

IoT 示例简介

准备工作

- 有一个 IBM Bluemix 帐号，可以通过 www.bluemix.net 申请
- 懂一点点 Java

IoT 示例简介

IoT Foundation 是什么？

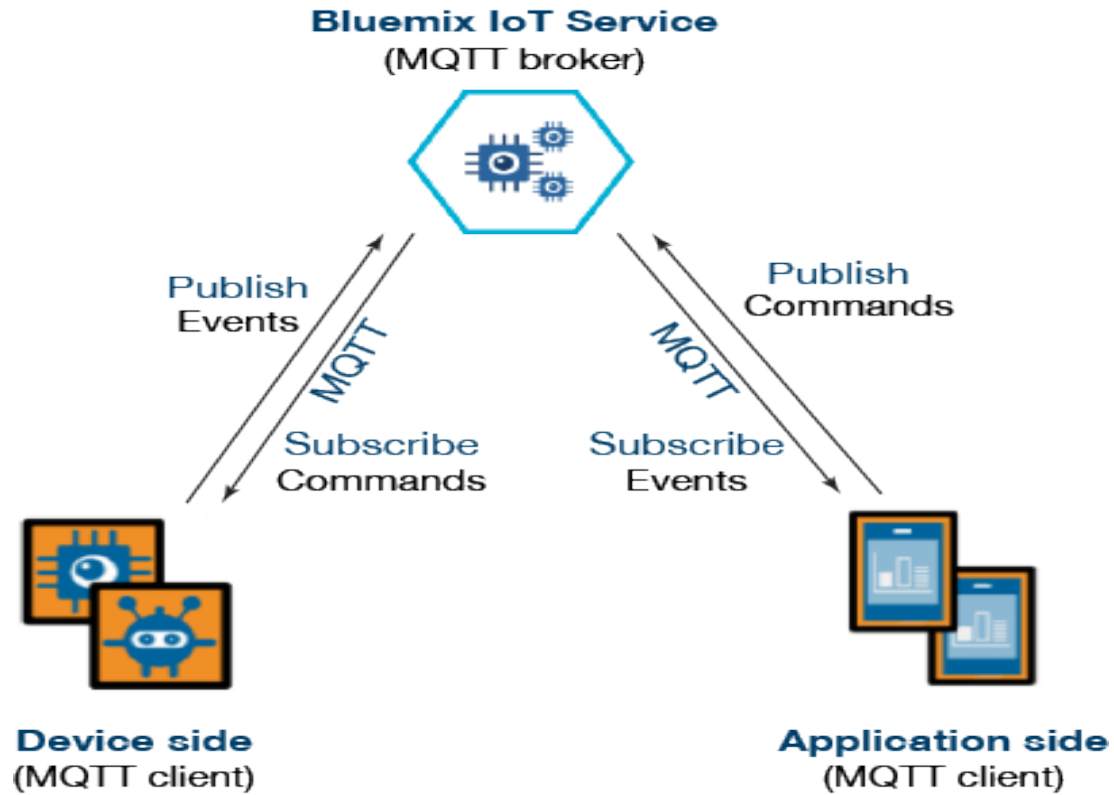
IBM Internet of Things Foundation 可以看作是物联网世界里的消息中心和中转站。在 IoT Foundation 里你可以设置并管理各种不同设备，各种应用可以通过它来访问设备的实时或者历史数据。

IoT 示例简介

MQTT 是什么？

MQTT（Message Queuing Telemetry Transport 消息队列遥测传输）是物联网或者说 Machine-to-Machine 的及时通讯协议。它是轻量级的分发 / 订阅式消息传输协议，你甚至可以通过它让传感器与卫星通信。该协议支持所有平台，可以把所有联网物品和外部连接起来，是物联网的重要组成部分。

IoT 示例简介



典型 IoT 应用架构图

IoT 示例程序

创建一个使用 Bluemix IoT 服务的典型应用通常包含三个步骤：

- 配置 IoT 服务，注册设备 (Device) 和应用 (App)
- 开发设备端程序
- 开发应用端程序

配置 Bluemix IoT 服务

登录到 Bluemix 主页 [https://www.ibm.com/bluemix.net](https://www.ibm.com/bluemix)



One key, many possibilities.

Your IBM id provides access to services, communities, support, online purchasing, and much more.

Tip: To easily return to your destination, bookmark the page after this page. Do not bookmark the sign in page.

[Create IBM id](#)

Sign in

[Forgot password?](#)

[Sign in](#)

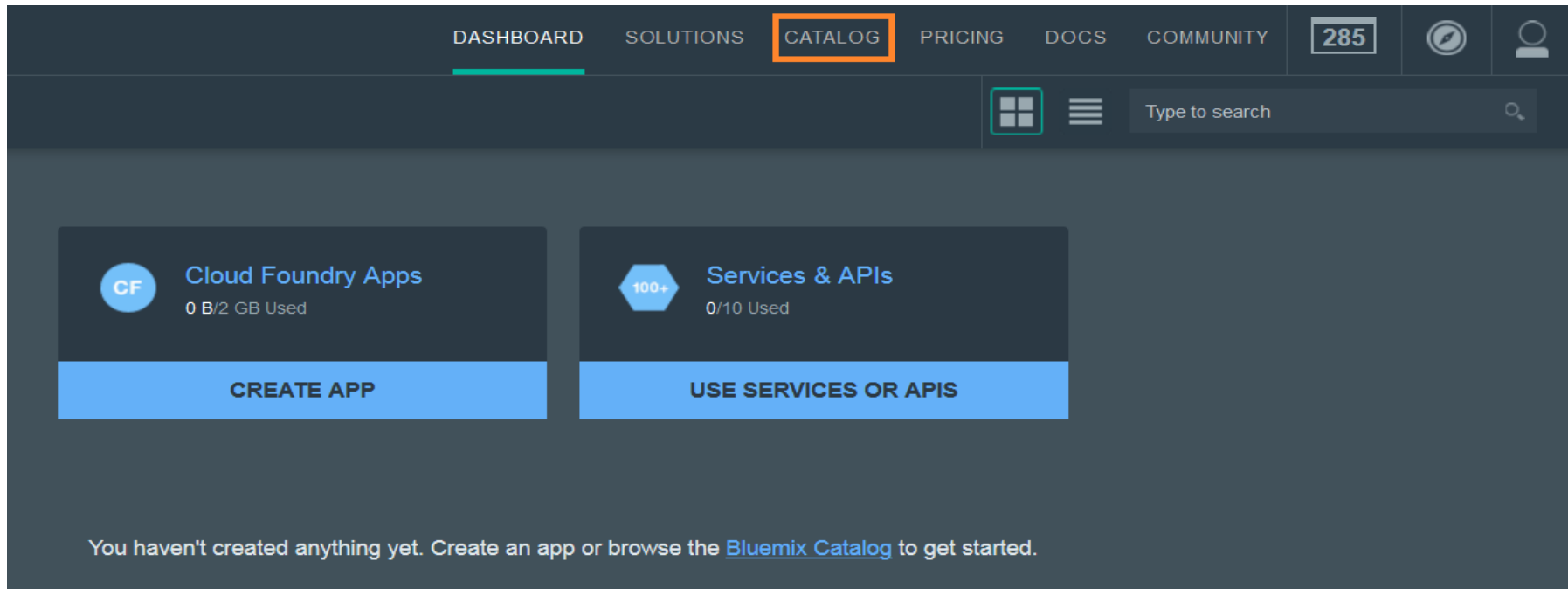
[Help and FAQ](#)

IBM Employees:
[Sign in with your intranet ID.](#)

[Link your Intranet ID and IBM id](#)

配置 Bluemix IoT 服务

菜单上选择 Catalog






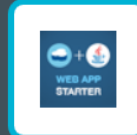










配置 Bluemix IoT 服务

选择 Internet of Things Foundation Starter.

