REYAX
TECHNOLOGY CORPORATION, LTD

4-DEC-2020 56312E31

RYC1001_Application_note

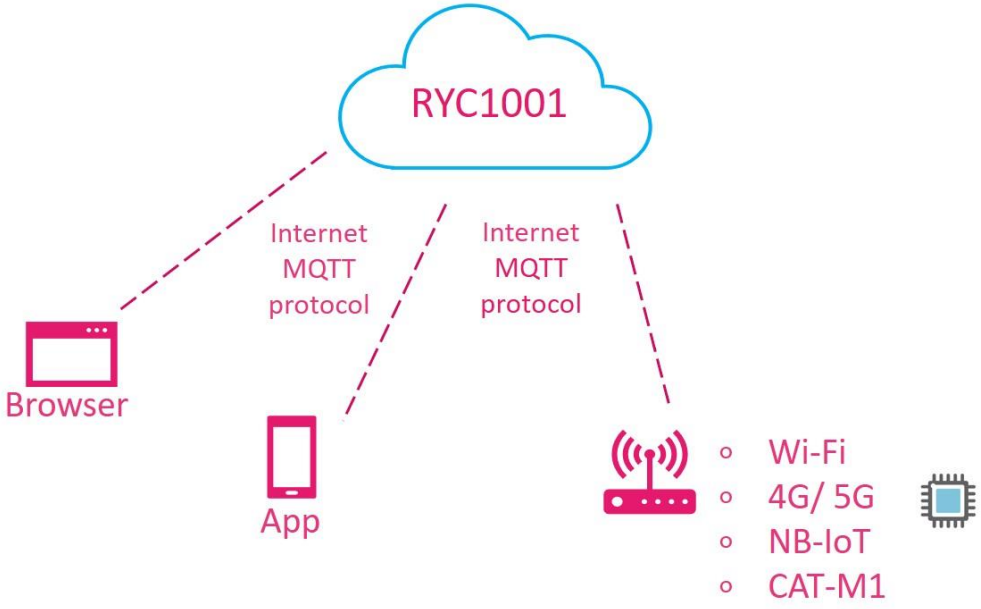
Windows MQTT.FX (PC demo)

Preface

RYC1001 is a cloud service built by REYAX Technology, and it is especially for the Internet of Things (IoT).

Through the MQTT communication protocol, the messages can be transmitted and well-received between user and device.

Besides, there are two types of messages running on REYAX IoT Cloud currently, namely **Command** and **Notification**.



The Purpose of the Command

Transmit the messages by commanding that users can achieve the following purposes:

- Publish a command to control a machine and check the execution result. (For example, the user sends the command to turn on the electric fan, and the electric fan responds "OK, it's turned on.")

- To storage the messages. (Like the memory concept, the messages stored in the memory zone can be continuously updated.)

The format of the command can be divided into Command and Command update. The Command update is it can only update the messages that currently exist in the Command.

Command update can also be used as a two-way acknowledgment (ACK) of the Command.

The Purpose of the Notification

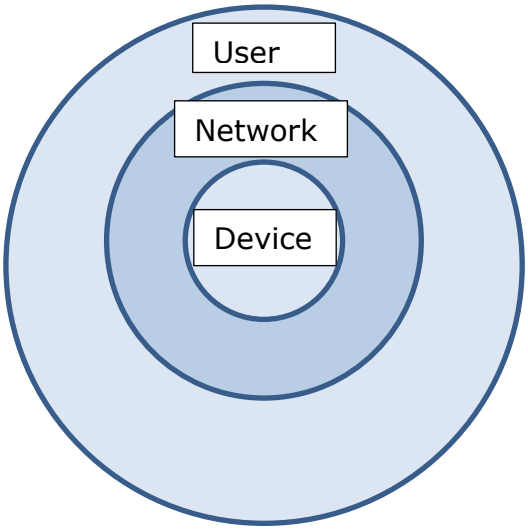
◆ Upload the data report. (For example, the data obtained by the device monitoring a sensor for a long time can be uploaded at a fixed time.)

