**电子标签集成开发指导**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Change Description** | **Author** |
| V1.0 | 2018/7/1 | 初始版本 | Ning |
| V1.1 | 2018/7/13 | 1、更新4.5节Ack消息的MQTT主题描述不准确。  2、修改若干文字描述错误； | Ning |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**C O N F I D E N T I A L**

. **Catalogue**

[1. 目的 3](#_Toc519257221)

[2. 产品架构 3](#_Toc519257222)

[3. 功能介绍 4](#_Toc519257223)

[3.1 MQTT服务协议 4](#_Toc519257224)

[3.2 主要功能 4](#_Toc519257225)

[3.2.1 标签状态监测 4](#_Toc519257226)

[3.2.2 更新标签标签显示图片 5](#_Toc519257227)

[3.3 主要流程介绍 5](#_Toc519257228)

[3.3.1 网关上电 5](#_Toc519257229)

[3.3.2 标签状态监测流程 6](#_Toc519257230)

[3.3.3 更新图片流程 7](#_Toc519257231)

[3.3.4 更新标签连接密码 8](#_Toc519257232)

[4. 消息格式定义 8](#_Toc519257233)

[4.1 Alive消息 8](#_Toc519257234)

[4.2 标签状态上报消息 9](#_Toc519257235)

[4.3 更新图片消息 10](#_Toc519257236)

[4.4 修改密码 14](#_Toc519257237)

[4.5 应答消息 14](#_Toc519257238)

[5. 调测指导 15](#_Toc519257239)

[5.1 配置网关MQTT参数 15](#_Toc519257240)

[5.2 安装MQTT客户端 16](#_Toc519257241)

[5.2.1 mqtt.fx的安装 16](#_Toc519257242)

[5.2.2 mqtt.fx运行 16](#_Toc519257243)

[5.3 消息交互调测 18](#_Toc519257244)

[5.3.1 监测标签状态上报数据 18](#_Toc519257245)

[5.3.2 更新标签图片 20](#_Toc519257246)

[5.3.1 更新标签的密码 22](#_Toc519257247)

# 目的

本文描述了电子标签，以及网关对的API接口，用于指导第三方集成电子标签以及网关设备。

# 产品架构

电子标签

网关

电子标签

Bluetooth4.0/5.0

MQTT服务器

IP网络

WiFI/Eth

客户端管理

JSON

JSON

IP网络

电子标签

网关

电子标签

Bluetooth4.0/5.0

WiFI/Eth

接口：

* 电子标签和基站之间采用蓝牙4.0/5.0进行通讯，为了确保电子标签和基站不被非法接入，电子标签和基站之间通讯前需要采用双向MD5鉴权协议以确保安全。
* 基站上具备WiFI，以太网，以及蓝牙模块。基站和MQTT服务器之间采用MQTT协议，协议的报文采用JSON格式。
* 蓝牙网关和电子标签之间支持多对多冗余连接，当1个蓝牙网关宕机后，另外一个蓝牙网关可以自动承担宕机网关下的电子标签的管理功能。

**工作机制：**

1、电子标签：标签开机后，会周期性向网关发送标签状态信息，该消息包括：电子标签ID，正在显示的图片ID，电量，环境温度，故障状态，版本等信息。

2、网关：

* 网关开机后会启动电子标签扫描，并且会尝试连接MQTT服务器。一旦扫描到电子标签广播报文，则向MQTT服务器发布广播报文信息。另外，当网关收到MQTT服务器的命令请求后，则会向电子标签下发相应的控制命令或者数据。

3、MQTT服务器：采用标准的MQTT服务器架构，主要起一个数据订阅和发布的作用。

4、后台管理：根据用户的需求向MQTT服务器订阅相应的网关消息，或者发起命令请求。

# 功能介绍

## MQTT服务协议

电子标签系统软件架构的核心是基于MQTT，所有通讯报文采用JSON格式，这样可以很方便的采用第三方工具进行调测，大大降低第三方集成的难度。因此建议开发者在开发前可以先了解MQTT协议和JSON协议。

网关上电后会启动三个MQTT主题的订阅和发布，具体如下：

* Publish主题：名称默认为kbeacon/publish/mac，用于网关上报电子标签广播数据，用于后台监测标签当前工作状态；采用Qos0上报；
* Pubaction主题：名称默认为kbeacon/ pubaction /mac，用于网关向后台管理模块返回执行的命令结果；

订阅主题：

* Subaction主题： 名称默认为kbeacon/subaction/mac，用于接收后台管理模块发来的执行命令。

期中mac为网关的MAC地址，采用12位十六进制标识，如A1A200F3A300，用于唯一标识一个网关。

## 主要功能

### 标签状态监测

标签开机后，网关会周期性监测标签的状态。并且把如下数据周期性状态上报到MQTT服务器，MQTT服务器再把消息上报到客户端管理模块，以实现对标签的状态监测。

* 标签的ID
* 标签的信号强弱（到基站的距离）；
* 标签当前显示的图片ID；
* 标签运行的故障状态；
* 标签电池电量；
* 标签环境的温度；

说明：

其中电池的电量是采用电压的形式提供，单位是mV，比如3.3V对应3300mV，实际电量可以根据电压进行换算；

标签传感器监测的环境温度（误差范围为±3℃），可以提供一个粗略的温度监测；

### 更新标签标签显示图片

网关开机后，就会连接到MQTT服务器，并且会监听MQTT服务器发来的消息；当客户端管理模块需要更新指定标签的图片时，可以通过MQTT服务器将图片下发到网关。网关进而会和标签建立连接，并且将图片信息下发到标签，并向MQTT服务器返回执行结果，MQTT服务器也会向客户端管理软件返回结果。

## 主要流程介绍

### 网关上电

MQTT服务器

网关1

客户端管理

发布(Publish主题)

(alive消息)

网关上电

订阅(Publish主题)

发布(alive消息)

可选

发布(Publish主题)

(alive消息)

网关上电后，会周期性的向MQTT服务器发送Alive消息，期中alive消息包括网关的ID（MAC地址），pubaction主题名称，以及subaction主题名称。

本消息的发送周期为45~60秒内发送1次，周期性发送。

客户端管理可以订阅本消息，以获得网关的pubaction主题，subaction主题，便于后续命令的执行。

### 标签状态监测流程

订阅(Publish主题)

可选

标签1

MQTT服务器

网关1

客户端管理

发布(主题:Publish)

(标签状态数据)

标签1状态上报

标签1状态上报

发布(主题:Publish)

(标签状态数据)

发布(主题:Publish)

(标签状态数据)

客户端管理软件通过订阅Publish主题，可以得知当前的标签位于哪一个网关的信号覆盖范围下面。并且可以建立标签和网关的对应关系。

此外，同一个标签的状态信息会被多个网关上报状态数据，并且会上报标签的信号强度信息。因此客户端管理软件可以建立一种“标签-网关”一对多的关系表，当客户端管理软件在向标签下发图片数据时，可以通过信号最强的网关（距离标签最近）向标签下发数据。此外，这么做的好处是，当某一个网关宕机时，客户端管理软件可以通过候选的网关向标签下发数据。

标签和网关的关系采用自组织冗余结构，无需人工配置。这样可以极大简化配置过程。比如当一个网关宕机时，标签会自动选择另外一个覆盖信号范围内的网关上报状态数据。

### 更新图片流程

标签

MQTT服务器

网关1

客户端管理

发布(主题：subaction)

(图片更新：标签ID,密码，图片)

发布(标签ID,密码，图片)

发布（主题:Pubaction）

(success1)

鉴权通过

下发图片

发布（主题:Pubaction）

(success, ack:1)

发布（主题:Pubaction）

(success0)

发布（主题:Pubaction）

(success, ack:0)

1、客户端管理根据先前标签和网关对应关系，选择一个网关发布图片更新消息，并且将图片信息转换成字符编码在JSON消息中携带，然后发布到网关的subaction主题。

2、MQTT服务器收到消息后，将更新请求发布到对应的网关。

3、网关收到请求后根据对应的请求，向MQTT服务器发送消息递交成功的应答消息，应答的Ack Cause为1。（注意本消息表示命令已经到达网关，但还没有开始下发到标签）

4、网关根据请求消息中携带的标签ID，扫描对应的标签并且建立连接，连接建立后采用双向MD5 Hash算法+随机码方式对密码进行验证（密码不会在空中传递，因此不会第三方蓝牙工具嗅探到），每次验证都会采用新的随机码。

* 双向MD5鉴权算法是指设备会验证网关的合法性，同时网关也会验证设备的合法性。

5、鉴权通过后，网关下发图片数据，标签更新图片数据成功后，网关将向MQTT服务器发送成功应答消息，应答的Ack Cause为0。

更新过程中如果有任何失败，网关将向MQTT服务器发送更新失败应答，并且携带对应的错误码（Ack Cause）

说明：命令执行过程中，网关会向MQTT服务器发送两次应答消息，第一次应答表明收到消息，消息进入缓存队列。第2次应答表示命令执行完成。

### 更新标签连接密码

标签

MQTT服务器

网关1

客户端管理

发布(主题：subaction)

(密码更新：标签ID,旧密码，新密码)

发布(密码更新)

发布（主题:Pubaction）

(success1)

旧密码鉴权通过

更新密码

发布（主题:Pubaction）

(success, ack:1)

发布（主题:Pubaction）

(success0)

发布（主题:Pubaction）

(success, ack:0)

标签的默认密码为8个ASCII的0，即00000000，客户端管理软件可以通过本命令更新标签的连接密码，更新完设备连接密码后，务必记住密码，密码一旦丢失，无法找回。

