# Alibaba Cloud 物#网平台

ベストプラクティス

Document Version20191220

1 デバイスアクセス	1
1.1 MQTT.fx を使用して IoT Platform に接続	1
1.2 Android Things を Alibaba Cloud IoT Platform に接続	9

## 1 デバイスアクセス

### 1.1 MQTT.fx を使用して IoT Platform に接続

ここでは、MQTT.fx を例に取り上げ、サードパーティの MQTT クライアントを使用して IoT Platform に接続する方法を説明します。 MQTT.fx は Java 言語で記述した Eclipse Paho ベー スの MQTT クライアントです。 サブスクライブ方式のメッセージングおよびパブリッシュ方式 のトピックを介したメッセージングをサポートします。

### 前提条件

IoT Platform コンソールでプロダクトとデバイスを作成し、デバイスの

**ProductKey、DeviceName、**および **DeviceSecret** を入手しました。 **MQTT.fx** の接続パ ラメーター設定時は、**ProductKey、DeviceName、**および **DeviceSecret** の値を使用しま す。プロダクトおよびデバイスを作成する際のヘルプは、「*#unique\_3*」、「*#unique\_4*」、およ び「*#unique\_5*」をご参照ください。

### 手順

1. MQTT.fx ソフトウェアをダウンロードしてインストールします。

Windows 版 MQTT.fx のダウンロード: http://mqtt-fx.software.informer.com/download/

Mac版 MQTT.fx のダウンロード: http://macdownload.informer.com/mqtt-fx/

2. MQTT.fx を起動し、[設定] アイコンをクリックします。

MQTT.fx - 1.7.1		
File Extras Help		
iot connection	Connect Disconnect	•
Publish Subscribe Scripts Broker Status	Log	
	Publish	0550 Q051 Q052 Retained

3. 接続パラメーターを設定します。

現在、TCP および TLS の 2 種類の接続モードがサポートされています。 この 2 つのモードの 違いは、クライアント ID と SSL/TLS の設定だけです。

手順は以下のとおりです。

a. 基本情報を入力します。 パラメーターの説明は、以下の表をご参照ください。

[全般] のデフォルトのパラメーターを保持、または必要に応じて値を設定することもでき ます。

Profile Name	iot connection	
Profile Type	MQTT Broker	MQTT
MQTT Broker Profile Settings		
Broker Address	fOAt5H5TOWF.iot-as-mqtt.cn-shanghai.aliyuncs.r	
Broker Port	1883	
Client ID	12345 securemode=3,signmethod=hmacsha1	Generate
General User Credentials S Connection Timeout Keep Alive Interval Clean Session Auto Reconnect	SSL/TLS Proxy LWT	
Max Inflight	10	
MQTT Version	✓ Use Default	
	3.1.1	
	Clear Publish History	
	Clear Subscription History	

パラメーター	説明			
Profile Name	カスタマイズプロファイル名を入力します。			
Profile Type	[MQTT ブローカー] を選択します。			
Broker Address	\${YourProductKey}.iot-as-mqtt. \${region}.aliyuncs.comの書式で接続ドメインを入力しま す。この書式で、可変の \$ <i>{region}</i> は、 <b>IoT Platform</b> のサー ビスリージョンのリージョン <b>ID</b> を示しています。 リージョン <b>ID</b> は、「リージョンおよびゾーン」をご参照ください。 コーディン グ例: alPUPCoxxxx.iot-as-mqtt.cn-shanghai.aliyuncs.com			