Starters // Choose a package of sample code and services, or start from scratch

Boilerplates

Get started with a new app, now

 Apache Spark Starter IBM	 Internet of Things Foundation Starter IBM	 Java Cache Web Starter IBM	 Java Cloudant Web Starter IBM	 Java DB Web Starter IBM	 Mobile Cloud IBM
 MobileFirst Services Starter IBM	 Node.js Cache Web Starter IBM	 Node.js Cloudant DB Web Starter IBM	 Personality Insights Java Web Starter IBM	 Personality Insights Node.js Web Starter IBM	 Node-RED Starter Community
 Ruby Sinatra Community	 Vaadin Rich Web Starter Community				

配置 Bluemix IoT 服务

Name 输入框里可以输入一个 app 名字，点击 Create

Internet of Things Foundation Starter
IBM

Get started with an Internet of Things Foundation application using Node-RED in Bluemix. Try the sample flow with a simulator and customize it for your own devices.

VERSION
0.4.19

TYPE
Bollerplate

VIEW DOCS

SDK for Node.js™

Cloudant NoSQL DB

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

VIEW DOCS

Pick a plan

Monthly prices shown are for country or region:

Error retrieving plans data

TERMS

Create an app:

Space: MyIoT

Name: bluemix-iot-demo

Host: bluemix-iot-demo

Domain: mybluemix.net

Selected Plan: SDK for Node.js™

Default

Cloudant NoSQL DB

Shared

CREATE

配置 Bluemix IoT 服务

点击导航栏 Overview, 然后点击 Add a service or API

The screenshot shows the Bluemix IoT service configuration page for an application named 'bluemix-iot-demo'. The page is divided into several sections:

- Navigation:** A left sidebar contains navigation links: 'Overview' (highlighted with an orange box), 'SDK for Node.js™', 'Files and Logs', 'Environment Variables', 'Start Coding', and 'SERVICES' (with 'Cloudant NoSQL DB' listed below).
- Application Header:** At the top, it shows the application name 'bluemix-iot-demo', a 'Back to Dashboard...' link, and an 'ADD GIT' button.
- Configuration Panel:** A central panel displays the current configuration for the 'SDK FOR NODE.JS™' service. It includes:
 - INSTANCES:** A dropdown menu set to '1'.
 - MEMORY QUOTA:** A dropdown menu set to '512' (MB per Instance).
 - AVAILABLE MEMORY:** A display showing '0 B'.
 - Buttons for 'SAVE' and 'RESET'.
- Service Selection:** Below the configuration panel, there are two large buttons: 'ADD A SERVICE OR API' (highlighted with an orange box) and 'BIND A SERVICE OR API'.
- Cloudant NoSQL DB:** A section showing the 'Cloudant NoSQL DB' service, with the instance name 'bluemix-iot-demo-cl...' and a 'Shared' status. It includes a 'Show Credentials' button and a 'Docs' link.
- APP HEALTH:** A section indicating 'Your application is staging.' with a refresh icon.
- ACTIVITY LOG:** A section showing a list of recent activities:
 - 8/5/15 11:56 AM: 0496170019@tongji.edu.cn started bluemix-iot-demo app
 - 8/5/15 11:55 AM: 0496170019@tongji.edu.cn updated bluemix-iot-demo app (changed routes)
 - 8/5/15 11:55 AM: 0496170019@tongji.edu.cn created bluemix-iot-demo app
- Cost Estimation:** A bottom right section with the text 'Estimate the cost of this app' and a right-pointing arrow button.

配置 Bluemix IoT 服务

勾选导航栏 Internet of Things, 右边选择 Internet of Things



配置 Bluemix IoT 服务

点击 CREATE



Internet of Things
IBM

PUBLISH DATE
06/23/2015

AUTHOR
IBM

TYPE
Service

LOCATION
US South

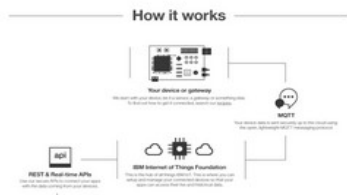
[VIEW DOCS](#)

The IBM Internet of Things service lets your apps communicate with and consume data collected by your connected devices, sensors, and gateways. Our recipes make it super easy to get devices connected to our Internet of Things cloud. Your apps can then use our real-time and REST APIs to communicate with your devices and consume the data you've set them up to collect.

- **Connect your devices securely to the cloud**
- **Build an app that talks to your devices**

Before your apps can get to work, you need to get your devices connected up! We have a set of verified instructions, or 'recipes', for connecting devices, sensors and gateways from a variety of partners and individuals.

Communications between your devices and the cloud happen via the open, lightweight MQTT protocol. For example you might have a sensor that collects and sends humidity readings every minute. Our REST and real-time APIs allow you to quickly pull that device data into your apps for further analysis.



The 'Add Service' form in the Bluemix IoT console. It includes the following fields and options:

- Space:** MyIoT
- App:** bluemix-iot-demo
- Service name:** Internet of Things-zr
- Selected Plan:** Free
- CREATE** button