## 标签ID地址定义规则

按照蓝牙协议的规则，标签的设备ID地址是根据IEEE定义的MAC地址按照字节反序得到，其中MAC地址前面两位表示厂商ID，目前所有出厂的标签前面两位都是DD33。印刷在标签标明的二维码是MAC地址。

如下图所示，如下标签的二维码为DD330A000986。**标签ID需要反序，**它对应的标签ID为8609000A33DD，您在下发各种JSON消息种用到的标签ID就是8609000A33DD。



# 消息格式定义

## Alive消息

方向：网关=>MQTT服务器

MQTT主题：publish

消息名称: alive

内容举例：

{

"msg": "alive",

"gmac": "A1A2A3A4A5A6",

"subaction": "kbeacon/subaction/A1A2A3A4A5A6",

"pubaction": "kbeacon/pubaction/A1A2A3A4A5A6"

}

其中gmac为网关的mac地址；

## 标签状态上报消息

方向：网关=>MQTT服务器

MQTT主题：publish

消息名称:advData

*注意：状态上报消息中的标签ID，跟标签外壳上二维码MAC地址并不相同，而是二维码MAC地址的反序，两者的对应关系参见《3.4 标签ID地址定义规则》。*

内容举例：

{

"msg": "advData",

"gmac": "66E800E29647",

"obj": [

{

"dmac": "795792F98100",

"rssi": "-71",

"data1": "0201060302A0FE10FF4B4D100000000B371E0001020304CA"

},

{

"dmac": "265892F98100",

"rssi": "-40",

"data1": "0201060302A0FE10FF4B4D100000000C83210001020304CA"

}

],

"seq": 70

}

本消息可以通过数组方式上报多个设备的状态信息，其中字段定义如下

1）gmac：网关的mac地址；

2）每一个设备状态信息包含如下内容

dmac：标签的ID；

rssi：网关接收到的标签的信号强度，该数值越小，信号强度越弱；

data1的字段含义定义如下：

"data1": "0201060302A0FE10FF4B4D100000000C83210001020304CA"

* 0201060302：保留
* A0FE：表示Beacon类型服务数据，所有的标签本字段固定为A0FE，如果不是A0FE，则可以丢弃本消息；
* 10：后续消息字段长度，即16字节
* FF4B4D：保留
* 10：传感器类型，10表示电子标签，如果不是10，则可以丢弃本设备上报消息；
* 00：标签的固件版本号，不同版本号的标签支持的能里会有差异。
* 0000：标签状态和能力字段，共2Byte，具体定义如下：

第1字节

|  |  |  |
| --- | --- | --- |
| 7~5bit | 4bit | 3~0bit |
| Reserved | 屏幕状态:  0:normal  1:lcd fault | Reserved |

第2字节

|  |  |  |
| --- | --- | --- |
| 7~5bit | 4bit | 3~0bit |
| Reserved | LCD点阵类型：  0: 从上往下，从右往左  1：从上往下，从左往右 | 标签类型：  0x0：2.9寸双色  0x1：2.9寸黑红白  0x2：4.2寸双色 |

* 0C83：电池电压信息，采用大端编码，（0C83电压为3203mV）
* 2100：环境温度，有符号数表示，大端编码，（2100表示33℃）
* 01020304：当前显示的图片ID，更新图片时由客户端管理软件下发指定，建议每次更新图片都分配一个唯一的图片，这样可以监控各个标签当前显示的图片。
* CA：标签在1米处的信号功率，采用符号数，（0xCA=-54dBm）用于计算标签和基站之间的距离；

Seq：为每次上报消息的序号，每次递增1

固件版本能里说明：

|  |  |
| --- | --- |
| 固件版本 | 支持的能力 |
| 0x00 | -支持2.9寸黑白标签  -支持更新密码  -支持修改广播周期 |
| 0x01 | -在0x01基础上新增支持图片压缩算法，刷新速度会更快 |

