

# Autel Enterprise 设备内上云 Android MQTT 集成指南

## 功能概述

本 MQTT 客户端实现了以下核心功能:

- 自动连接和重连机制
- 设备信息管理(网关和无人机)
- 主题自动订阅
- 消息发布与接收
- JSON 消息解析
- 完整的生命周期管理

## 1. 添加依赖

在 app 模块的 `build.gradle` 文件中添加:

```
dependencies {  
    // MQTT 客户端库  
    implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.5'  
    // 协程支持  
    implementation 'org.jetbrains.kotlin:kotlinx-coroutines-android:1.6.4'  
}
```

## 2. 添加权限

在 `AndroidManifest.xml` 中添加:

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

## 3. 使用说明

### 3.1 初始化和连接

```
// 在 Activity 或 Fragment 中  
private val mqttHelper = MqttHelper()  
  
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    // 初始化 MQTT 客户端  
    mqttHelper.init()  
    // 开始连接(包含自动重连机制)  
    mqttHelper.connectWithRetry()  
}
```

## 3.2 发布消息

```
// 发布消息示例
val message = JSONObject().apply {
    put("method", "get_info")
    put("data", JSONObject().put("device_id", "123"))
}.toString()

mqttHelper.publish("device/command", message)
```

## 3.3 消息接收处理

MqttHelper 会自动处理以下类型的消息:

- 设备信息消息 (device/info)
- 网关相关消息
- 无人机相关消息

消息格式示例:

```
{
  "method": "gateway_info",
  "data": {
    "gateway_sn": "GW123456",
    "drone_sn": "DR789012"
  }
}
```

## 3.4 主题订阅

系统会自动订阅以下主题:

- 设备信息主题: `device/info`
- 网关主题:

```
thing/product/{gateway_sn}/osd
thing/product/{gateway_sn}/state
thing/product/{gateway_sn}/services_reply
thing/product/{gateway_sn}/events
thing/product/{gateway_sn}/requests
sys/product/{gateway_sn}/status
thing/product/{gateway_sn}/property/set_reply
thing/product/{gateway_sn}/get_token
```

- 无人机主题:

```
thing/product/{drone_sn}/osd
thing/product/{drone_sn}/state
```

## 3.5 资源释放

```
override fun onDestroy() {
    super.onDestroy()
    // 取消所有订阅
    mqttHelper.unsubscribeAll()
    // 关闭客户端
    mqttHelper.close()
}
```

## 4. 完整示例

### MainActivity.kt

```
class MainActivity : AppCompatActivity() {
    private val mqttHelper = MqttHelper()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // 初始化并连接
        mqttHelper.init()
        mqttHelper.connectWithRetry()

        // 发送消息示例
        binding.sendButton.setOnClickListener {
            val message = JSONObject().apply {
                put("method", "get_info")
                put("data", JSONObject().put("device_id", "123"))
            }.toString()

            mqttHelper.publish("device/command", message)
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        mqttHelper.unsubscribeAll()
        mqttHelper.close()
    }
}
```

## 5. 重要特性

1. 自动重连机制
  - 连接失败后使用指数退避算法重试
  - 最大重试间隔为 60 秒
2. 消息质量保证
  - 使用 QoS 1 确保消息至少送达一次
  - 自动处理连接丢失情况
3. 资源管理
  - 自动管理协程作用域
  - 完整的取消订阅流程

- 优雅的关闭处理
- 4. 错误处理
  - 完整的异常捕获和日志记录
  - 连接状态监控
  - 自动重连机制

## 6. 注意事项

---

1. 网络权限
  - 确保应用具有必要的网络权限
  - 检查网络连接状态
2. 内存管理
  - 在组件销毁时调用 close() 方法
  - 确保取消所有订阅
3. 消息处理
  - 注意消息格式的正确性
  - 处理 JSON 解析异常
4. 调试
  - 使用 logcat 查看详细日志
  - 标签为 "MQTT" 的日志记录了所有重要事件

## 7. 参考资料

---

- [Eclipse Paho Android Service](#)
- [MQTT Version 3.1.1](#)
- [Kotlin Coroutines](#)