パラメーター	説明
Broker Port	1883 に設定します。

パラメーター	説明
Client ID	\${clientId} securemode=3,signmethod=hmacsha1 の書 式で値を入力します。コーディング例:12345 securemode=3, signmethod=hmacsha1 パラメーターの記述は以下のとおりで す。
	<ul> <li>・ \${clientId}は、カスタマイズクライアントIDです。64文</li> <li>字以内の任意の値にすることができます。 クライアントIDの 値はお使いのデバイスの MAC アドレスまたは SN コードを使 用するよう推奨します。</li> </ul>
	<ul> <li>securemode は接続のセキュリティモートです。 ICP モート を使用する場合は、securemode=3 に設定します。 TLS モー ドを使用する場合は、securemode=2 に設定します。</li> <li>signmethod は、使用する署名メソッドです。 IoT Platform では hmacmd5 および hmacsha1 をサポートしています。</li> </ul>
	<b>注</b> 注: クライアント ID 情報を入力した後は、[全般」をクリックしない ようにしましょう。

**b.** [ユーザー資格情報] をクリックして、[ユーザー名] と [パスワード] を入力します。

Profile Name	iot connection	
Profile Type	MQTT Broker	
MQTT Broker Profile Settings		
Broker Address	fOAt5H5TOWF.iot-as-mqtt.cn-shanghai.aliyuncs.c	
Broker Port	1883	
Client ID	12345 securemode=3,signmethod=hmacsha1	Generate
General User Credentials	SSL/TLS Proxy LWT	
User Name	device&fOAt5H5TOWF	
Password	•••••	

パラメーター	説明
User Name	デバイス名とプロダクトキーを <b>"&amp;"</b> 文字で直接つないだもの を指定する必要があります。 書式 <b>:</b> \${YourDeviceName}& \${YourPrductKey}. 例 <b>: device&amp;fOAt5H5TOWF.</b>

パラメーター	説明			
Password	入力パラメーターの暗号化値を入力する必要があります。 IoT Platform ではパスワードを簡単に生成できるパスワードジェネ レータを提供しています。 ご自身で暗号化することもできます。			
	・ パスワードジェネレータのパラメーター <b>:</b> 			
	<ul> <li>productKey:デバイスが属するプロダクトの一意の識別</li> <li>子。この情報はコンソールのデバイス詳細ページで閲覧できます。</li> </ul>			
	- <b>deviceName :</b> デバイスの名前 この情報はコンソールのデ バイス詳細ページで閲覧できます。			
	- <b>deviceSecret :</b> デバイスシークレット この情報はコンソー ルのデバイス詳細ページで閲覧できます。			
	<ul> <li>timestamp:(省略可能)現在のシステム時間のタイムスタンプ</li> </ul>			
	<ul> <li>clientId:カスタマイズクライアントID。クライアントID の \${clientId} と同じ値にする必要があります。</li> </ul>			
	<ul> <li>method:署名アルゴリズム。クライアント ID の signmethod と同じ値にする必要があります。</li> </ul>			
	・ パスワードを手動で生成します。			
	A. パラメーターをソートして結合します。			
	clientId、deviceName、productKey、および			
	timestamp の各パラメーターを辞書式順序でソートし結			
	合します。 (タイムスタンプを設定していない場合は、文			
	字列にタイムスタンプを記述しないようにします)。 結合			
	した文字列の例:clientId12345deviceNamedevicep			
	roductKeyf0At5H5T0WF			
	<b>B.</b> 暗号化します。			
	Client ID で定義されている署名アルゴリズムにより結合 文字列を暗号化する秘密鍵として、お使いのデバイスの deviceSecret を使用します。			
	デバイスの deviceSecret は <b>abc123</b> 、暗号化形式			
	はhmacshal(abc123,clientId12345deviceN			
	amedeviceproductKeyfOAt5H5TOWF)とします。			

**c. TLS** 接続モードを使用している場合は、SSL/TLS の情報を設定することが必須となりま す。接続モードが TCP の場合、SSL/TLS の設定は必須ではありません。 [SSL/TLS を有効にする] チェックボックスをオンにし、プロトコルとして TLSv1 を選択 します。

General	User Credentials	SSL/TLS	Proxy	LWT			
	Enable SSL/T	LS 🗸			Protocol	TLSv1	•
• CA sign	ed server certificate						
CA certificate file							
CA cert	ificate keystore						
Self signed certificates							
🕖 Self sigr	ned certificates in keyste	ores					