## 更新图片消息

### 非压缩方式更新图片

方向：MQTT服务器=>网关

MQTT主题：subaction

消息名称:dData

内容举例：

{

"msg":"dData",

"mac":"375692f98100",

"seq":4,

"auth1":"00000000",

"dType": "hex",

"data":"01010203041280FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFFFFFFFFFFF80FFFF80FFFFFFFFFFFFFFE7FFFFFFFF80FFFF80FFFFFFFFFFFFFF079FFFFFFF80FFFF80FFFFFFFFFFFFFF001FFFFFFF80FFFF80FFFFFFFFFFFFFF001FFFFE07FFF03FFFFFFFFFFFFFFFFFC01FFFFE07FFF03FFFFFFFFFFFFFFFFFE7FFFFFE07FFF03FFFFFFFFFFFFFFFFFFFFFFFFE07FFF03FFFFFFFFFFFFFFFFFE63FFFFE07FFF03FFFFFFFFFFFFFFFFFE43FFFFE07FFF03FFFFFFFFFFFFFFFFFC41FFFFE07FFF03F80FFFFFFFFFFFFFFC89FFFFE07FFF03F80FFFFFFFFFFFFFFC89FFFFE07FFF03F80FFFFFFFFFFFFFFC09FFFFE07FFF03F80FFFFFFFFFFFFFFE11FFFFE07FFF03F80FFFFFFFFFFFFFFF3BFFFFE07FFF03F80FFFFFFFFFFFFFFFFFFFFFE07FFF03F80FFFFFFFFFFFFFFF9FFFFFE07FFF03F80FFFFFFFFFFFFFFE17FFFFE07FFF03F80FFFFFFFFFFFFFFE13FFFFE03FFF03F80FFFFFFFFFFFFFFC11FFFFE0000003F80FFFFFFFFFFFFFFCD9FFFFE0000003F80FFFFFFFFFFFFFFC19FFFFE0000003F80FFFFFFFFFFFFFFE01FFFFE0000003F80FFFFFFFFFFFFFFF03FFFFE0000003F80FFFFFFFFFFFFFFF87FFFFE0000003F80FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF80FFFFFFFFFFFFFFF1FFFFFFFFFFFFFF80FFFFFFFFFFFFFFE07FFFFFFFFFFFFF80FFFFFFFFFFFFFFC03FFFFFFFFFFFFF80FFFFFFFFFFFFFFC41FFFFFFFFFFFFF80FFFFFFFFFFFFFFEF9FFFFFFFFFFFFF80FFFFFFFFFFFFFF079FFFFFFFFFFFFF80FFFFFFFFFFFFFF003FFFFFFFFFFFFF80FFFFFFFFFFFFFF003FFFFFFF80000000FFFFFFFFFFFFFFF81FFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFF80000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE17FFFFFFFFFFFFFFFFFFFFFFFFFFFFFE13FFFFFFFFFFFFFFFFFFFFFFFFFFFFFC11FFFFFFFFFFFFFFFFFFFFFFFFFFFFFCD9FFFFFFFFFFFFFFFFFFFFFFFFFFFFFC19FFFFFFFFFFFFFFFFFFFFFFFFFFFFFE01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF03FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF87FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE3FFFFFE1FFFFFFFFFFFFFFFFFFFFFFFC01FFFFE01FFFFFFFFFFFFFFFFFFFFFFC01FFFFE003FFFFFFFFFFFFFFFFFFFFFC61FFFFE0003FFFFFFFFFFFFFFFFFFFFE7FFFFFE00007FFFFFFFFFFFFFFFFFFF03FFFFFF000007FFFFFFFFFFFFFFFFFF001FFFFFFF00000001FFFFFFFFFFFFFF801FFFFFFFFC000003FFFFFFFFFFFFFFFE1FFFFFFFFE000003FFFFFFFFFFFFFFFFFFFFFFFFC0000003FFFFFFFFFFFFFFE7FFFFFFF000000003FFFFFFFFFFFFFF079FFFFE000007FFFFFFFFFFFFFFFFFF001FFFFE0000FFFFFFFFFFFFFFFFFFFF001FFFFE0007FFFFFFFFFFFFFFFFFFFFC03FFFFE007FFFFFFFFFFFFFFFFFFFFFF7FFFFFE07FFFFFFFFFFFFFFFFFFFFFFFFFFFFFE7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE3FFFFFFFFFFFFFFFFFFFFFFFFFFFF0003FFFFFFFFFFFFFFFFFFFFFFFFFF000003FFFFFFFFFFFFFFFFFFFFFFFF00000003FFFFFFFFFFFFFFFFFFFFFF0000000003FFFFFFFFFFFFFFE07FFFFE0000000003FFFFFFFFFFFFFFE01FFFFE000000083FFFFFFFFFFFFFFFE01FFFFE0001FFF87FFFFFFFFFFFFFFFFF3FFFFE07FFFFF87FFFFFFFFFFFFFFFEF9FFFFE007FFFF87FFFFFFFFFFFFFFFE01FFFFE000000F87FFFFFFFFFFFFFFFE01FFFFE0000000007FFFFFFFFFFFFFFF81FFFFE0000000003FFFFFFFFFFFFFFFFFFFFFFF800000003FFFFFFFFFFFFFFFFFFFFFFFFFC000003FFFFFFFFFFFFFFF07FFFFFFFFFFC0003FFFFFFFFFFFFFFE03FFFFFFFFFFFFC01FFFFFFFFFFFFFFC01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFCF1FFFFFFFFFFFFFFFFFFFFFFFFFFFFFC79FFFFFFFFFFFFFFFFFFFFFFFFFFFFFE01FFFFFFFFFFFFFC1FFFFFFFFFFFFFFE03FFFFFFFFFFFFFC3FFFFFFFFFFFFFFF03FFFFFFFFFFFFFC3FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC3FFFFFFFFFFFFFFEFFFFFFFFFFFFFFFC3FFFFFFFFFFFFFFE3FFFFFFFFFFFFFFC3FFFFFFFFFFFFFFF0FFFFFFFFFFFFFFC3FFFFFFFFFFFFFFFC3FFFFE0000000003FFFFFFFFFFFFFFF00FFFFE0000000003FFFFFFFFFFFFFFE007FFFE0000000003FFFFFFFFFFFFFFE0E3FFFE0000000003FFFFFFFFFFFFFFEFF3FFFE0000000003FFFFFFFFFFFFFFFFF7FFFE0000000003FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC000FFFFFFFFFFFFFFFFFFFFC7FFFFFF80007FFFFFFFFFFFFFFFFFFFE7FFFFFF00003FFFFFFFFFFFFFFFFFFFE3FFFFFF00001FFFFFFFFFFFFFFFFFFFE07FFFFF00001FFFFFFFFFFFFFFFFFFFE01FFFFE01601FFFFFFFFFFFFFFFFFFFF01FFFFE0FFC1FFFFFFFFFFFFFFFFFFFFFDFFFFE0FFE1FFFFFFFFFFFFFFFFFFFFFFFFFFE0FFE1FFFFFFFFFFFFFFFFFFFF1FFFFFE0000010003FFFFFFFFFFFFFFE13FFFFE0000000003FFFFFFFFFFFFFFC13FFFFE0000000003FFFFFFFFFFFFFFCD9FFFFE0000000003FFFFFFFFFFFFFFC19FFFFE0000000003FFFFFFFFFFFFFFE01FFFFE0000000001FFFFFFFFFFFFFFE01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF03FFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE6FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE6FFFFFFFF81FFF807FFFFFFFFFFFFFFE07FFFFFF800FFC000FFFFFFFFFFFFFFF003FFFFF000FF80007FFFFFFFFFFFFFF001FFFFF000FE00003FFFFFFFFFFFFFFE01FFFFE000FC00003FFFFFFFFFFFFFE6FFFFFFE001F800001FFFFFFFFFFFFFE6FFFFFFE0FFF007FC1FFFFFFFFFFFFFE0FFFFFFE1FFE00FFC1FFFFFFFFFFFFFE03FFFFFE1FF801FFE1FFFFFFFFFFFFFF001FFFFE0FF007FFC1FFFFFFFFFFFFFF801FFFFE00000FFF81FFFFFFFFFFFFFFE61FFFFE00001FC001FFFFFFFFFFFFFFFFFFFFFF00003FC003FFFFFFFFFFFFFFFFFFFFFF8000FFC003FFFFFFFFFFFFFF80FFFFFF8001FFC007FFFFFFFFFFFFFF007FFFFFE007FFC01FFFFFFFFFFFFFFF003FFFFFFFFFFFFFFFFFFFFFFFFFFFFE3E1FFFFFFFFFFFFFFFFFFFFFFFFFFFFE3F1FFFFFFFFFFFFFFFFFFFFFFFFFFFFE3F1FFFFFFFFFFFFFFFFFFFFFFFFFFFFF021FFFFE0000000001FFFFFFFFFFFFFF003FFFFE0000000003FFFFFFFFFFFFFF803FFFFE0000000003FFFFFFFFFFFFFFE0FFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFF0000000003FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC00000001FFFFFFFFFFFFFFFFFFFFFFF8000000007FFFFFFFFFFFFFFFFFFFFFF0000000007FFFFFFFFFFFFFFFFFFFFFF0000000003FFFFFFFFFFFFFFFFFFFFFF0000000003FFFFFFFFFFFFFFFFFFFFFF07FFFFFF03FFFFFFFFFFFFFFFFFFFFFE0FFFFFFF83FFFFFFFFFFFFFFFFFFFFFE0FFFFFFFC3FFFFFFFFFFFFFFFFFFFFFE0FFFFFFFC3FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFE0000000003FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF00003FFFFFFFE000001FFFFFFFF007FF00007FFFFFFFF800001FFFFFFFF007FF0000FFFFFFFFFC00001FFFFFFFF007FF0001FFFFFFFFFC00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001F80000007E00001FFFFFFFF007FF0001F00000007E00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF000007FFFFFF8000001FFFFFFFF007FF00007FFFFFFFF000001FFFFFFFF007FF0000FFFFFFFFF800001FFFFFFFF007FF0001FFFFFFFFFC00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0003FFFFFFFFFE00001FFFFFFFF007FF0003F00000007F00001FFFFFFFF007FF0003F00000003F00001FFFFFFFF007FF0003F80000007F00001FFFFFFFF007FF0003FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0000FFFFFFFFFC00001FFFFFFFF007FF00007FFFFFFFF800001FFFFFFFF007FF00003FFFFFFFF000001FFFFFFFF007FF000007FFFFFF0000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF00000FFFFFFF8000001FFFFFFFF007FF00003FFFFFFFF000001FFFFFFFF007FF0000FFFFFFFFF800001FFFFFFFF007FF0000FFFFFFFFFC00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0003FFFFFFFFFE00001FFFFFFFF007FF0003F00000007F00001FFFFFFFF007FF0003F00000003F00001FFFFFFFF007FF0003FA000000FF00001FFFFFFFF007FF0003FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0000FFFFFFFFFC00001FFFFFFFF007FF0000FFFFFFFFF800001FFFFFFFF007FF00003FFFFFFFF000001FFFFFFFF007FF000007FFFFFF0000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF00000FE07FFFC000001FFFFFFFF007FF00007FE07FFFF000001FFFFFFFF007FF0000FFE07FFFFC00001FFFFFFFF007FF0001FFE07FFFFC00001FFFFFFFF007FF0001FFE07FFFFE00001FFFFFFFF007FF0003FFE07FFFFE00001FFFFFFFF007FF0003FFE07FFFFE00001FFFFFFFF007FF0003F0007E007F00001FFFFFFFF007FF0003F0007E003F00001FFFFFFFF007FF0003F00000007F00001FFFFFFFF007FF0003FFFFFFFFFE00001FFFFFFFF007FF0003FFFFFFFFFE00001FFFFFFFF007FF0001FFFFFFFFFE00001FFFFFFFF007FF0000FFFFFFFFFC00001FFFFFFFF007FF0000FFFFFFFFF800001FFFFFFFF007FF00003FFFFFFFF000001FFFFFFFF007FF00000FFFFFFF8000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFF007FF0000000000000000001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"}

其中：

mac：标签设备ID；

seq：消息序号，客户端管理软件每次发送新的消息，序号需要加1处理；

auth1：标签设备密码，出厂默认密码为8个0

dType: 数据格式类型,hex表示图片的数据内容为十六进制

data：图片内容，具体字段定义如下：

举例："data":"01010203041280FFFF………..

* 01：图片更新类型，0x01表示采用非压缩十六进制编码图片；
* 01020304：图片ID，更新成功后，标签的状态上报消息会携带最新的图片ID，您可以通过这个图片ID跟业务建立对应关系。 比如1245对应价格为14.7元，当用于会议系统时，可以用图片ID表示人员ID。
* 1280：图片长度，采用小端编码十六进制，即4736，对应2.9寸的标签；
* 接下来的内容及图片的编码内容；图片的编码方式参考文档《电子标签图片编码格式.docx》

### 压缩方式更新图片

方向：MQTT服务器=>网关

MQTT主题：subaction

消息名称:dData

内容举例：

{

"msg": "dData",

"mac": "4631000A33DD",

"seq": 19871,

"auth1": "00000000",

"dType": "ascii",

"data": ""

}

*注意：版本号大于1的标签才支持压缩方式更新图片。*

其中：

mac：标签设备ID；

seq：消息序号，客户端管理软件每次发送新的消息，序号需要加1处理；

auth1：标签设备密码，出厂默认密码为8个0

dType: 数据格式类型,ascii表示图片的数据内容为基于ASCII的压缩算法。

data：图片内容，具体字段定义如下：

举例："data":"01010203041280FFFF………..

* 02：图片更新类型，0x02表示采用压缩的ASCII算法更新图片；
* 00000091：图片ID，更新成功后，标签的状态上报消息会携带最新的图片ID，您可以通过这个图片ID跟业务建立对应关系。 比如1245对应价格为14.7元，当用于会议系统时，可以用图片ID表示人员ID。
* 1280：图片长度，采用小端编码十六进制，即4736，对应2.9寸的标签；
* 0A：压缩Code的个数，0A表示10个。
* FFFE3F1FC7000F0787C0：压缩的Code
* 接下来的内容是压缩图片内容；图片的编码方式参考文档《电子标签图片编码格式.docx》

图片压缩算法原理如下：

1、统计图片中出现频率很高的字符，采用1个字符替代。 比如0xFF出现频率很高。 则采用’G’替代，按照顺序采用’G’~’Z’字符替代，最多20个字符。

2、采用小写a~z表示字符个数(1~)。 比如0xFFFFFF，压缩编码为cG.