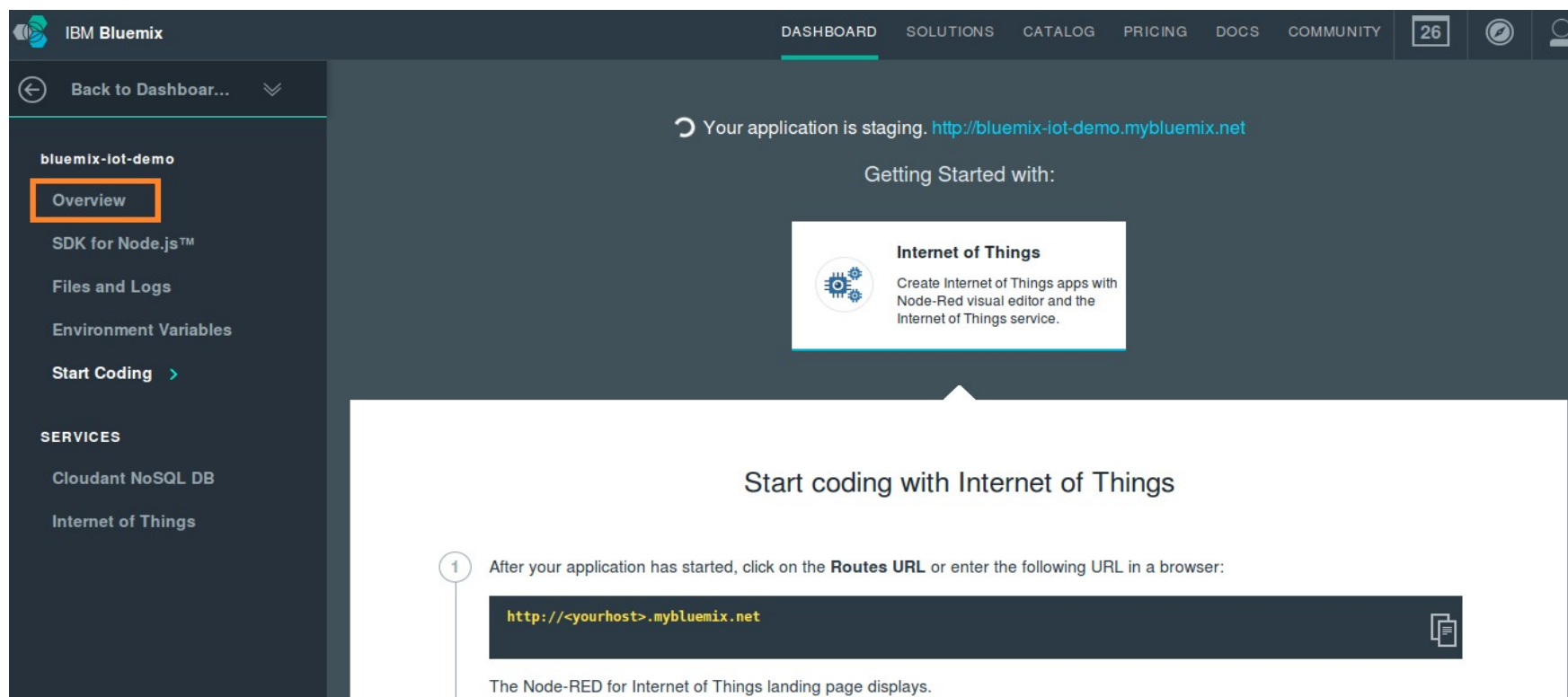
配置 Bluemix IoT 服务

点击 RESTAGE

The screenshot shows the IBM Bluemix dashboard for an application named 'bluemix-iot-demo'. The main content area displays a success message: 'Your app is running. <http://bluemix-iot-demo.mybluemix.net>'. Below this, there is a 'Getting Started with:' section featuring a card for 'Internet of Things' with the text 'Create Internet of Things apps with Node-Red visual editor and the...'. A modal dialog box titled 'Restage Application' is overlaid on the screen. The dialog contains the following text: 'Your 'bluemix-iot-demo' app must be restaged to use the new 'Internet of Things-zr' service. Restaging makes this service available for use. Do you want to restage it now?'. At the bottom of the dialog, there are two buttons: 'RESTAGE' (highlighted with an orange border) and 'CANCEL'. In the background, a numbered step '1' indicates: 'After your application has started, click on the **Routes URL** or enter the following URL in a browser: `http://<yourhost>.mybluemix.net`'. Below this, it states: 'The Node-RED for Internet of Things landing page displays.'

配置 Bluemix IoT 服务

点击 Overview



The screenshot displays the IBM Bluemix dashboard interface. At the top, there is a navigation bar with links for DASHBOARD, SOLUTIONS, CATALOG, PRICING, DOCS, and COMMUNITY, along with a date indicator (26) and a user profile icon. The left sidebar contains a navigation menu for the 'bluemix-iot-demo' application, with 'Overview' highlighted by an orange box. Other menu items include 'SDK for Node.js™', 'Files and Logs', 'Environment Variables', 'Start Coding >', and a 'SERVICES' section with 'Cloudant NoSQL DB' and 'Internet of Things'. The main content area shows a message: 'Your application is staging. <http://bluemix-iot-demo.mybluemix.net>'. Below this, a 'Getting Started with:' section features a card for 'Internet of Things' with a gear icon and the text: 'Create Internet of Things apps with Node-Red visual editor and the Internet of Things service.' A large white box contains the heading 'Start coding with Internet of Things' and a numbered step (1) that reads: 'After your application has started, click on the **Routes URL** or enter the following URL in a browser:'. Below this text is a dark box containing the URL `http://<yourhost>.mybluemix.net` and a copy icon. A final line of text states: 'The Node-RED for Internet of Things landing page displays.'

配置 Bluemix IoT 服务

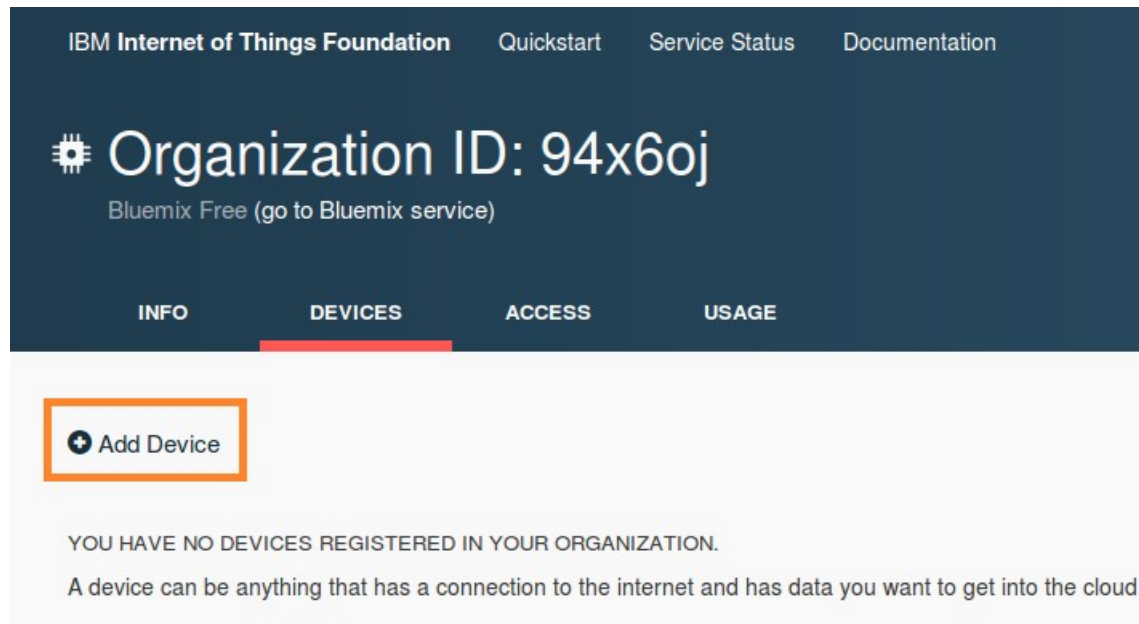
点击 Launch dashboard

The screenshot shows the IBM Bluemix Internet of Things Foundation dashboard. The top navigation bar includes 'DASHBOARD', 'SOLUTIONS', 'CATALOG', 'PRICING', 'DOCS', and 'COMMUNITY'. The main content area is titled 'Hi! Welcome to Internet of Things Foundation' and contains three columns of instructions:

- Connect your devices:** Use our recipes to find out how to add your devices. We work with partners and have sample connection recipes for many devices. Launch the Internet of Things Foundation dashboard and add your devices by clicking the 'Add a device button' under the 'Devices' tab. A button labeled 'Launch dashboard' is highlighted with an orange box.
- Learn how to build your app:** When you have added your devices, you can come back to Bluemix to start building your app using your real-time and historical device data. Read the docs to find out how to make the most out of your app. A button labeled 'Go to docs' is present.
- Learn how to extend your app:** Use other Bluemix services to extend your app to start creating a great Internet of Things app. Here are some of the services you could utilise: Twilio (Third Party), Cloudant NoSQL DB (IBM), Dash DB (IBM), Geospatial Analytics (IBM), Time Series Database, and IBM Analytics for Hadoop.