- d. 必要な情報をすべて入力して、[OK] をクリックします。
- 4. [接続] をクリックして IoT Platform に接続します。

MQTT.tx - 1.7.0		
File Extras Help		
iot connection	Connect     Disconnect	
Publish Subscribe Scripts Broker S	tatus Log	
) home/garden/fountain	▼ Publish	QoS 0 QoS 1 QoS 2 Retained

### メッセージ通信テスト

MQTT.fx と IoT Platform が正常に接続されているかどうかテストします。

1. MQTT.fx で、[サブスクライブ] をクリックします。

2. デバイスのトピックを入力し、[サブスクライブ] をクリックします。

👦 MQTT.fx - 1.7.1	anation a businesses	
File Extras Help		
iot connection	Connect Disconnect	₽ 🔴
Publish Subscribe Scripts Broker	Status Log	
/a1jxcmPgkra/light/user/get	Subscribe	QoS 0 QoS 1 QoS 2 Autoscroll OST
Topics Collector (0) Scan Stop OGT		
		DUB Retained

トピックに正常にサブスクライブすると、トピックはトピック一覧に表示されます。

₩QTT.fx - 1.7.1	
File Extras Help	
iot connection	Connect Disconnect
Publish Subscribe Scripts Broker Status I	Log
/tOAt5H5TOWF/device/get	subscribe Qd51 Qd51 Qd52 Atsoci Q+
/IOAISHSTOWF/device/get	/YOAtSHSTOWF/device/get
Topics Collector (0) Scan Stop Colw	

**3.** *IoT Platform* コンソールの [デバイス詳細] ページの [トピック一覧] で、サブスクライブしたト ピックの [パブリッシュ] ボタンをクリックします。 4. メッセージの内容を入力し、[OK] をクリックします。

IoT Platform	Devices > Device	 Publish		×		
Products Devices	Product : light Vie Device Inform	Note: When the Topic is being used by an applica caution to prevent applications exceptions.	tion, perform operation:	s with	riceSecret : ******** Sho	w
Rules Extended Services My Services ~	Device Topic Li:	Topic : /llastic/device/get * Notification Content:				
Documentation	Device Topic	Test message.	13/1000		ublished A lotifications: A	actions Publish
	/alasekinemg	* Qos: 0 \[ 1			F	<sup>f</sup> ublish Publish
			ОК	Cancel		

5. MQTT.fx に戻り、メッセージが受信されたかどうか確認します。

WQTT.fx - 1.7.1		
File Extras Help		
iot connection	Connect Disconnect	<b>e</b> 😑
Publish Subscribe Scripts Broker Status	Log	
/fOAt5H5TOWF/device/get	Subscribe QoS 0 QoS 1	L Qo5 2 Autoscrol OST
/fOAt5H5TOWF/device/get	/fOAI5H5TOWF/device/get	1 QoS 0
earry manager make ended	/fOAI5H5TOWF/device/get	2 QoS 0
Topics Collector (0) Scan Stop or		
	/fOAt5H5TOWF/device/get	2
	23-07-2018 14:37:02:528:22940 Test message	Qos 0