◆ Report the status of the device. (For example, the status of a fan may show as “the fan is currently turned on.”)

The Internet Access Authority of REYAX IoT Cloud

The user will have an exclusive Network ID when starting the cloud service and will have an exclusive Device ID in the network.

The users can only see the device messages that belong to their own network.



Although both the user and the device can transmit Command and Notification.

But the logic in general use is that the Command is transmitted by the user mostly, and updated by the device after operating. Or transmitted the Notification by the device at a fixed time.

RYC1001 Example Account

<URL>	iot.reyax.com
<Username>	zFtbPwRwxE
<Password>	GhMshMhWG9
<Network_ID>	28
<Device_ID>	xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0

Windows MQTT demo software : MQTT.FX

Install MQTT software. (We use the MQTT.FX for the example.)

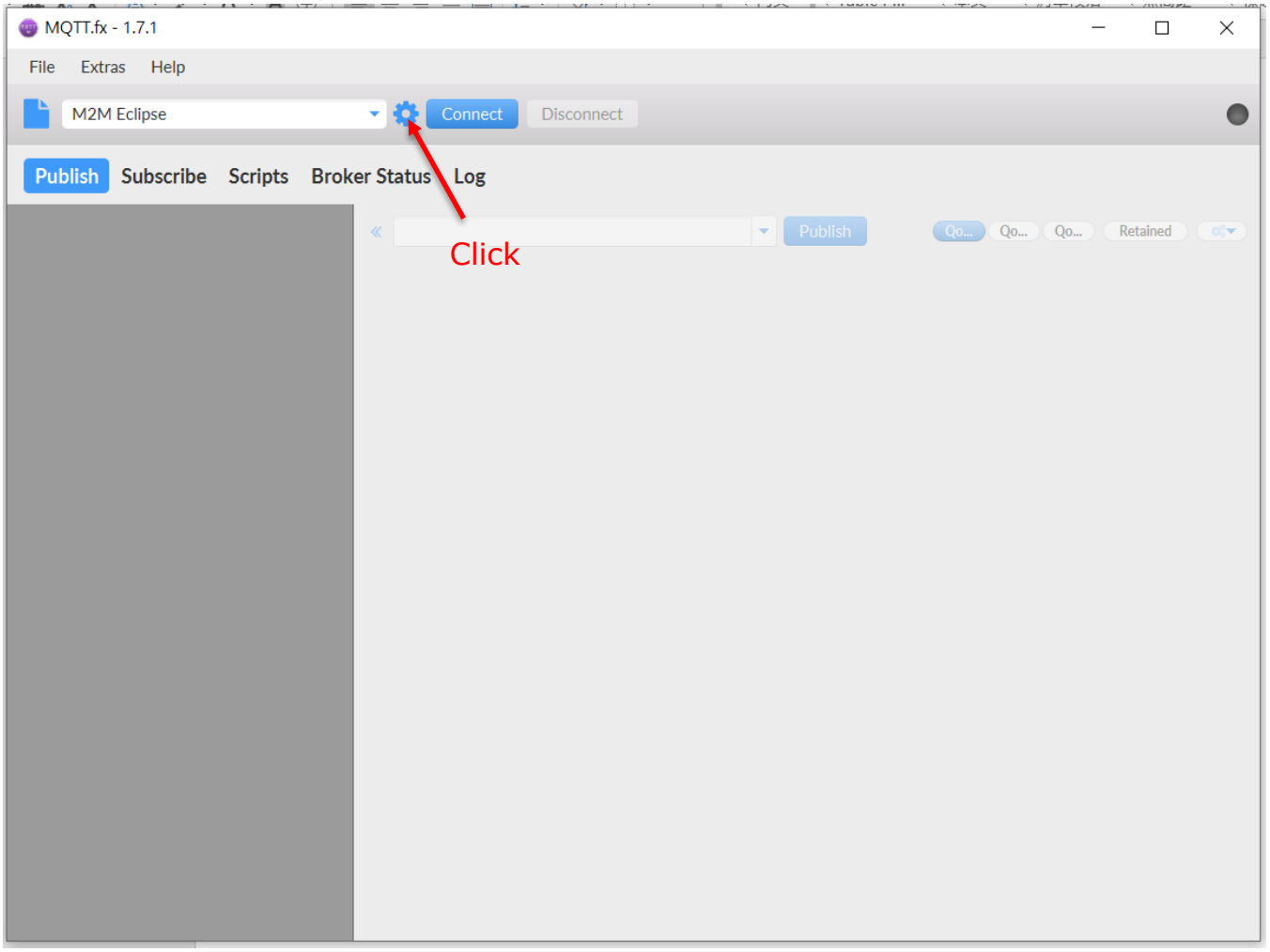
The link: https://mqttfx.jensd.de/index.php/download Click “Latest Release” and download the file, for example: If your computer is a window system, please choose the file “mqttfx-1.7.1-windows-x64”.