配置 Bluemix IoT 服务

点击 DEVICES 页面，然后点击 Add Device



配置 Bluemix IoT 服务

输入可选的 Device Type 和 Device ID, 点击 Continue

IBM Internet of Things Foundation Quickstart Service Status Documentation

Organization ID: 94x60j
Bluemix Free (go to Bluemix service)

INFO **DEVICES** ACCESS USAGE

To help you get the IoT Foundation connection software onto your device, visit our [Recipes](#).

Let us know your device type and device ID (for example, the MAC address), so the device can be associated with a selected organization.

Create a device...

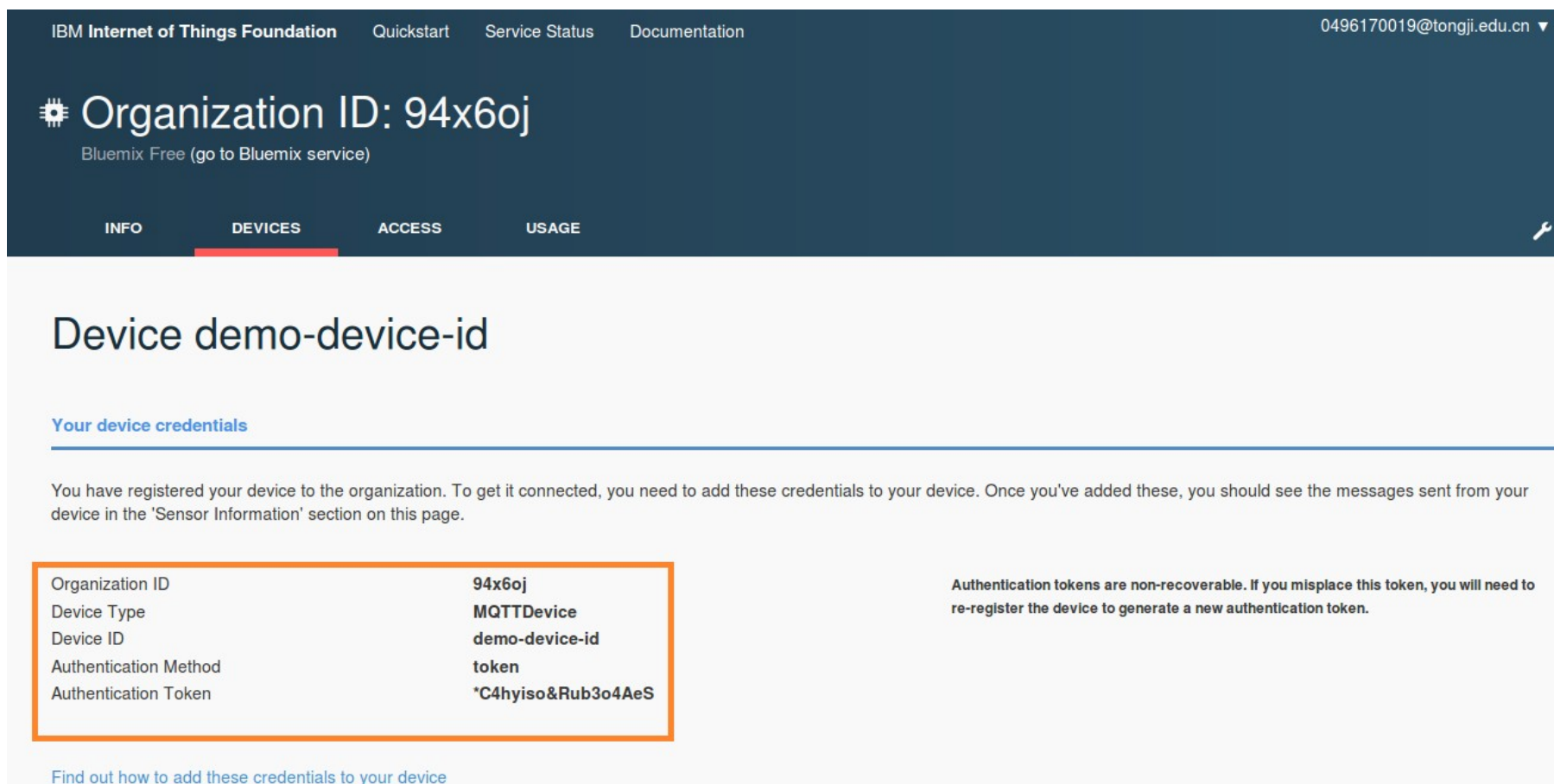
Device Type: MQTTDevice

Device ID: demo-device-id

Continue Cancel

配置 Bluemix IoT 服务

把生成的设备信息保存到文本文件里以备后面使用



The screenshot shows the Bluemix IoT console interface. At the top, there is a navigation bar with links for 'IBM Internet of Things Foundation', 'Quickstart', 'Service Status', and 'Documentation'. The user's email '0496170019@tongji.edu.cn' is displayed in the top right. Below the navigation bar, the organization information is shown: 'Organization ID: 94x6oj' and 'Bluemix Free (go to Bluemix service)'. A menu bar contains 'INFO', 'DEVICES', 'ACCESS', and 'USAGE', with 'DEVICES' highlighted. The main content area is titled 'Device demo-device-id' and includes a section for 'Your device credentials'. A table lists the credentials for the device, with the 'Authentication Token' value highlighted in an orange box. A warning message states that authentication tokens are non-recoverable. A link at the bottom provides instructions on how to add these credentials to a device.

IBM Internet of Things Foundation Quickstart Service Status Documentation 0496170019@tongji.edu.cn ▼

Organization ID: 94x6oj
Bluemix Free (go to Bluemix service)

INFO DEVICES ACCESS USAGE

Device demo-device-id

Your device credentials

You have registered your device to the organization. To get it connected, you need to add these credentials to your device. Once you've added these, you should see the messages sent from your device in the 'Sensor Information' section on this page.

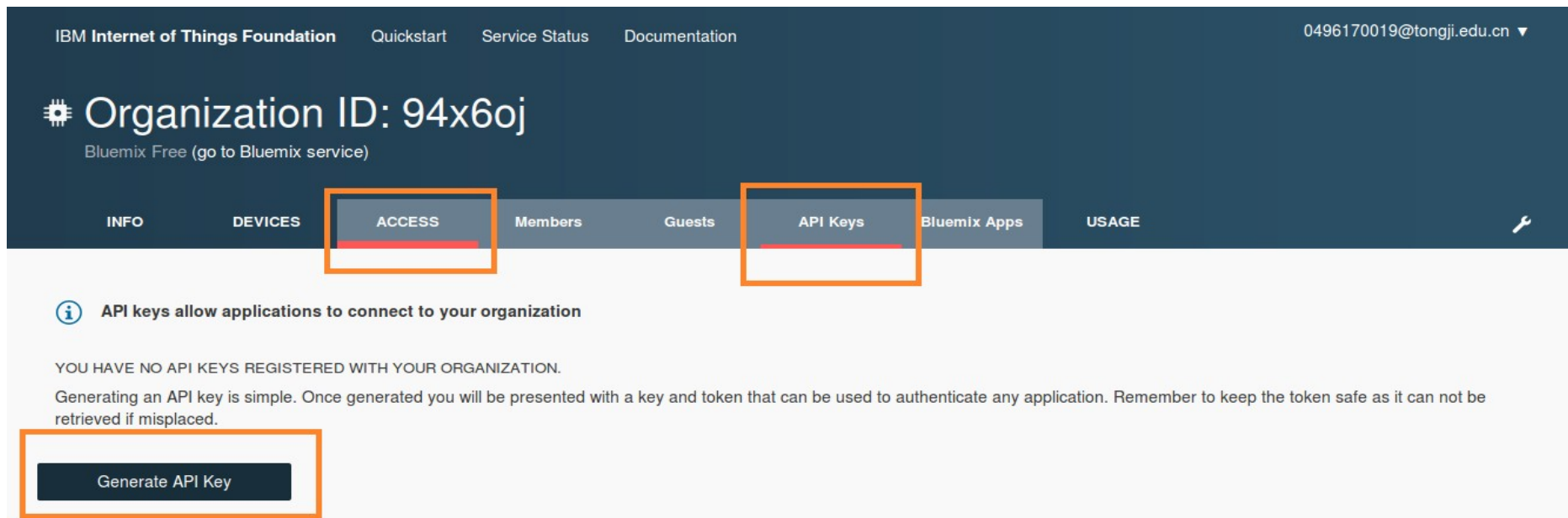
Organization ID	94x6oj
Device Type	MQTTDevice
Device ID	demo-device-id
Authentication Method	token
Authentication Token	*C4hyiso&Rub3o4AeS

Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token.

[Find out how to add these credentials to your device](#)

配置 Bluemix IoT 服务

点击 ACCESS 页面，然后选择 API Keys，点击 Generate API Key



IBM Internet of Things Foundation Quickstart Service Status Documentation 0496170019@tongji.edu.cn ▼

Organization ID: 94x6oj
Bluemix Free (go to Bluemix service)

INFO DEVICES ACCESS Members Guests API Keys Bluemix Apps USAGE

i API keys allow applications to connect to your organization

YOU HAVE NO API KEYS REGISTERED WITH YOUR ORGANIZATION.

Generating an API key is simple. Once generated you will be presented with a key and token that can be used to authenticate any application. Remember to keep the token safe as it can not be retrieved if misplaced.

Generate API Key

配置 Bluemix IoT 服务

输入 Comment, 点击 OK

IBM Internet of Things Foundation Quickstart Service Status Documentation 0496170019@tongji.edu.cn ▼

Organization ID: 94x6oj
Bluemix Free (go to Bluemix service)

INFO DEVICES ACCESS Members Guests API Keys Bluemix Apps USAGE

Generate API Key

[Your API key information](#)

API Key a-94x6oj-mmi9mqt13x

Authentication Token I8L72I0Tqm@2PPYxV

Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token.

Comment

This comment will be visible to all guests and members of this organization and should describe key use.

配置 Bluemix IoT 服务

到此我们得到如下信息：

设备信息：

Organization ID	94x6oj
Device Type	MQTTDevice
Device ID	demo-device-id
Authentication Method	token
Authentication Token	*C4hyiso&Rub3o4AeS

应用信息：

App ID	bluemix-iot-demo
API Key	a-94x6oj-mmi9mqt13x
Authentication Token	l8L72l0Tqrn@2PPYxV

运行示例程序

更改配置文件

MyData/app.conf

#Configuration files for App Side Applications

#The org field is the same org field as the Device side
org=94x6oj

A unique id you choose it by yourself, maybe, abcdefg123456
appid=bluemix-iot-demo

The key field from App Keys info you copied previously
key=a-94x6oj-mmi9mqt13x

The Auth Token field from App Keys info you copied previously
token=l8L72l0Tqrn@2PPYxV

#T or F, T means using SSL, while F means not
isSSL=F

MyData/device.conf

#Configuration files for Device Side Applications

#The org field from Device info you copied previously
org=94x6oj

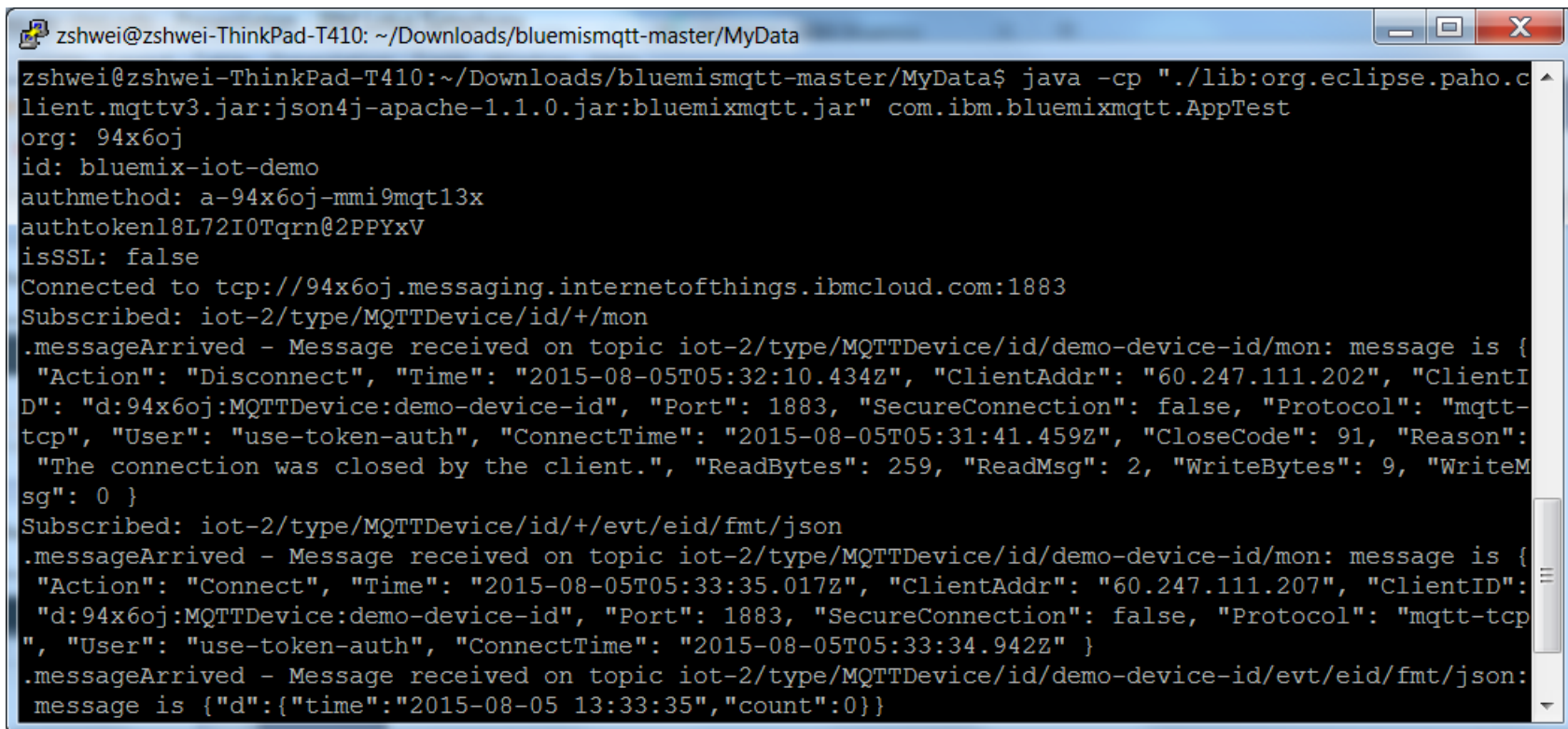
#The id field from Device info you copied previously
deviceid=demo-device-id

#The auth-token field from Device info you copied previously
token=*C4hyiso&Rub3o4AeS