### ログの表示

MQTT.fx で操作ログとエラーログを閲覧するには、[ログ] をクリックします。

MO	TT.fx - 1	1.7.1		
Pile	Extras	Help		
-			A	-0 <b>•</b>
			• ¥	Disconnect
Pu	blish	Subscribe 5	Scripts Broker Status	
2018-	07-23	14:34:28,347	INFO MqttFX ClientModel	: sucessfully subscribed to topic /fOAt5H5TOWF/device/get (QoS 0)
2018-	07-23	14:34:53,108	INFO MqttFX ClientModel	: messageArrived() with topic: /fOAt5H5TOWF/device/get
2018-	07-23	14134153,118	INFO MqttFX ClientModel	: messageArrived() added: message #1 to topic '/fOAtSHSTOWF/device/get'
2018-	07-23	14:35:05,801	INFO MqttFX ClientModel	: rebuildMessagesList()
2018-	07-23	14:35:05,807	INFO MqttFX ClientModel	: repulldMessagesList()
2018-	07-23	14:35:05,807	THEO Refer ibeController	: sucessfully unsubscribed from topic: /funtsusfum/device/get
2018-	07-23	14-35-46 103	TNEO NottEY ClientHodel	· undustriale
2018-	67-23	14:35:46,102	TNEO HattEX (lientHode)	: attent to addRecentSubscriptionTonic
2018-	07-23	14:35:46.104	INFO MattEX ClientModel	: addBecentSubscriptionTonic : de.iensd.mottfx.entities.Tonic2403d4071
2018-	07-23	14:35:46.105	INFO HattFX ClientHodel	: attempt to add PublishTopic
2018-	07-23	14:35:46.109	INFO MgttFX ClientModel	: sucessfully subscribed to topic /fOAtSHSTOWF/device/get (QoS 0)
2018-	07-23	14:37:02,938	INFO HqttFX ClientHodel	: messageArrived() with topic: /fOAt5H5TOWF/device/get
2018-	07-23	14:37:02,939	INFO MgttFX ClientModel	: messageArrived() added: message #2 to topic '/fOAt5H5TOWF/device/get'
2018-	07-23	14:37:34,214	INFO PublishController	: publish
2018-	07-23	14:37:34,216	INFO MqttFX ClientModel	: attempt to add PublishTopic
2018-	07-23	14:37:34,216	INFO MqttFX ClientModel	: sucessfully published message to topic /fOAt5H5TOWF/device/get (QoS 0,
2018-	07-23	14:38:12,095	INFO PublishController	: publish
2018~	07-23	14:38:12,096	INFO MqttFX ClientModel	: attempt to add PublishTopic
2018-	07-23	14:38:12,097	INFO MqttFX ClientModel	: sucessfully published message 发送消息 to topic /fOAt5HSTOWF/device/get
2018-	07-23	14:38:29,967	INFO PublishController	: publish
2018-	07-23	14:38:29,968	INFO MqttFX ClientModel	: attempt to add PublishTopic
2018-	07-23	14138129,968	INFO MqttFX ClientModel	: sucessfully published message 发展消息 to topic /fOAtSHSTOWF/device/get
2018-	07-23	14:39:30,697	INFO SubscribeController	: onSubscribe
2018-	07-23	14:39:30,715	TNFO MattEX ClientModel	: repullomessagesList()
2018-	07-23	14:37:30,719	THEO HattEX (lightHode)	: addDacantSubscriptionTonic : do ioned mottfy antitias TanicOLRobeleS
2018-	07-23	14-30-30 730	TNEO NattEY CliantHodal	· autoecentionstraption opt. · de Jensu autorites represented
2018-	67-23	14:39:30,720	TNEO HattEX (lientHode)	: addbublishTonic : /f01t585T00E/device/test
2018-	07-23	14:39:30.727	INFO MgttFX ClientModel	: sucessfully subscribed to topic /fQAtSHSTOWF/device/test (0oS 0)
2018-	07-23	14:39:39.010	INFO PublishController	: publish
2018-	07-23	14:39:39.011	INFO MgttFX ClientModel	: attempt to add PublishTopic
2018-	07-23	14:39:39,011	INFO MqttFX ClientModel	: sucessfully published message 发送演包 to topic /fOAt5H5TOWF/device/test
2018-	07-23	14:39:39,037	INFO MqttFX ClientModel	: messageArrived() with topic: /fOAt5H5TOWF/device/test
2018-	07-23	14:39:39,038	INFO MqttFX ClientModel	: messageArrived() added: message #3 to topic '/fOAt5H5TOWF/device/test'