压缩的demo参见：https://github.com/kkmhogen/ESLBin2Json.git

## 修改密码

方向：MQTT服务器=>网关

MQTT主题：subaction

消息名称:dData

内容举例：

{

"msg": "dData",

"mac": "625692f98100",

"seq": 12,

"data": {

"tag": "mdf",

"pwd":"12345678",

},

"dType": "json",

"auth1": "00000000"

}

mac：标签设备ID；

seq：消息序号，客户端管理软件每次发送新的消息，序号需要加1处理；

auth1：当前标签设备密码；

dType: data字段的数据类型

data：字段

tag：mdf标识修改命令

pwd：表示修改后新的设备密码，即12345678

## 应答消息

方向：网关=>MQTT服务器

MQTT主题：pubaction

消息名称:dAck

内容举例：

{

"msg": "dAck",

"mac": "625692F98100",

"rslt": "succ",

"seq": 8,

"cause": 0

}

说明：

mac：标签设备ID；

seq：对应的请求命令的序号；

rslt：succ表示成功，fail表示失败；

cause：错误码表示

错误码对应的含义如下：

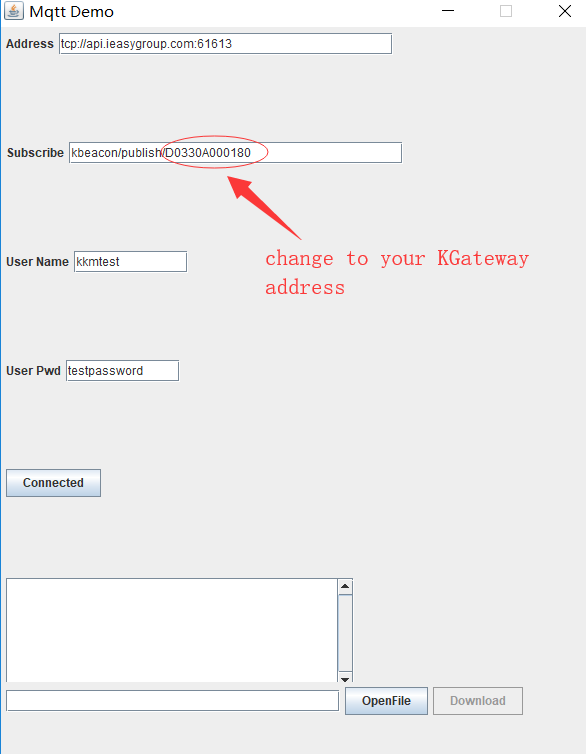
|  |  |  |
| --- | --- | --- |
| rslt字段 | 值(十进制) | 含义 |
| succ | 1 | 收到命令，正在执行 |
| 0 | 命令执行完成 |
| 非法下载的数据内容(对应data字段内容) | 65 | 等待执行消息条数超过网关支持最大能力 |
| 129 | 设备搜索失败 |
| 130 | 网关不支持该设备 |
| 131 | 设备连接失败 |
| 133 | 其它内部错误 |
| 257 | 密码错误 |
| 258 | 鉴权算法不支持 |
| 259 |  |
| 289 | 请求刷新图片尺寸和实际墨水屏尺寸定义不符合 |
| 290 | 接收到的数据长度跟图片头定义不符合 |
| 291 | 墨水屏故障 |
| 298 | 未知的消息内容 |

# Demo资源

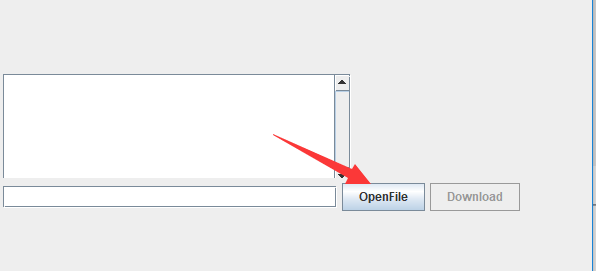
## MQTT客户端demo

我们开发了一个基于Java的MQTT客户端demo，您可以通过该demo连接到MQTT服务器，并向标签下发数据。该Demo基于JDK1.7开发.

下载地址: https://github.com/kkmhogen/KGatewayClientDemo.git



打开JSON命令文件，就可以下发。 对于JSON文件，你需要修改对应的标签ID。JSON文件可以通过另外一个工具生成。



## BIN图片转换JSON客户端demo

我们开发了一个可以将BIN格式的图片转换为MQTT的JSON消息的Demo，基于C语言和 Visual Studio2010开发。

链接地址：<https://github.com/kkmhogen/ESLBin2Json.git>

标签图片制作参考文档《电子标签图片制作.docx》

# 调测指导

## 配置网关MQTT参数

网关出厂的时候默认参数是连接到如下测试用的MQTT服务器。

地址为：tcp://api.ieasygroup.com:61613

测试用户名：kkmtest

密码：testpassword

开发阶段可以使用上述地址进行测试和联调。

网关的其它参数配置参见《电子标签网关介绍和配置.docx》

## 安装MQTT客户端

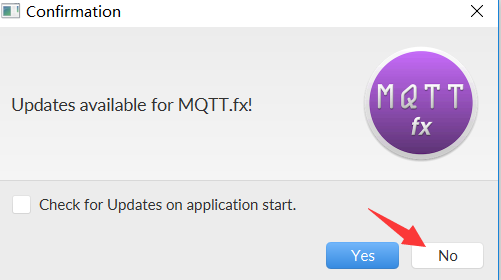
目前，第三方MQTT客户端很多，推荐使用Mqtt.fx，该客户端支持windows，linux，mac。

### mqtt.fx的安装

MQTT.fx 是目前主流的mqtt客户端，可以快速验证是否可以与IoT Hub 服务交流发布或订阅消息。MQTT代理服务器将接收到的主题topic发送给给所有订阅的客户端。

下载链接：   
<http://www.jensd.de/apps/mqttfx/1.5.0/>

### mqtt.fx运行

1、打开软件，打开时会提示有更新，直接点击No，最好别点击yes，会报错，然后用不了。   
 

2、打开界面后，进入设置界面，

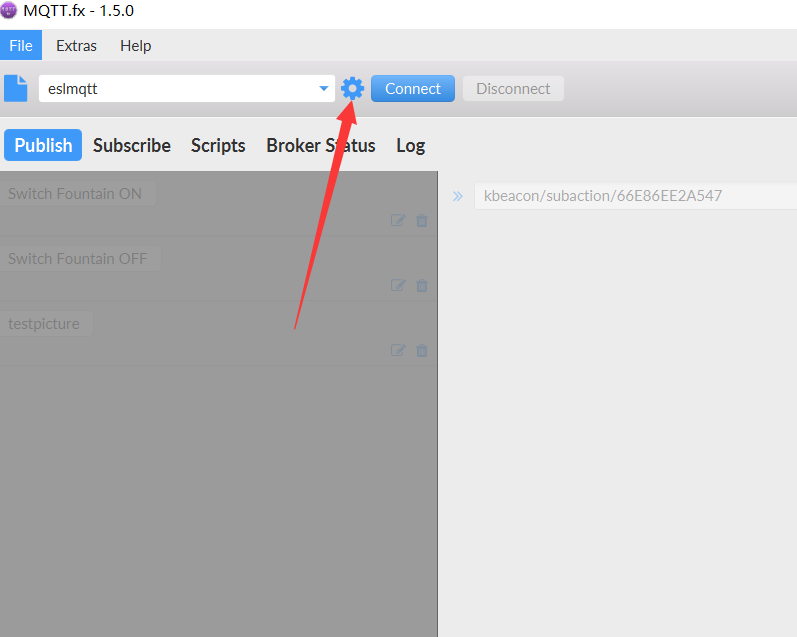
输入MQTT服务器的地址，以及用户名，IP地址，点击确认。

测试MQTT服务器信息如下：

* Broker Address：api.ieasygroup.com
* Broker Port: 61613
* Client ID: 选择随机产生，点击Generate
* 其它参数参考如下截图。