#T or F, T means using SSL, while F means not
isSSL=F

运行示例程序

运行 App 端程序

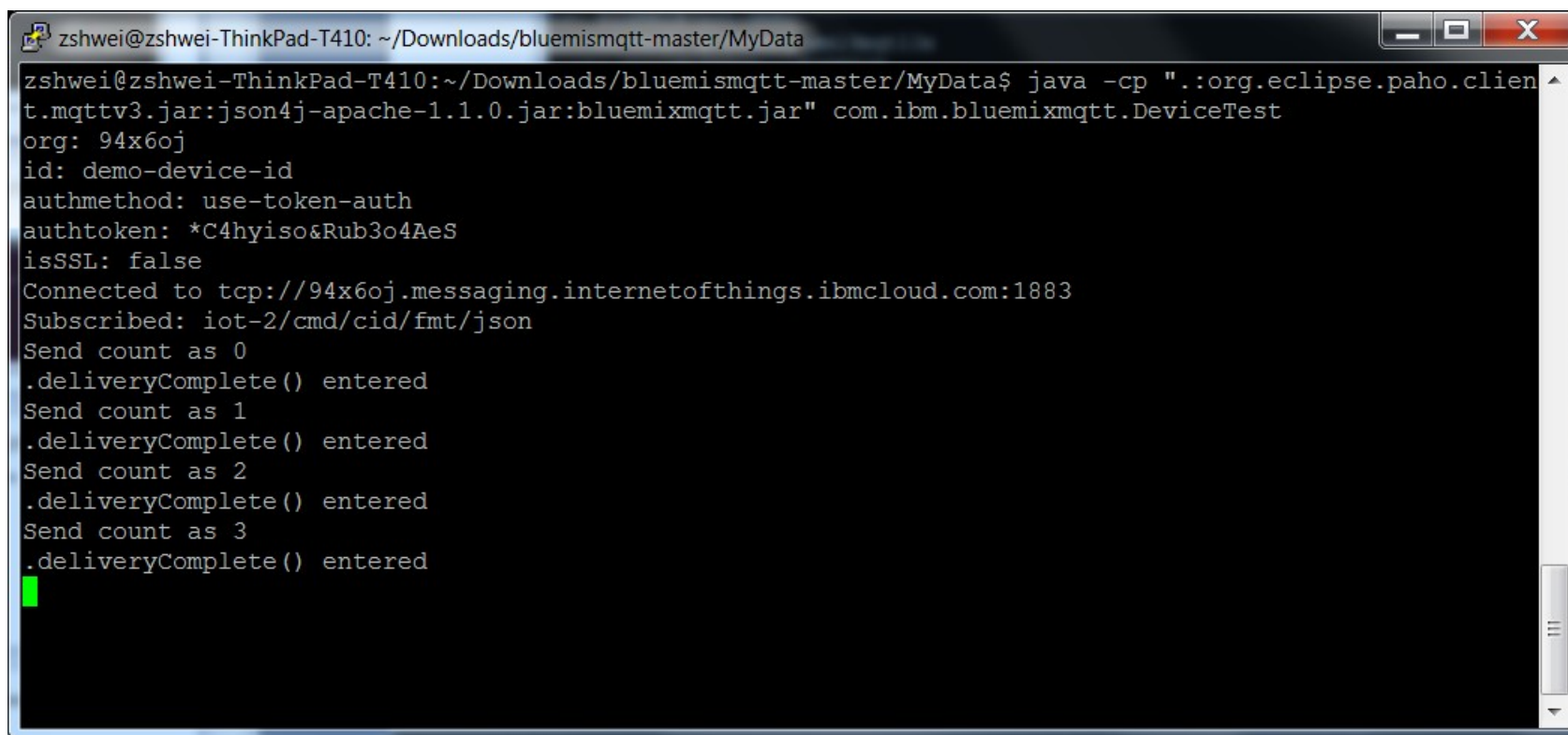


```
zshwei@zshwei-ThinkPad-T410: ~/Downloads/bluemismqtt-master/MyData$ java -cp "./lib:org.eclipse.paho.client.mqttv3.jar:json4j-apache-1.1.0.jar:bluemismqtt.jar" com.ibm.bluemismqtt.AppTest
org: 94x6oj
id: bluemix-iot-demo
authmethod: a-94x6oj-mm19mqt13x
authtoken18L72I0Tqrn@2PPYxV
isSSL: false
Connected to tcp://94x6oj.messaging.internetofthings.ibmcloud.com:1883
Subscribed: iot-2/type/MQTTDevice/id/+/mon
.messageArrived - Message received on topic iot-2/type/MQTTDevice/id/demo-device-id/mon: message is {
  "Action": "Disconnect", "Time": "2015-08-05T05:32:10.434Z", "ClientAddr": "60.247.111.202", "ClientID": "d:94x6oj:MQTTDevice:demo-device-id", "Port": 1883, "SecureConnection": false, "Protocol": "mqtt-tcp", "User": "use-token-auth", "ConnectTime": "2015-08-05T05:31:41.459Z", "CloseCode": 91, "Reason": "The connection was closed by the client.", "ReadBytes": 259, "ReadMsg": 2, "WriteBytes": 9, "WriteMsg": 0 }
Subscribed: iot-2/type/MQTTDevice/id/+/evt/eid/fmt/json
.messageArrived - Message received on topic iot-2/type/MQTTDevice/id/demo-device-id/mon: message is {
  "Action": "Connect", "Time": "2015-08-05T05:33:35.017Z", "ClientAddr": "60.247.111.207", "ClientID": "d:94x6oj:MQTTDevice:demo-device-id", "Port": 1883, "SecureConnection": false, "Protocol": "mqtt-tcp", "User": "use-token-auth", "ConnectTime": "2015-08-05T05:33:34.942Z" }
.messageArrived - Message received on topic iot-2/type/MQTTDevice/id/demo-device-id/evt/eid/fmt/json: message is {"d":{"time":"2015-08-05 13:33:35","count":0}}
```

java -cp ./lib:org.eclipse.paho.client.mqttv3.jar:json4j-apache-1.1.0.jar:bluemismqtt.jar com.ibm.bluemismqtt.AppTest