### 1.2 Android Things を Alibaba Cloud IoT Platform に接続

この記事では、室内空気試験プロジェクトを例として Google Android Things を Alibaba Cloud IoT Platform に接続する方法を説明します。

### ハードウェア

・ 本プロジェクトのハードウェア一覧

以下の表に、室内空気試験プロジェクトに必要なハードウェアを示します。

ハードウェア	画像	備考
NXP Pico i.MX7D		Android Things 1.0
開発ボード 		<b>注</b> :
		<b>Raspberry Pi</b> も使 用できます。

ハードウェア	画像	備考
<b>DHT12</b> 温湿度センサー		I2C データ通信方式 をサポートします。
<b>ZE08-CH2O</b> ホルムアルデヒド検 出センサー		UART データ通信 方式をサポートしま す。

L



・NXP i.MX7D 開発ボードピンのレイアウト図

NXP Pico i.MX7D の詳細は、『https://developer.android.com/things/hardware/imx7d-pico-io』 をご参照ください。.

#### ・ ハードウェア接続図



- 温湿度センサー (DHT12) の SCL (クロックライン) ピンと SDA (データライン) ピンを開 発ボードの I2C SCL ピンと SDA ピンに接続します。
- ホルムアルデヒド検出センサー (ZE08-CH2O)の TXD (データ伝送) ピンを開発ボードの RXD (データ受信) ピンに接続し、ZE08-CH2Oの RXD ピンを開発ボードの TXD ピンに 接続します。

Alibaba Cloud IoT Platform コンソールでプロダクトとデバイスを作成

1. IoT Platform コンソールにログインします。

### 2. IoT Platform Pro でプロダクトを作成します。

[プロダクト] ページで [プロダクトの作成] をクリックします。 プロダクトを作成するバー ジョンとして **Pro Edition** を選択します。 詳細は、*#unique\_3*をご参照ください。

IoT Platform	Products	Create Product	$\times$			
Products	All(2)	* Select version: Basic Edition Pro Edition				
Rules	Product List	* Product Name: Air_test				
Extended Services	Product Name :	* Node Type:			Refresh	Create Product
Documentation	Product Name	* Data Type:		tal vices	Created At	Actions
	documentationt	Product Description:			08/02/2018, 17:13:49	View Delete
	light	Enter a product description.			07/18/2018, 16:41:40	View Delete
		0/100		rev 1		Items per Page: 10 $\checkmark$
		OK Car	ncel			

3. 新しく作成したプロダクトの機能を定義します。

[プロダクト詳細] ページで [フィーチャーの定義] > [追加] をクリックし、プロダクトのプロパ ティを定義します。詳細は、*#unique\_*7をご参照ください。

### 表1-1:室内空気試験プロジェクトに必要なプロパティ

プロパティ名	識別子	データ型	値の範囲	説明
Temperature	temperature	float	-50~100	<b>DHT12</b> によっ て検出されま す。
Humidity	humidity	float	0~100	<b>DHT12</b> によっ て検出されま す。

プロパティ名		識別子		データ型	値	ī の範囲	説明
Formaldehy concentratio	aldehyde ch2o entration			double 0~3		~3	<b>ZE08</b> により検 出されます。
IoT Platform Products Devices Rules	Produc Air_te Produc Pro	ts > Product Detai est Pro Edition tKey : duct Information	s Copy Notifications	ProductSecret : ******* Define Feature Dev	Show ice Log	Total Devices:0 Ma Online Debugging	inage
Extended Services My Services 🗸	Define	Feature A standard custom fea	l feature is automatically c itures.	reated based on the device type of t	e product. You ca	an also add optional features or create yo	ur own View TSL Add
Documentation	Featu	re Type	Feature Name:	Identifier:	Data Type	Data Definition	Actions
	Prope	erties	Temperature	temperature	float	Value Range:-50 - 1	00 Edit Delete
	Prope	erties	Humidity	humidity	float	Value Range:0 - 100	) Edit Delete
	Prope	erties	Formaldehyde concentration	ch2o	double	Value Range:0 - 3	Edit Delete

