
  "code": 200,
  "msg": "",
  "data": "5225113333",
}
```

```
"version":"2.0",
"method":"thing.property.set",
}
```

2.3 设备服务调用

topic: xiots/user/{{user_id}}/thing/{{device_id}}/service/invoke

pub/sub: pub

示例 payload:

```
{
  "id":"1670663264",
  "version":"2.0",
  "method":"thing.service.invoke.poweroff",
  "params":{}
}
```

2.4 服务调用响应

topic: xiots/user/{{user_id}}/thing/+/service/invoke_reply

pub/sub: sub

注意: topic 中的 **+** 为 mqtt 中的通配符 + 号, 此处用于匹配真实的设备 ID, 客户端接收到订阅消息后, 需要通过 "/" 拆分 topic, 获取到此处的真实设备 ID。

平台响应 payload:

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

2.5 消息通知

topic: xiots/user/{{user_id}}/sys/message/notice

pub/sub: sub

示例 payload:

```
{
  "id":"1670663264",
  "version":"2.0",
  "method":"thing.message.push",
}
```

```
"params":{"device_id":"5588-84551acff-589331adc-***\"}
```