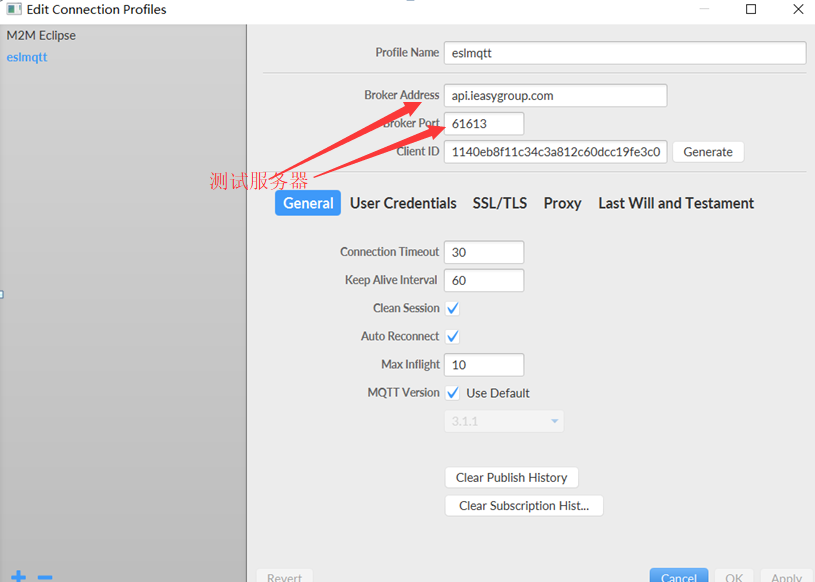
如上服务器仅仅用于开发阶段测试使用。如果是商用，建议安装自己的MQTT服务器。

第一步：单击设置参数按钮



第二步：设置连接参数，采用默认测试服务器

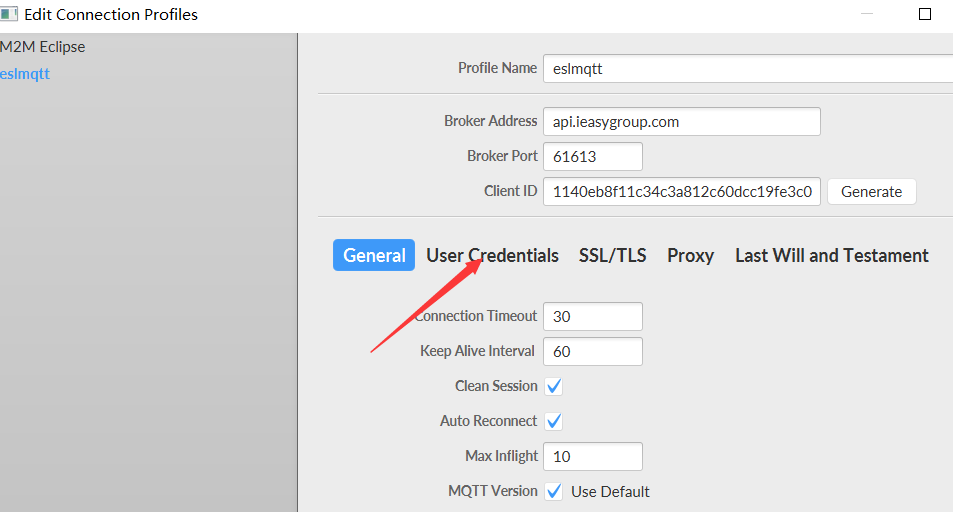
* 默认的测试服务器是：api.ieasygroup.com
* 默认端口为61613，
* Client ID可以点解Generate随机产生。
* 其它参数保持默认。

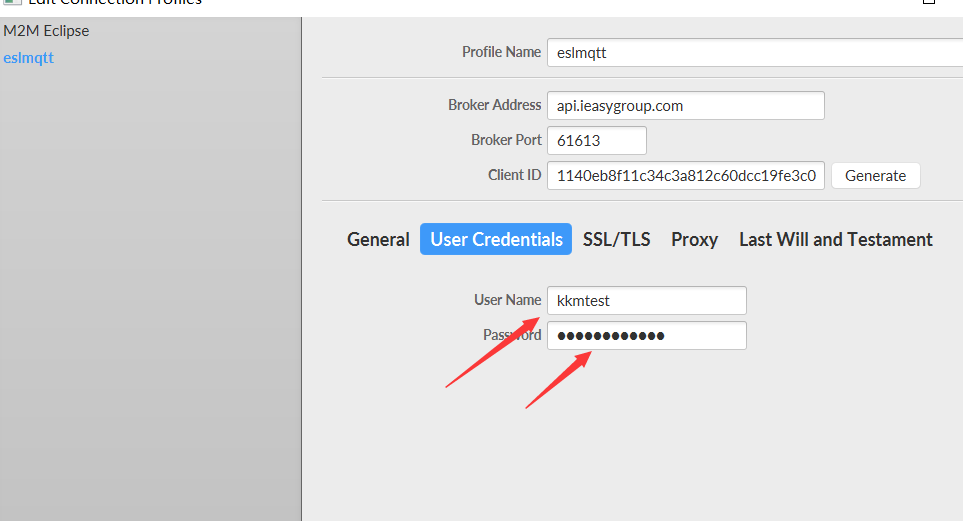


第三步骤：设置用户名和密码

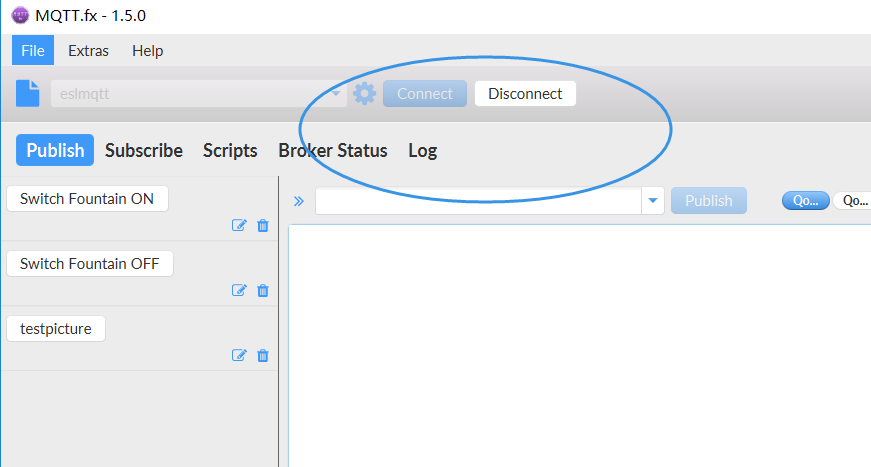
* 点击“User Credentials”切换到用户名密码输入界面。
* 输入用户名和密码。

测试服务器默认用户名为：kkmtest，默认密码为：testpassword





3、设置完后，点击connect，连接成功后，connect按钮会变灰色。

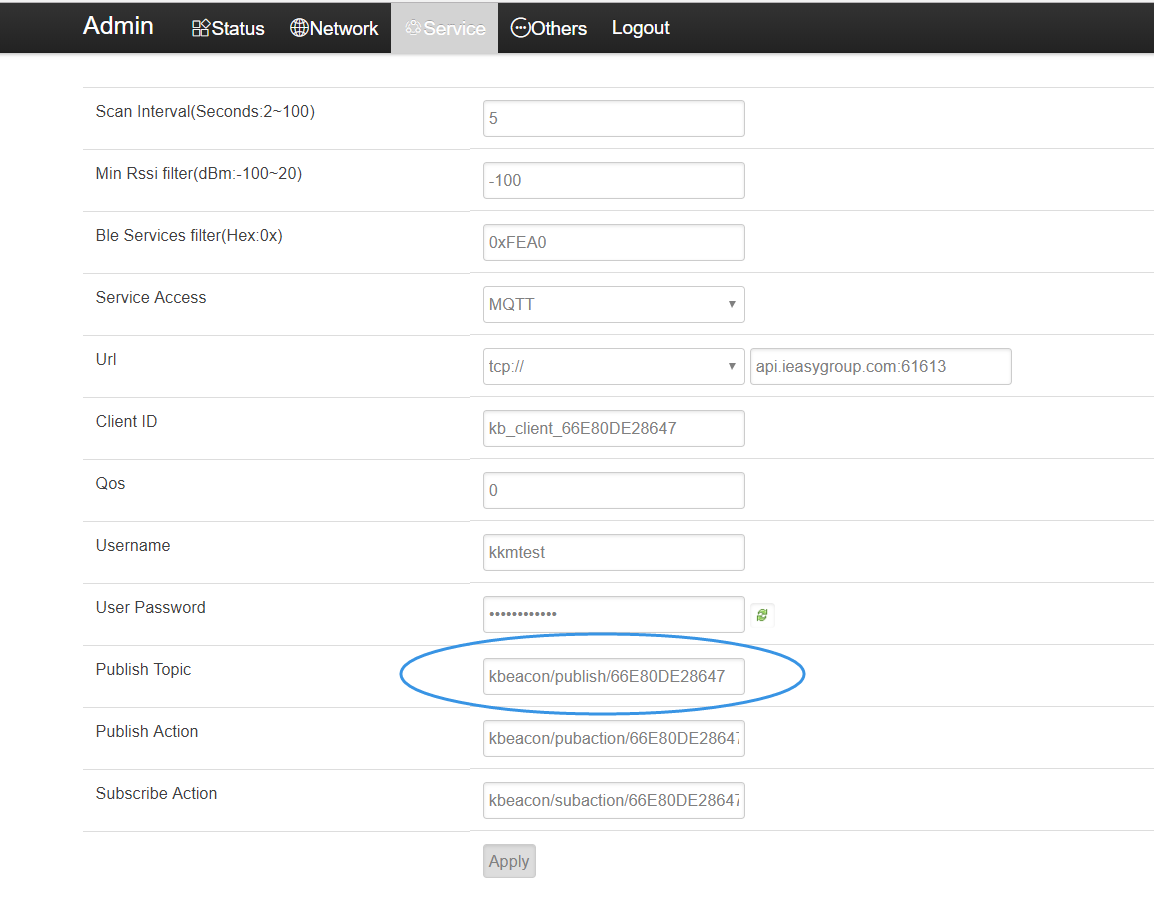


## 消息交互调测

### 监测标签状态上报数据

1）获取网关publish主题名称。

* 方式1：如果您没有登录到网关修改过mqtt publish主题名称，则主题名称默认为：kbeacon/publish/mac地址，例如kbeacon/publish/66E80DE28647
* 方法2：如果您修改过publish主题名称，则可以通过chrome浏览器登录到网关，登录的方法参见《电子标签网关介绍和配置.docx》，在Services页面可以看到publish主题名称。

