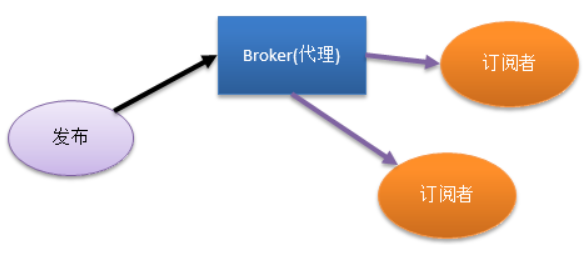
1. **什么是mqtt**

MQTT(消息队列遥测传输)基于发布/订阅范式的消息协议。它工作在 TCP/IP协议族上，是为硬件性能低下的远程设备以及网络状况糟糕的情况下而设计的发布/订阅型消息协议，为此，它需要一个消息中间件 。

**特点：**

1. 轻量级通信协议：实现mqtt协议的开销比较小，无需太多额外数据/文本
2. 基于TCP/IP协议
3. 基于“客户端-服务器”模式
4. 基于“发布/订阅”模式。解耦了发布者和订阅者之间的关系（还有“请求/回答”模式，典型的同步模式，例如http）
5. **入门笔记：**

很多的资料中介绍MQTT协议用于低端的产品比较适合。所以常常会看到他与物联网一起出现。模式图如下：



MQTT有要三种角色的存在。

**Broker**代理：中间件，用于处理信息并发送到相应的订阅者。

可以用mqtt.fx很好地模拟客户端

**发布者**：用于发布信息到代理上面。注意：发布者也可以是订阅者。

**订阅者**：就是用于接受信息的客户端。

通信时需要有”topic”，订阅者订阅主题，发布者发布的消息带有主题，broker则会将消息分发给每一个订阅主题的客户端

基于TCP/IP网络协议

在消息的传输上又做了三种模式的规定——分别为：最多一次、至少一次、只有一次。

1. **mqtt报文结构：**

报文由三部分组成：固定报头（2~5个字节）、可变报头（10字节）和有效载荷（包含以长度为前缀的字串）

1. **mqtt操作行为：**
   1. 状态存储：为了提供服务质量保障，客户端和服务器有必要存储会话状态。
   2. 网络连接：mqtt要求基础**传输层**能够提供有序的、可靠的、双向传输的字节流，所以选择了TCP/IP协议。
2. **注意：**

mqtt协议定义了大部分通信规则，但是有一部分通信规则没有规定，交由MQTT服务端定义，比如：用户名/密码格式、发布消息的有效载荷格式（json/纯文本......）、客户端支持的最大订阅数......

1. **小细节：**
   1. **大端、小端：**

**大端模式**，是指数据的高字节保存在内存的低地址中，而数据的低字节保存在内存的高地址中。

**小端模式**，是指数据的高字节保存在内存的高地址中，而数据的低字节保存在内存的低地址中

1. **mqtt 代理boker安装**

最简单的方法是找到mosquitto官网，下载.exe文件<https://mosquitto.org/download/> 运行安装，双击可执行文件，或者在服务中开启。

1. **mqtt客户端/服务器编码**

基本参考：<https://www.cnblogs.com/reject-ant/p/9669116.html>

**注意：回调对象需要手动创建(new)，自动注入可能会失效(这里出bug卡了很久)**

而new MemoryPersistence() 一般在 options.setCleanSession(true);（默认）的时候使用

connect：需要使用jar包中的mqttClient构造函数构造一个客户端，并调用connect函数，函数可能需要配置参数，所以可能需要配置MqttConnectOptions配置类

publish：需要构造MqttTopic和MqttMessage用来向指定主题publish信息。并取回token。

subscribe：直接调用订阅的函数即可

理解大端、小端，明白mqtt报文的结构，知道各个部分的大小。了解了各种类型控制报文（CONNACK、SUBSCRIBE......）的格式。清楚mqtt的操作行为。