

开发文档-mqtt

开发文档-mqtt

mqtt协议

emqx

MQTTX

mqtt协议

MQTT（Message Queuing Telemetry Transport）是一种轻量级、开放式、基于发布/订阅模式的通信协议，特别适用于物联网（IoT）和移动设备等资源受限环境。特点如下

- 1. **发布/订阅模式**：MQTT采用发布/订阅模式，其中有两个主要角色：发布者（Publisher）和订阅者（Subscriber）。发布者负责发送消息，而订阅者则接收它们。
- 2. **主题（Topic）**：消息的发布和订阅通过主题进行，主题是消息的标识符。发布者将消息发布到特定的主题，而订阅者则通过订阅特定的主题来接收相关消息
- 3. **保持连接（Keep Alive）**：MQTT客户端和服务端之间的连接可以保持活动状态，以确保双方之间的有效通信。通过在连接过程中定义保持连接间隔，可以避免不必要的断开。
- 4. **遗嘱消息（Last Will and Testament）**：客户端可以指定一个遗嘱消息，该消息将在客户端异常断开连接时发布到预定义的主题。这有助于其他订阅者了解客户端的状态。

emqx

本项目中选用了成熟的mqtt服务器 emqx作为项目的服务器。使用开源版本的docker来运行后在 <http://localhost:18083/> 即可通过控制台操作服务器控制台。



物联网设备和后端都需要连接这个服务器来进行通讯。具体的实现细节请查看《接口文档》，在此不做赘述。

为了让后端了解服务器的流量情况，要开启客户端订阅系统主题的权限。

为了让后端了解服务器的连接情况，应该使用emqx提供的规则引擎，设置额如下规则

7mk70qm1

备注

SQL 编辑器 ⓘ
了解更多 SQL 语法，请参考 [SQL 语法与示例](#)。

1 SELECT

2 *

3 FROM

4 "\$events/client_connected",

5 "\$events/client_disconnected"

SQL 示例

事件

动作

将处理结果输出到数据桥接或内置函数中。

test

HTTP 服务

+ 添加动作

编辑动作

×

动作 *

使用数据桥接转发

数据桥接

webhook:test

+

名称 *

test

请求方法 *

POST

URL ② *

http://10.162.58.80:8000/mqtt/online

请求头

键	值	添加
content-type	application/json	删除

启用 TLS

☐

请求体 ②

取消

更新

这里用一个要特别注意的点是使用emqx的http服务的时候，url不能写成localhost或者127.0.0.1，否则完全不生效。应该填入当前设备在网络中的ipv4地址。这个点在网上的资料里强调的很少，花了我数小时来解决。

后续为了方便多平台移植，没有采用此方案，使用订阅系统主题的方式来实现此功能。

MQTTX

这里使用的客户端模拟器是配套的MQTTX，MQTTX有着很友好的交互体验。使用的时候只需要输入对应的client信息和服务器信息，点击连接即可工作。没有任何的上手难度。