运行示例程序

运行 Device 端程序

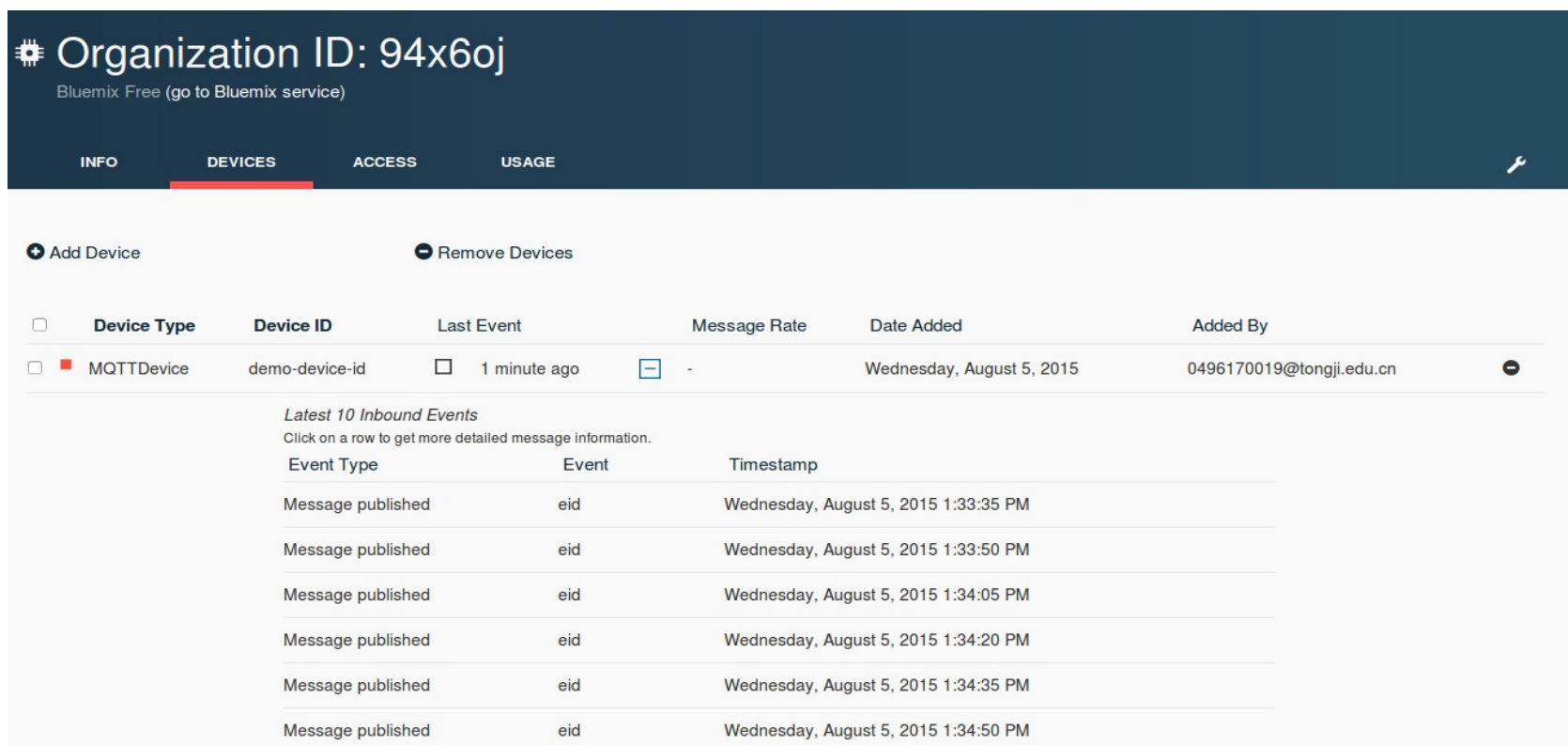


```
zshwei@zshwei-ThinkPad-T410: ~/Downloads/bluemismqtt-master/MyData
zshwei@zshwei-ThinkPad-T410:~/Downloads/bluemismqtt-master/MyData$ java -cp ".:org.eclipse.paho.client.mqttv3.jar;json4j-apache-1.1.0.jar;bluemismqtt.jar" com.ibm.bluemismqtt.DeviceTest
org: 94x6oj
id: demo-device-id
authmethod: use-token-auth
authtoken: *C4hyiso&Rub3o4AeS
isSSL: false
Connected to tcp://94x6oj.messaging.internetofthings.ibmcloud.com:1883
Subscribed: iot-2/cmd/cid/fmt/json
Send count as 0
.deliveryComplete() entered
Send count as 1
.deliveryComplete() entered
Send count as 2
.deliveryComplete() entered
Send count as 3
.deliveryComplete() entered
```

java -cp .;org.eclipse.paho.client.mqttv3.jar;json4j-apache-1.1.0.jar;bluemismqtt.jar com.ibm.bluemismqtt.DeviceTest

运行示例程序

从 dashboard 的 DEVICES 页面查看 event 的历史记录，可以看到该设备每隔 15 秒发送一条 event.



The screenshot displays the IBM Bluemix dashboard for an organization with ID 94x6oj. The 'DEVICES' tab is selected, showing a table of devices. One device, 'MQTTDevice' with ID 'demo-device-id', is listed. Below the table, the 'Latest 10 Inbound Events' are shown, indicating a regular message publishing rate.

Organization ID: 94x6oj
Bluemix Free (go to Bluemix service)

INFO DEVICES ACCESS USAGE

+ Add Device - Remove Devices

<input type="checkbox"/>	Device Type	Device ID	Last Event	Message Rate	Date Added	Added By	
<input type="checkbox"/>	MQTTDevice	demo-device-id	<input type="checkbox"/> 1 minute ago	<input type="checkbox"/> -	Wednesday, August 5, 2015	0496170019@tongji.edu.cn	<input type="checkbox"/>

Latest 10 Inbound Events
Click on a row to get more detailed message information.

Event Type	Event	Timestamp
Message published	eid	Wednesday, August 5, 2015 1:33:35 PM
Message published	eid	Wednesday, August 5, 2015 1:33:50 PM
Message published	eid	Wednesday, August 5, 2015 1:34:05 PM
Message published	eid	Wednesday, August 5, 2015 1:34:20 PM
Message published	eid	Wednesday, August 5, 2015 1:34:35 PM
Message published	eid	Wednesday, August 5, 2015 1:34:50 PM

设备 (Device) 端例程解析

设备端例程主要包括源文件 (java) 和一个配置文件。配置文件里定义了源文件里会使用到的一些参数。源文件按功能可以划分为三部分：

1. 连接 IoT 服务
2. 发送事件 (event) 到 App
3. 订阅由 App 发送的命令 (command)

设备 (Device) 端例程解析

设备端配置文件

```
# MyData/device.conf
```

```
#Configuration files for Device Side Applications
```

```
#The org field from Device info you copied previously  
org=94x6oj
```

```
#The id field from Device info you copied previously  
deviceid=demo-device-id
```

```
#The auth-token field from Device info you copied previously  
token=*C4hyiso&Rub3o4AeS
```

```
#T or F, T means using SSL, while F means not  
isSSL=F
```

设备 (Device) 端例程解析

连接 IoT 服务

```
private MqttHandler handler = null;  
handler = new DeviceMqttHandler();  
handler.connect(serverHost, clientId, authMethod, authToken, isSSL);
```

serverHost

Bluemix IoT 服务可以通过如下两个主机地址访问，分别对应 TCP 和 TLS（Transport Layer Security）的端口，其中的 org-id 是在你注册 Bluemix 应用时产生的组织机构 id.

```
tcp://<org-id>.messaging.internetofthings.ibmcloud.com:1883  
ssl://<org-id>.messaging.internetofthings.ibmcloud.com:8883
```

clientId

格式为 d:<org-id>:<type-id>:<device-id> . 其中 org-id 同上， type-id 和 device-id 分别是之前步骤里注册设备时使用的 id.

authMethod

例程里使用 **use-token-auth**

isSSL

可选 TCP 或者 TSL

设备 (Device) 端例程解析

向 App 发布 event

```
//Format the Json String
JSONObject contObj = new JSONObject();
JSONObject jsonObj = new JSONObject();
try {
    contObj.put("count", count);
    contObj.put("time", new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()));

    jsonObj.put("d", contObj);
} catch (JSONException e1) {
    e1.printStackTrace();
}

System.out.println("Send count as " + count);
//Publish device events to the app
//iot-2/evt/<event-id>/fmt/<format>
handler.publish("iot-2/evt/" + MqttUtil.DEFAULT_EVENT_ID + "/fmt/json", jsonObj.toString(), false, 0);
```

Event 的格式为 `iot-2/evt/<event-id>/fmt/<format>` . 其中 `<event-id>` 用于区分各种不同的 event 类型, 可以按需自行设定. `<format>` 我们设置为 `json` , 因此我们的消息内容需以 `json` 编码. 另外它必须有一个等层的属性 `'d'`.