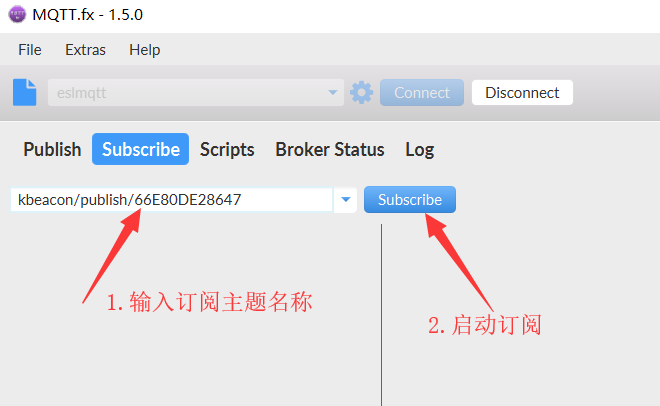


2）订阅网关的标签状态监测发布主题，在MQTT.fx中切换到Subscribe页面。

输入主题名称为：kbeacon/publish/66E80DE28647；

*注意：订阅主题中的ID(66E80DE28647)是网关的设备ID，不是标签的设备ID。*

再点击Subscribe则可以启动订阅。



订阅主题支持通配符操作，比如订阅：kbeacon/publish/# ，则可以订阅所有的基站上报消息。

3）观察标签状态数据上报：

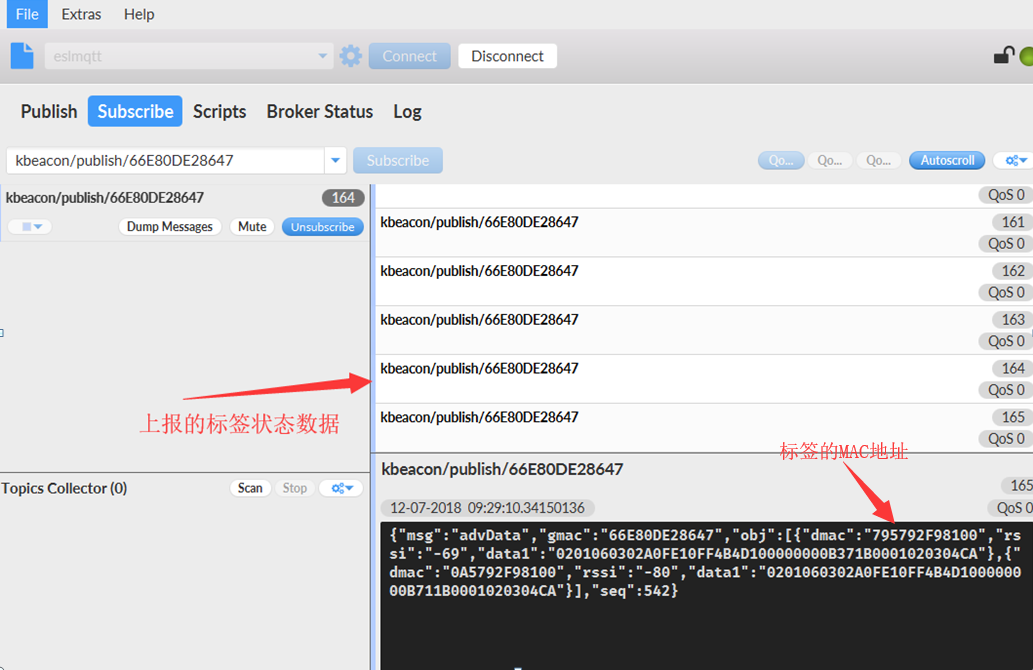
如果网关旁边有标签，则等待10来秒钟后，则会看到上报的数据。

从下图可以看出，网关监测到2个电子标签的状态上报，标签的ID（dmac地址）分别为：

* 标签1：795792F98100，rssi信号强度为-60dBm
* 标签2：0A5792F98100，rssi信号强度为-66dBm，

*注意：状态上报消息中的标签ID，跟标签外壳上二维码MAC地址并不相同，而是二维码MAC地址的反序，两者的对应关系参见《3.4 标签ID地址定义规则》。*

并且在data1中包含状态上报数据



### 更新标签图片

电子标签的图片制作参见《电子标签图片制作.docx》

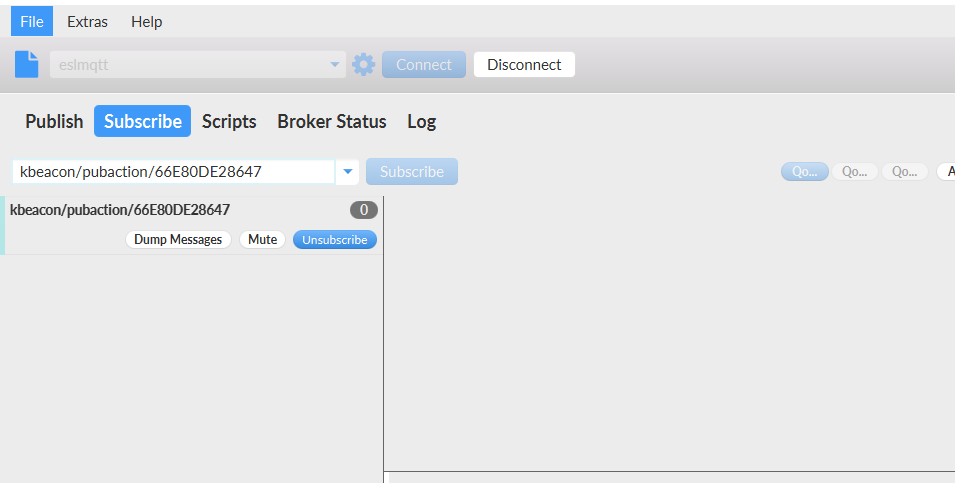
步骤1）获取网关pubaction主题名称，

网关通过该主题向MQTT服务器发布图片更新应答，对于MQTT客户端而言，需要订阅该主题，以便接收到网关的更新应答，这样可以得知图片是否更新成功。

* 如果您没有登录到网关修改过mqtt pubaction主题名称，则主题名称默认为：kbeacon/pubaction/mac地址，例如kbeacon/pubaction/66E80DE28647

*注意：订阅主题中的mac地址(66E80DE28647)是网关的设备ID，不是标签的设备ID。*

步骤2）切换到MQTT.fx软件的Subscribe页面，启动pubaction主题的订阅。



步骤3）获取网关subaction主题名称，网关通过该主题接收MQTT服务器发来的图片更新请求。

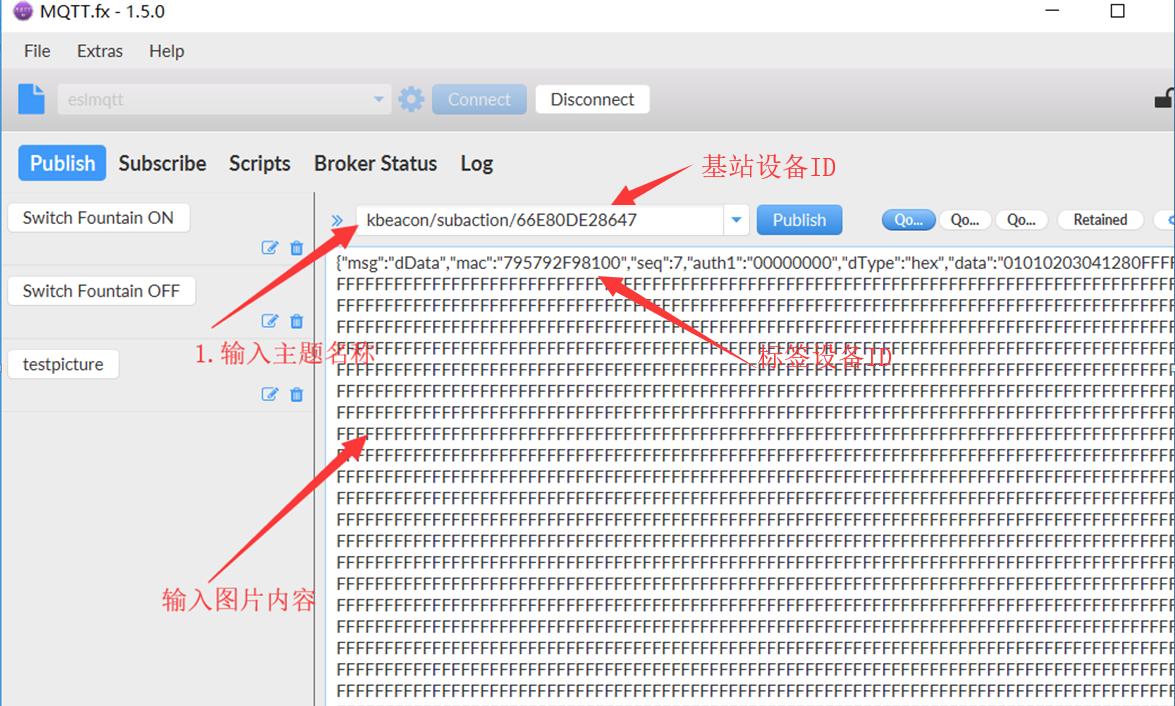
* 如果您没有登录到网关修改过mqtt pubaction主题名称，则主题名称默认为：kbeacon/subaction/mac地址，例如kbeacon/subaction/66E80DE28647