The MQTT Manual of REYAX IoT Cloud

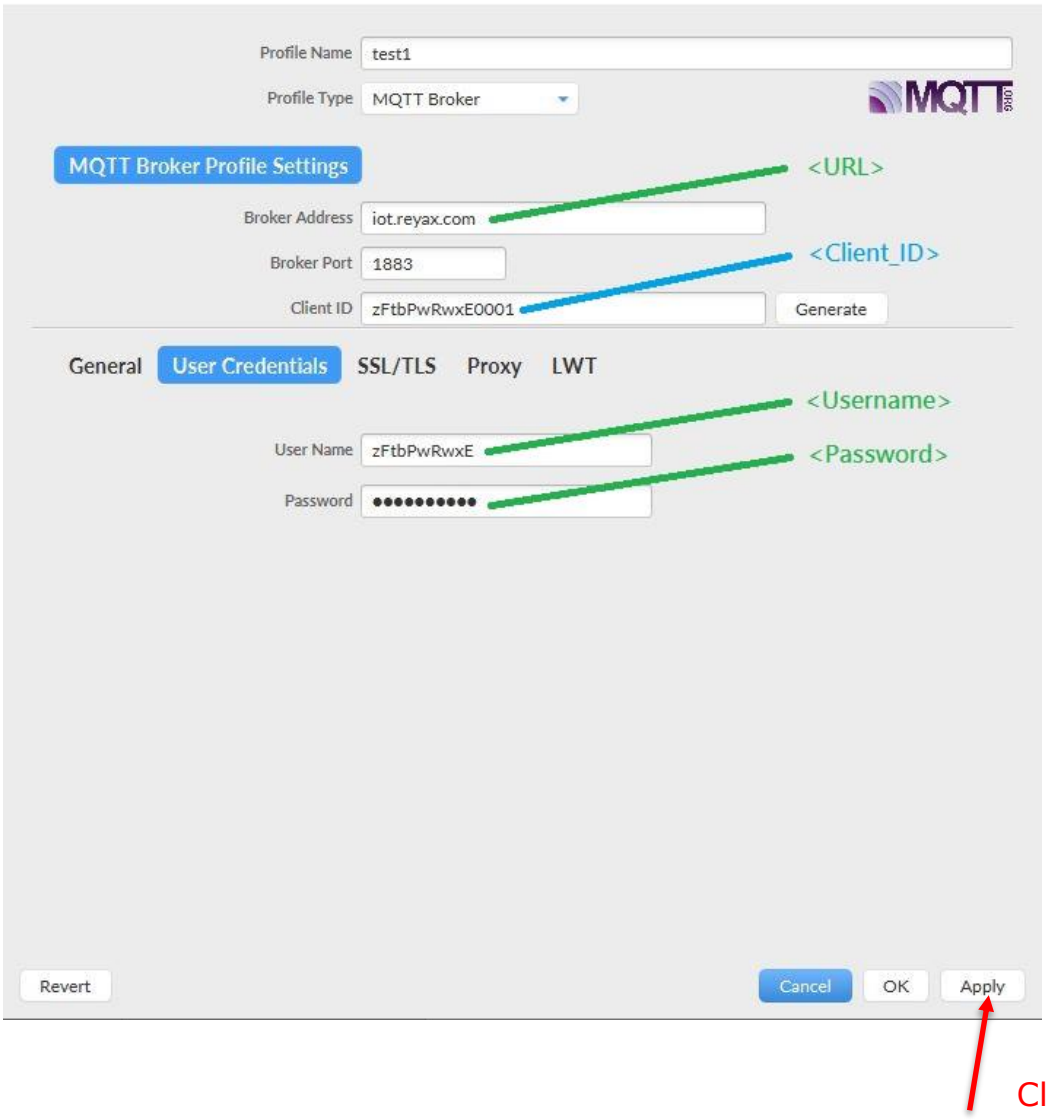
In this manual, we will use the MQTT application on the PC to demo the transmission between the user and the device. But in the practical application of IoT, it can transmit between different devices.