下面是我们的例程中 event 的一个例子:

```
{
  "d": {
    "count": 3,
    "time": "2014-12-30 16:14:59"
  }
}
```

设备 (Device) 端例程解析

订阅 App 发布的 command

```
//Subscribe the Command events
//iot-2/cmd/<cmd-type>/fmt/<format-id>
handler.subscribe("iot-2/cmd/" + MqttUtil.DEFAULT_CMD_ID + "/fmt/json", 0);
```

Command 的格式为 `iot-2/cmd/<cmd-type>/fmt/<format-id>`。其中 `<cmd-id>` 用于区分各种不同的 command 类型，可以按需自行设定。`<format>` 我们同样设置为 `json`。当有 command 收到后一个 callback 的函数 `messageArrived` 会被执行。这里我们根据收到的命令里的值设置设备的计数器。

```
public void messageArrived(String topic, MqttMessage mqttMessage) throws Exception
{
    super.messageArrived(topic, mqttMessage);
    //Check whether the event is a command event from app
    if (topic.equals("iot-2/cmd/" + MqttUtil.DEFAULT_CMD_ID + "/fmt/json"))
    {
        String payload = new String(mqttMessage.getPayload());
        JSONObject jsonObject = new JSONObject(payload);
        String cmd = jsonObject.getString("cmd");
        //Reset the count
        if (cmd != null && cmd.equals("reset"))
        {
            int resetcount = jsonObject.getInt("count");
            count = resetcount;
            System.out.println("Count is reset to " + resetcount);
        }
    }
}
```

应用端 (App) 例程解析

跟设备端例程类似，应用端例程也包括源文件 (java) 和一个配置文件。配置文件里定义了应用端源文件里会使用到的一些参数。应用端源文件按功能可以划分为三部分：

1. 连接 IoT 服务
2. 订阅由设备端发送的事件 (event)
3. 发送命令 (command) 到设备端

应用端 (App) 例程解析

配置文件:

```
# MyData/app.conf
```

```
#Configuration files for App Side Applications
```

```
#The org field is the same org field as the Device side  
org=94x6oj
```

```
# A unique id you choose it by yourself, maybe, abcdefg123456  
appid=bluemix-iot-demo
```

```
# The key field from App Keys info you copied previously  
key=a-94x6oj-mmi9mqt13x
```

```
# The Auth Token field from App Keys info you copied previously  
token=l8L72l0Tqrn@2PPYxV
```

```
#T or F, T means using SSL, while F means not  
isSSL=F
```


应用端 (App) 例程解析

连接 IoT 服务

```
handler = new AppMqttHandler();  
handler.connect(serverHost, clientId, authMethod, authToken, isSSL);
```

serverHost

<org-id>.messaging.internetofthings.ibmcloud.com

其中 org-id 为注册 IoT 服务时产生的 id

ClientId

格式为 a:<org-id>:<app-id> . 其中 org-id 同上, app-id 是之前步骤里注册时用户输入的 id.

authMethod

之前步骤里生成的 key

authToken

之前步骤里生成的 token

IsSSL

可选 TCP 或者 TSL

应用端 (App) 例程解析

订阅设备发布的 event

应用端可以订阅两种类型的 event, 设备端的 event, 以及系统监测信息例如设备与 IoT 服务的连接与断开。

订阅设备端的 event 的格式为 `iot-2/type/<type-id>/id/<device-id>/evt/<event-id>/fmt/<format-id>`. 此格式需要和设备端发送 event 的格式一致。订阅系统监测信息格式为 `iot-2/type/<type-id>/id/<device-id>/mon`。格式里可以使用 + 作为通配符来匹配一类设备。

```
handler.subscribe("iot-2/type/" + MqttUtil.DEFAULT_DEVICE_TYPE + "/id+/mon", 0);
//Subscribe Device Events
//iot-2/type/<type-id>/id/<device-id>/evt/<event-id>/fmt/<format-id>
handler.subscribe("iot-2/type/" + MqttUtil.DEFAULT_DEVICE_TYPE + "/id+/evt/" + MqttUtil.DEFAULT_EVENT_ID + "/fmt/json", 0);
```

当有 event 收到时一个 callback 的函数 `messageArrived` 会被执行。这里我们检查收到的 event 里的一个计数器, 如果到达 4 就发出一个 command 通知设备重置为 0.

```
public void messageArrived(String topic, MqttMessage mqttMessage) throws Exception {
    super.messageArrived(topic, mqttMessage);
    Matcher matcher = pattern.matcher(topic);
    if (matcher.matches()) {
        String deviceid = matcher.group(1);
        String payload = new String(mqttMessage.getPayload());
        //Parse the payload in Json Format
        JSONObject jsonObject = new JSONObject(payload);
        JSONObject contObj = jsonObject.getJSONObject("d");
        int count = contObj.getInt("count");
        System.out.println("Receive count " + count + " from device " + deviceid);
        //If count >=4, start a new thread to reset the count
        if (count >= 4) {
            new ResetCountThread(deviceid, 0).start();
        }
    }
}
```

应用端 (App) 例程解析

向设备发布 command

```
JSONObject jsonObj = new JSONObject();
try {
    jsonObj.put("cmd", "reset");
    jsonObj.put("count", count);
    jsonObj.put("time", new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()));
} catch (JSONException e) {
    e.printStackTrace();
}
System.out.println("Reset count for device " + deviceid);
//Publish command to one specific device
//iot-2/type/<type-id>/id/<device-id>/cmd/<cmd-id>/fmt/<format-id>
handler.publish("iot-2/type/" + MqttUtil.DEFAULT_DEVICE_TYPE
+ "/id/" + deviceid + "/cmd/" + MqttUtil.DEFAULT_CMD_ID
+ "/fmt/json", jsonObj.toString(), false, 0);
```

command 的格式为 `iot-2/cmd/<cmd-type>/fmt/<format-id>` 其中 `<cmd-type>` 用于区分各种不同的 command 类型，可以按需自行设定。`<format>` 我们设置为 `json`，因此我们的 command 需以 `json` 编码。

资源连接

IoT 文档库

<https://docs.internetofthings.ibmcloud.com/#/>

MQTT 主页

<http://mqtt.org/>

使用 Arduino Uno 和 IBM IoT Foundation 构建云就绪的温度传感器

<http://www.ibm.com/developerworks/cn/cloud/library/cl-bluemix-arduino-iot1/index.html>

使用 IBM IoT Foundation 和 IBM Bluemix 构建自己的可穿戴设备应用程序

<http://www.ibm.com/developerworks/cn/data/library/ba/ba-bluemix-diy-iot-wearable-app/>

使用 Bluemix 提供的 IoT 和 GPS 服务提高您的健康水平

<http://www.ibm.com/developerworks/cn/cloud/library/cl-getmoarsteps-app/>

使用 Geospatial Analytics 构建互联车辆 IoT 应用程序

<http://www.ibm.com/developerworks/cn/mobile/mo-connectedcar-app/index.html>

为您的 IBM Bluemix 应用程序构建一个可以远程控制 Raspberry Pi 2 监视器

<http://www.ibm.com/developerworks/cn/cloud/library/cl-raspberrypi-iot-remote-monitoring-app/>