步骤4）切换到MQTT.fx的Publish页面，发送更新图片信息

* 输入发布的主题名称为：kbeacon/subaction/66E80DE28647；
* 输入图片的内容，下面有一个例子
* 点击Publish发布消息；

*注意：发布主题中的mac地址(66E80DE28647)是网关的设备ID，不是标签的设备ID。JSON消息中的mac才是标签的设备ID。*

*注意：标签ID，跟标签外壳上二维码MAC地址并不相同，而是二维码MAC地址的反序，两者的对应关系参见《3.4 标签ID地址定义规则》。*



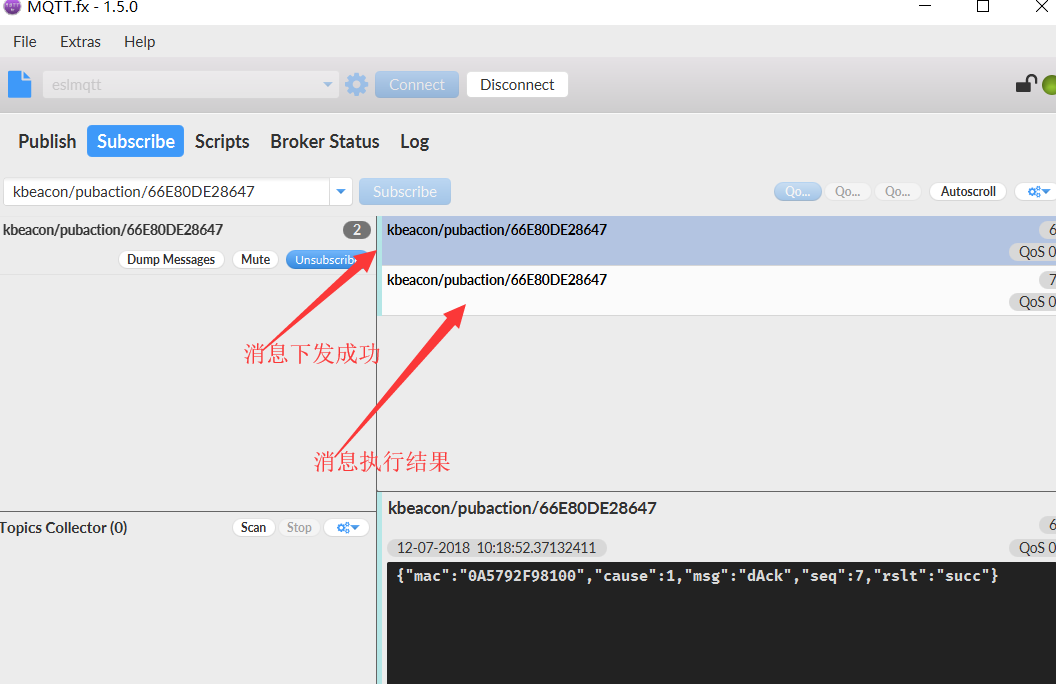
样例：如下消息表示在标签上显示“Test pass”，您可以用记事本打开如下文档，然后将内容拷贝到mqtt.fx的消息框中。

* 记住，您需要将mac替换为标签的id，（在5.3.1节中可以获取标签的id）。
* 另外，如果您修改过标签的密码，则需要将auth1字段对应的密码更新为您修改过的密码。



步骤5）查看图片更新执行结果

如果执行失败，则可以根据返回的cause原因值，查找对应的原因。这里的原因值对应的是十进制，在4.5节消息中可以根据错误码获取对应的原因。



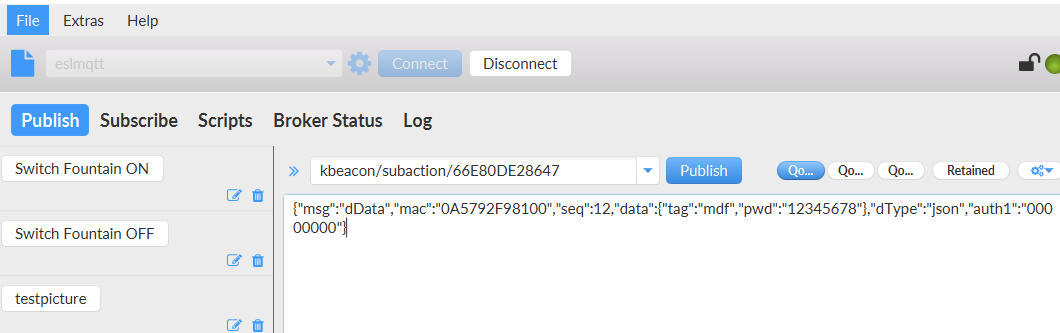
### 更新标签的密码

更更新标签图片操作类似。只是需要在输入图片内容框中，将消息内容替换为修改密码的消息。

同样，消息执行完之后，会有两条消息应答。

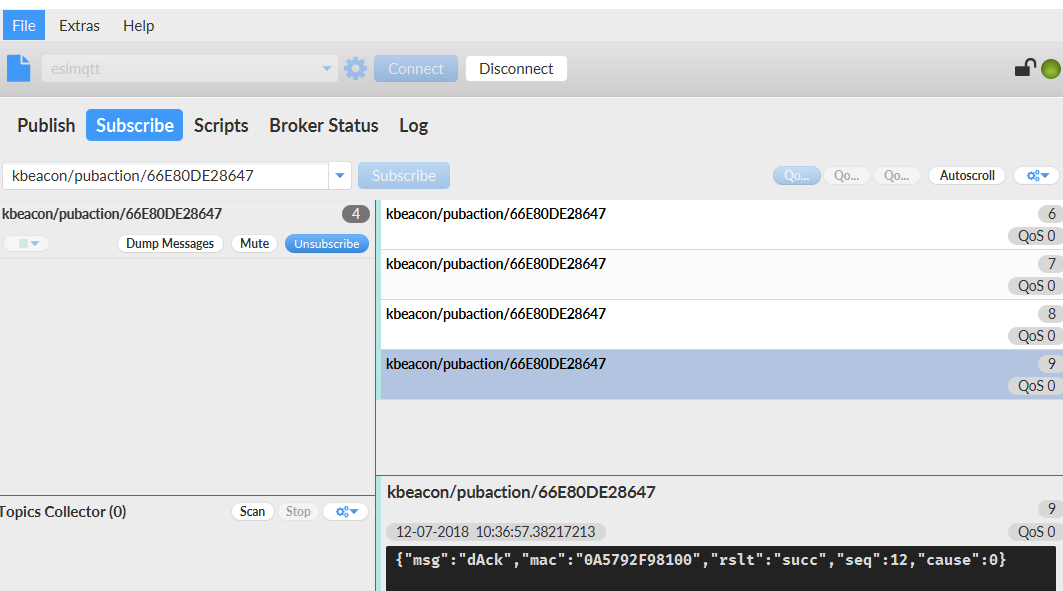
举例：如下消息表示将标签的密码从“00000000”修改为“12345678”

更新请求消息：



{"msg":"dData","mac":"0A5792F98100","seq":12,"data":{"tag":"mdf","pwd":"12345678"},"dType":"json","auth1":"00000000"}

执行结果，更新密码成功：



## 安装MQTT服务器

目前，第三方MQTT服务端软件有好几种，下面以服务器apache-apollo-1.7.1为例，在windows环境下测试。

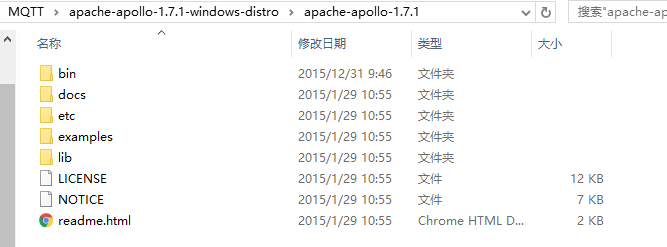
### 下载软件

<http://activemq.apache.org/apollo/download.html>

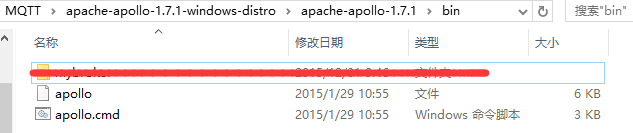
也可以在百度网盘目录下载。

选择windows版本的。

这个软件是免安装的，因此下载后直接解压，如下图所示：

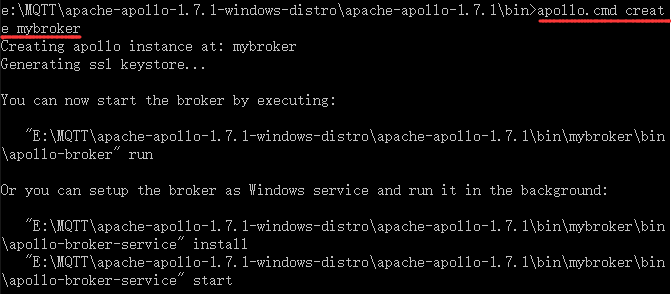


bin下包含apollo和apollo.cmd两个文件：

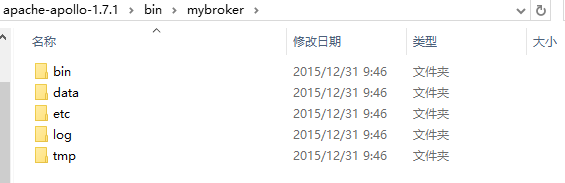


### 安装MQTT服务

运行apache-apollo-1.7.1\bin\apollo.cmd，输入create mybroker（名字任意取，这里是根据[官网](http://activemq.apache.org/apollo/documentation/getting-started.html)介绍的来取的）创建服务器实例，服务器实例包含了所有的配置，运行时数据等，并且和一个服务器进程关联。如果双击apollo.cmd出现闪一下就关闭的情况，则需要在命令行中敲入命令：



create mybroker之后会在bin目录下生成mybroker文件夹。



#### 配置用户名以及IP地址

上面步骤create mybroker之后会在bin目录下生成mybroker文件夹

**1）添加用户：**

其中etc\users.properties文件包含连接MQTT服务器时用到的用户名和密码，后面会介绍，可以修改原始的admin=password，可以接着换行添加新的用户名密码。