4. デバイスを作成

[デバイス] ページで新しく作成したプロダクトの名前を選択し、[デバイスの追加] をクリック してデバイスを作成します。詳細は、*#unique\_*4をご参照ください。

IoT Platform	Devices	
Products	Total Devices: Add Device Refi	resh
Devices		
Rules	Device List Otte: When the deviceName is left blank, Alibaba Cloud will assign a	
Extended Services	GOLD as the device vame. Add De	vice
My Services V	Product :	
Documentation	DeviceN Last Online Actions DeviceName :	
	OK Cancel I Next > Items per Page:	10 ~
	Batch Delete Batch Disable Batch Enable	

### Android Things クライアントの開発

**1. Android Studio** を使用して **Android Things** プロジェクトを作成し、インターネットの権 限を追加します。

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

2. eclipse.paho.mqtt を Gradle ファイルに追加します。

```
implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.
0'
```

3. DHT12 のデータを I2C で読み込む設定をします。

```
private void readDataFromI2C() {
        try {
            byte[] data = new byte[5];
            i2cDevice.readRegBuffer(0x00, data, data.length);
            // check data
            if ((data[0] + data[1] + data[2] + data[3]) % 256 ! =
data[4]) {
                humidity = temperature = 0;
                return;
            }
            // humidity data
            humidity = Double.valueOf(String.valueOf(data[0]) + "."
+ String.valueOf(data[1]));
            Log.d(TAG, "humidity: " + humidity);
            // temperature data
            if (data[3] < 128) {
                temperature = Double.valueOf(String.valueOf(data[2])
+ "." + String.valueOf(data[3]));
            } else {
                temperature = Double.valueOf("-" + String.valueOf(
           "." + String.valueOf(data[3] - 128));
data[2]) +
            Log.d(TAG, "temperature: " + temperature);
        } catch (IOException e) {
            Log.e(TAG, "readDataFromI2C error " + e.getMessage(), e
);
        }
    }
```

4. Ze08-CH2O のデータを UART で読み込む設定をします。

```
try {
    // data buffer
    byte[] buffer = new byte[9];
    while (uartDevice.read(buffer, buffer.length) > 0) {
        if (checkSum(buffer)) {
            ppbCh2o = buffer[4] * 256 + buffer[5];
            ch2o = ppbCh2o / 66.64 * 0.08;
        }
    }
}
```

5. Alibaba Cloud IoT Platform とクライアントを接続し、データを報告します。

```
/*
Payload format
Ł
  "id": 123243,
  "params": {
    "temperature": 25.6,
    "humidity": 60.3,
    "ch2o": 0.048
  },
"method": "thing.event.property.post"
}
*/
MqttMessage message = new MqttMessage(payload.getBytes("utf-8"));
message.setQos(1);
String pubTopYourPc = "/sys/${YourProductKey}/${YourDeviceName}/
thing/event/property/post";
mqttClient.publish(pubTopic, message);
```

### デバイスのリアルタイムデータを表示

デバイスを有効にすると、IoT Platform コンソールの [デバイスの詳細] ページの [ステータス] カラムからデバイスのリアルタイムデータを閲覧できます。

IoT Platform	Devices > Device Details			
Products	Temperature-sensor			
Devices	Product : Air_test View	ProductKey : Cop	DeviceSecret : *******	** Show
Rules	Device Information Events	Invoke service		
Extended Services	Ctatue Last monthl device properties		Dool time Dofr	chart Form
My Services $\checkmark$	Status Last reported device properties.			
Documentation	Formaldehyde concentration	Temperature	Humidity	
	0.03 <b>mg/m</b> ³	10C°	27%	
	Last update: 2018/07/19 15:50:16	Last update: 2018/07/19 15:50:16	Last update: 2018/07/19 15:50:16	
	View logs	View logs	View logs	