◆ **If you want to execute “Subscribe” and “Publish” in Windows at the same time, please execute two Windows programs.**

1. MQTT.FX software setup



Setup MQTT server information



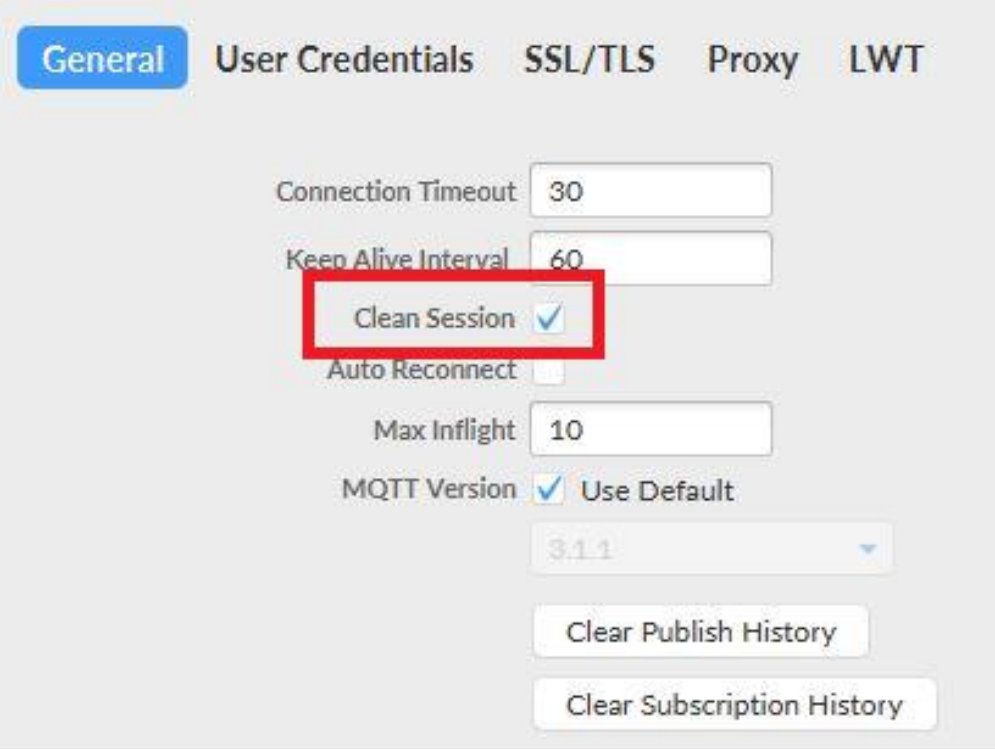
Broker Address (URL): iot.reyax.com (port:1883)
Client ID (Client_ID): Username+xxxx (any 4-digit number)

Example:
zFtbPwRwxE0001 (The first device.)
zFtbPwRwxE0002 (The second device.)