下面举例增加了一个用户名为test，密码为“testpassword”

\apache-apollo-1.7.1\bin\mybroker\etc\users.properties里，打开users.properties文件：

　　## ---------------------------------------------------------------------------  
　　## Licensed to the Apache Software Foundation (ASF) under one or more  
　　## contributor license agreements. See the NOTICE file distributed with  
　　## this work for additional information regarding copyright ownership.  
　　## The ASF licenses this file to You under the Apache License, Version 2.0  
　　#  
　　# admin=ENC(Cf3Jf3tM+UrSOoaKU50od5CuBa8rxjoL)  
　　#

admin=password

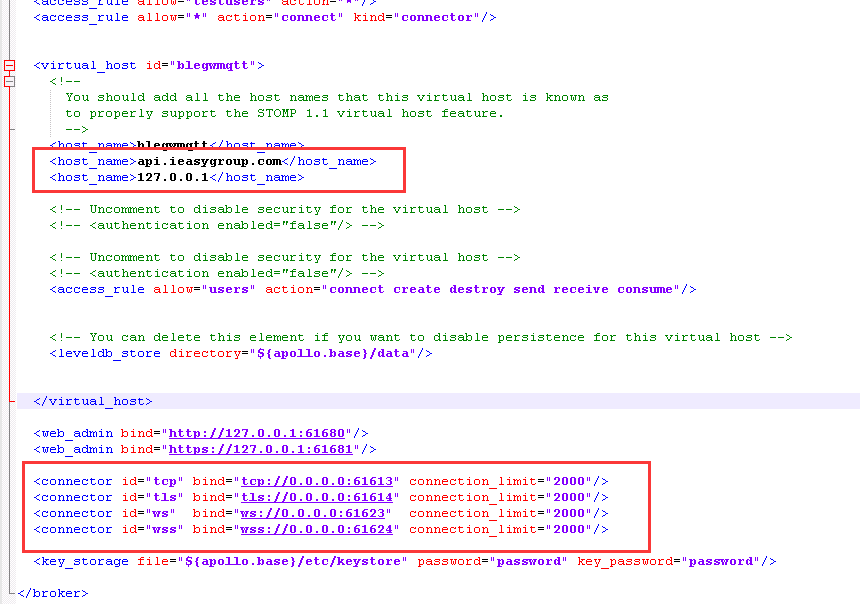
test=testpassword

**2）修改服务器IP地址**

打开etc\apollo.xml文件下是配置服务器信息的文件：

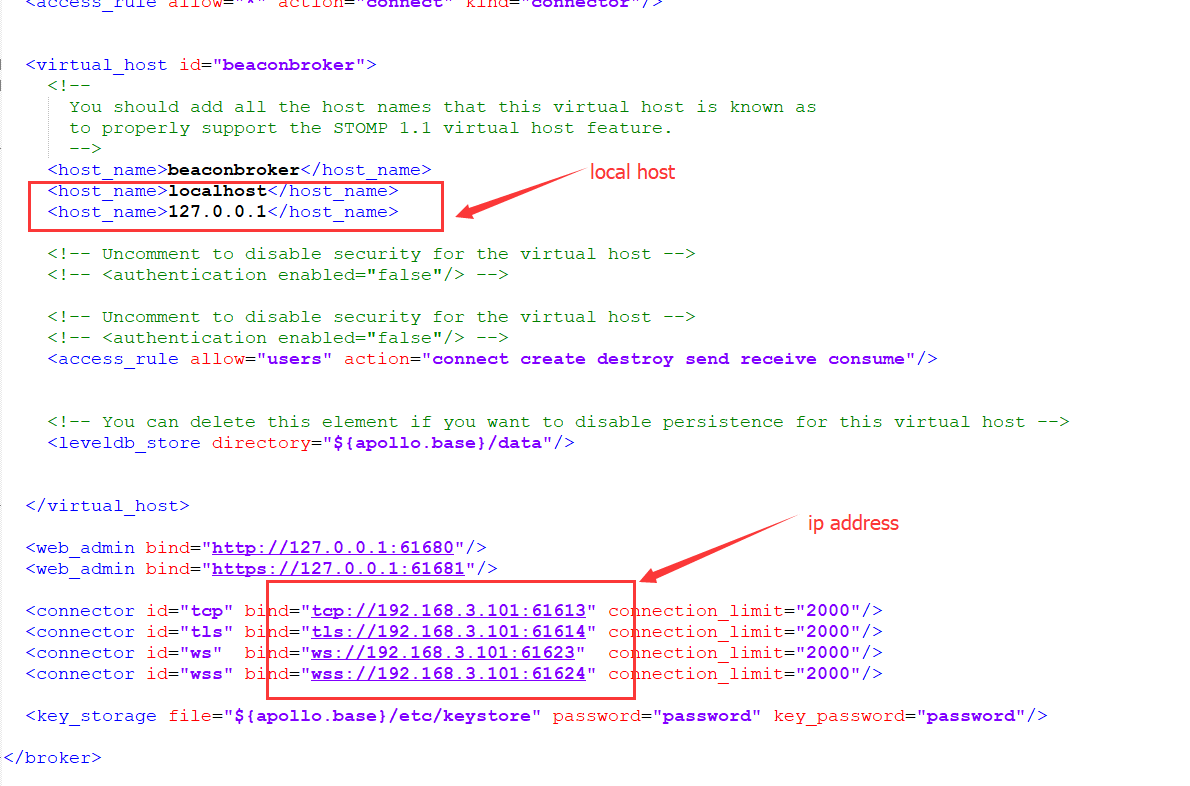
* 云端部署场景：如果MQTT采用域名对外服务，比如域名为api.ieasygroup.com，则

下面的api.ieasygroup.com是指云端的域名



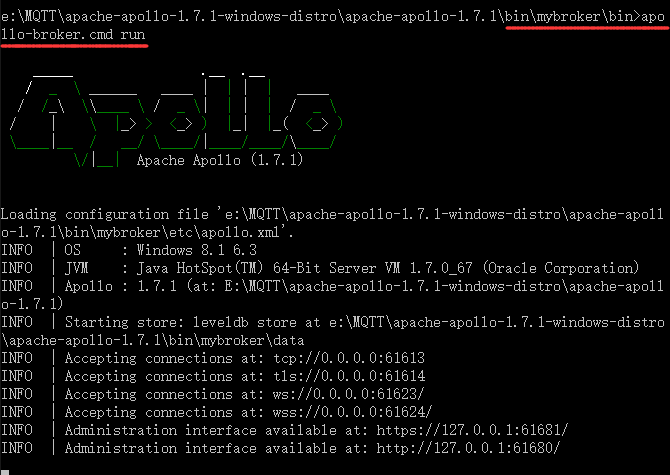
* 本地局域网部署场景： MQTT服务器采用IP地址对外服务，建议采用如下方式配置。

下面的IP地址是指局域网的IP地址。



### 运行MQTT服务

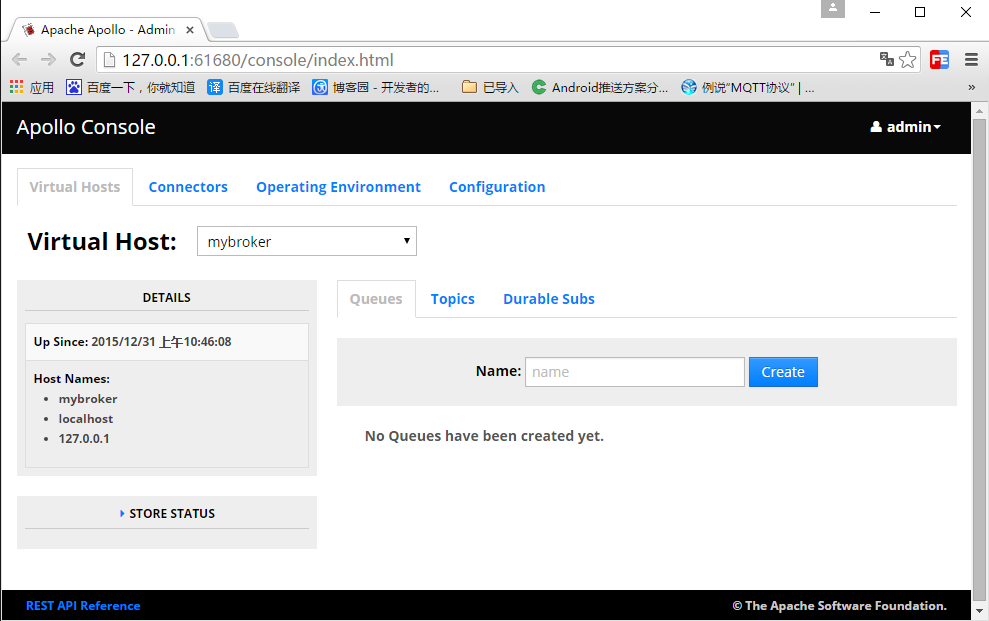
打开cmd，运行apache-apollo-1.7.1\bin\mybroker\bin\apollo-broker.cmd run 开启服务器，如下图：



### 验证MQTT服务运行

可以在浏览器中输入<http://127.0.0.1:61680/>，其自动转入：http://127.0.0.1:61680/console/index.html，apollo的登录页面。

经过上面的简单步骤，服务器基本上就已经完成。输入admin,password就可以登录了，如下图：



### 采用MQTT客户端登录验证

请先参考5.2节安装MQTT客户端软件，连接方式跟5.2节类似，只是需要修改服务端的连接参数。