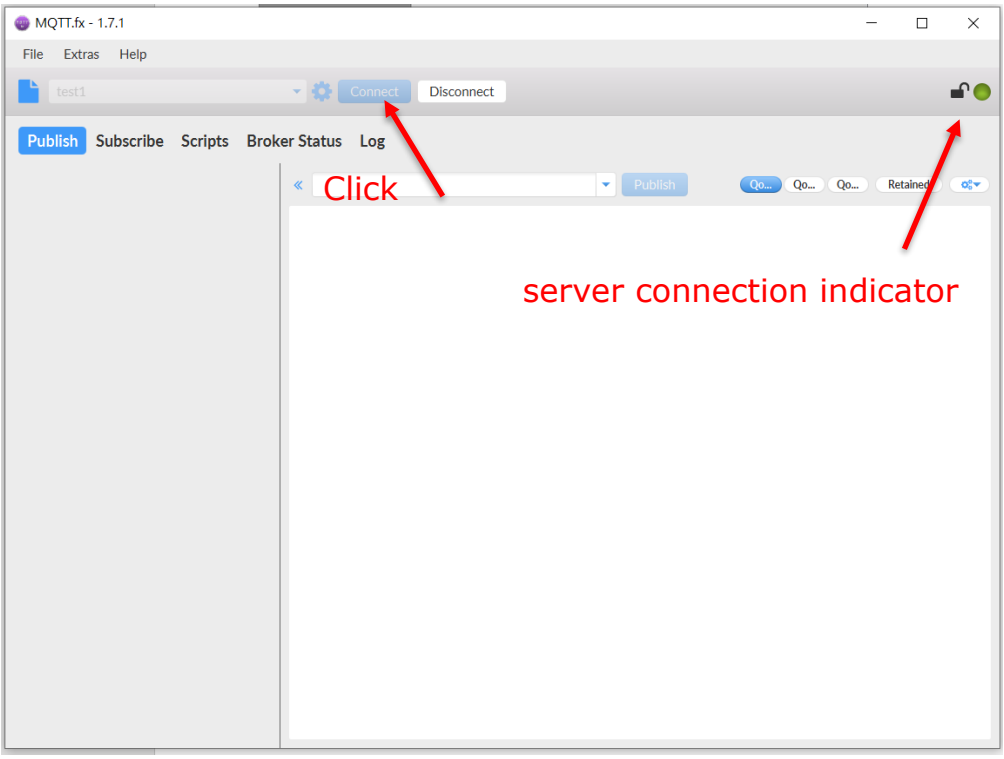
Notice: The Client ID of each device needs to be unique and can't repeat.

User Name (Username) & password (Password): Provide by REYAX Technology.

General:

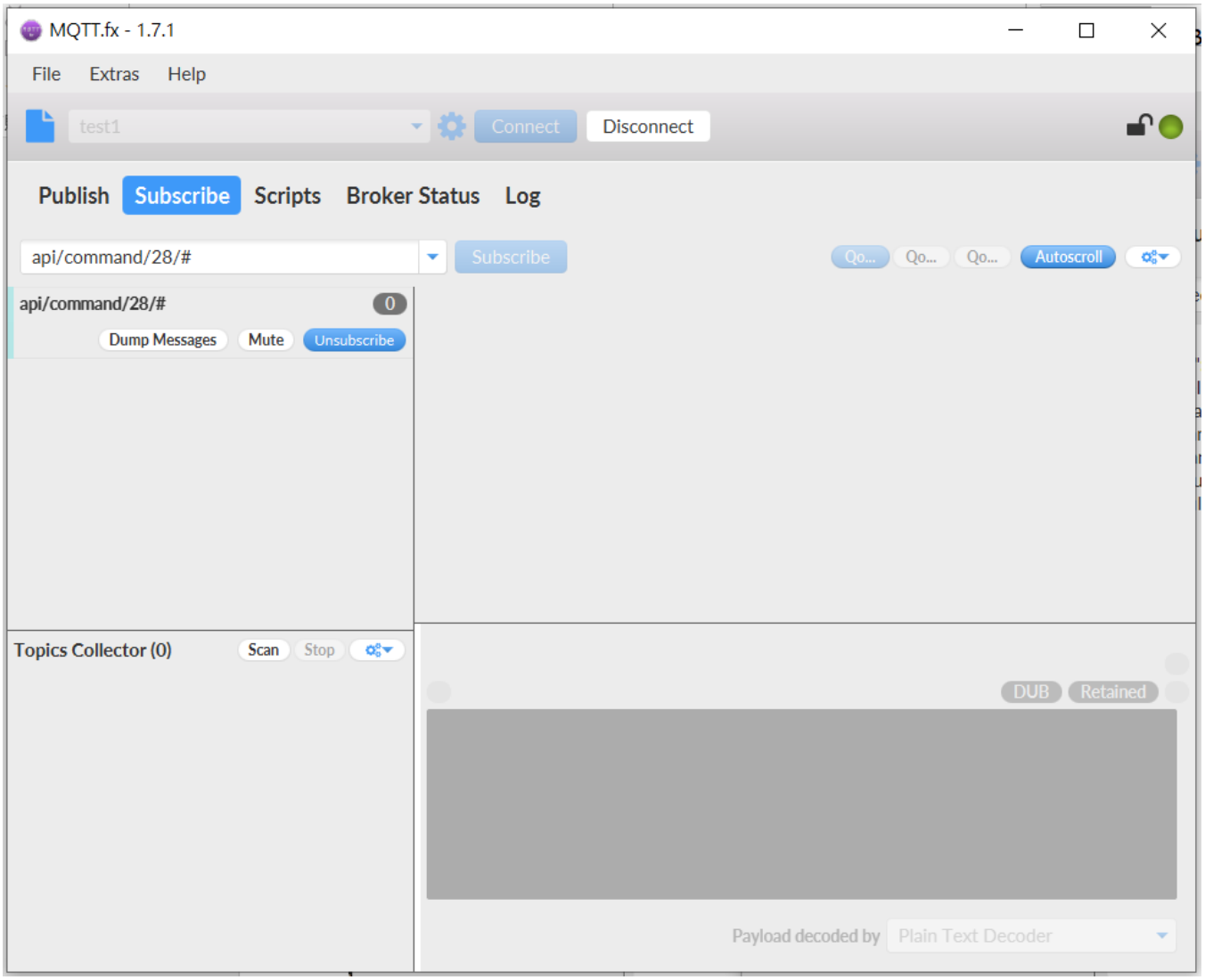


Back to the main page, and click "connect".



2.1 Subscribe command message

Subscribe to the "api/command/<Network_ID>/#" topic:



2.2 Publish command message

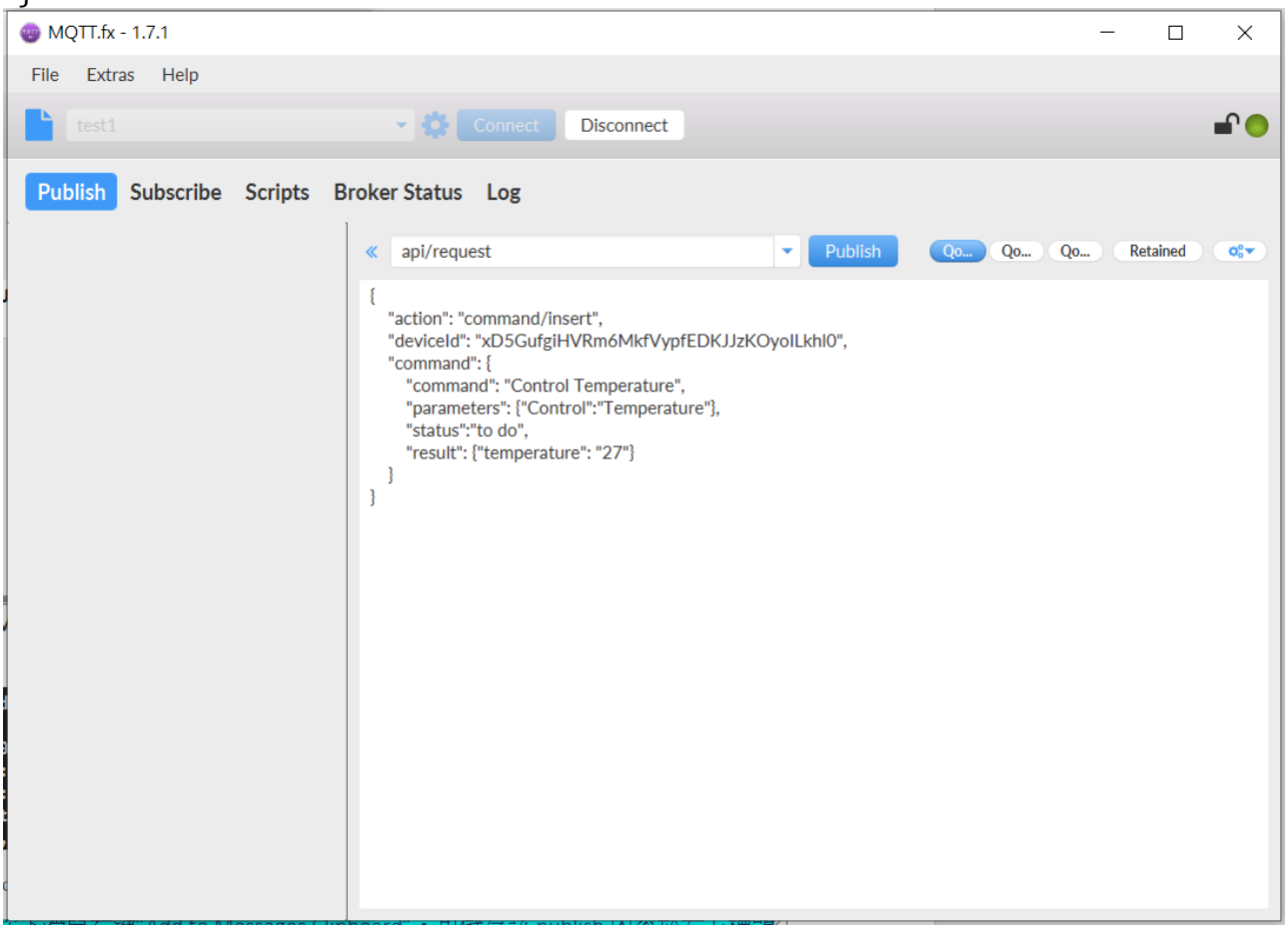
Publish the command message to the "api/request" topic:

The command format is as follows:

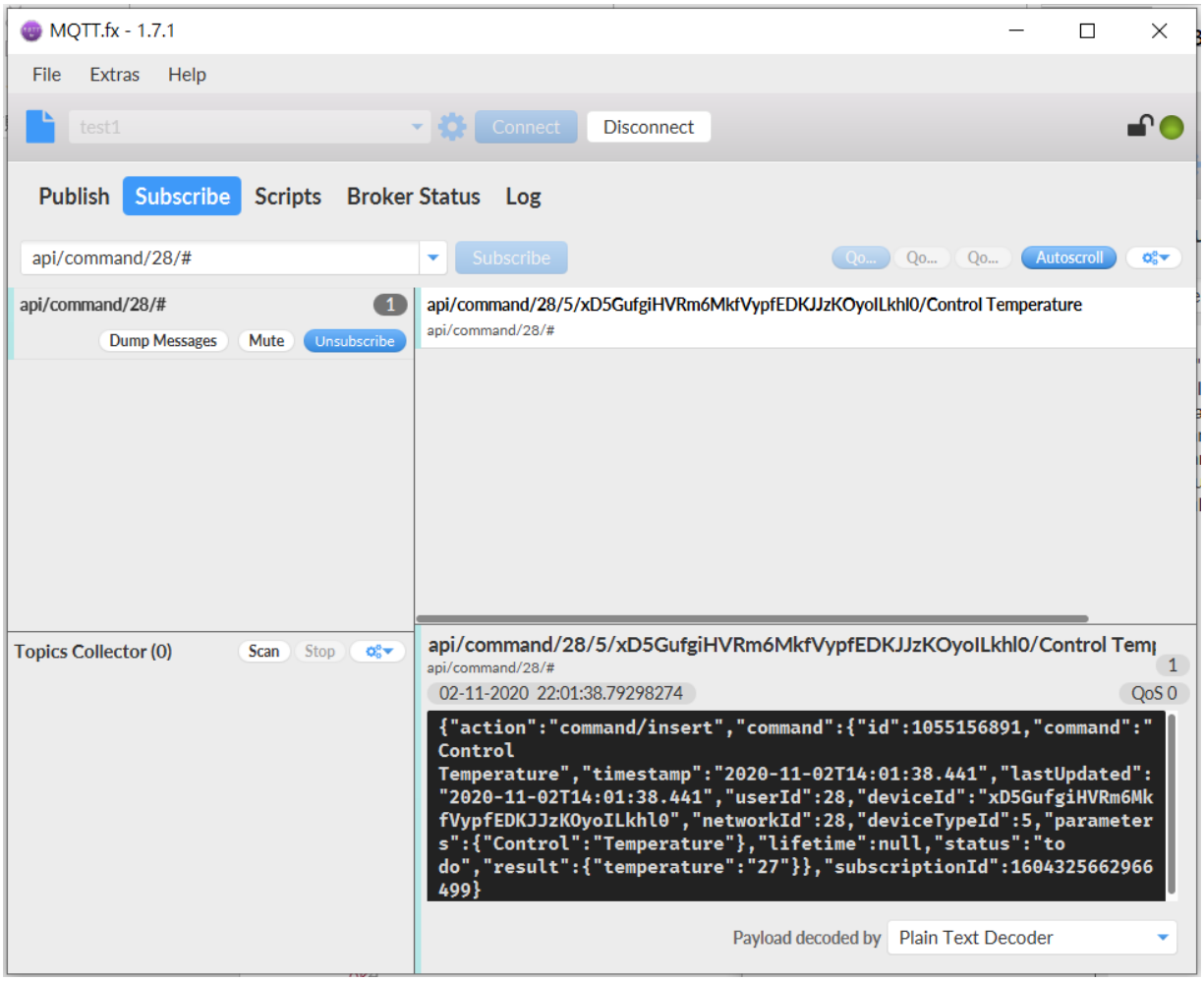
```
{"action":"command/insert","deviceId":"<Device_ID>","command":{"command":"<Com
    mand_Name>","parameters":{"<Parameter1>":"<Parameter2>"},"status":"<Status
    >", "result":{"<Result1>":"<Result2>"}}}
```

EX:

```
{
   "action": "command/insert",
   "deviceId": "xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0",
   "command": {
      "command": "Control Temperature",
      "parameters": {"Control":"Temperature"},
      "status":"to do",
      "result": {"temperature": "27"}
   }
}
```



2.3 Received command message



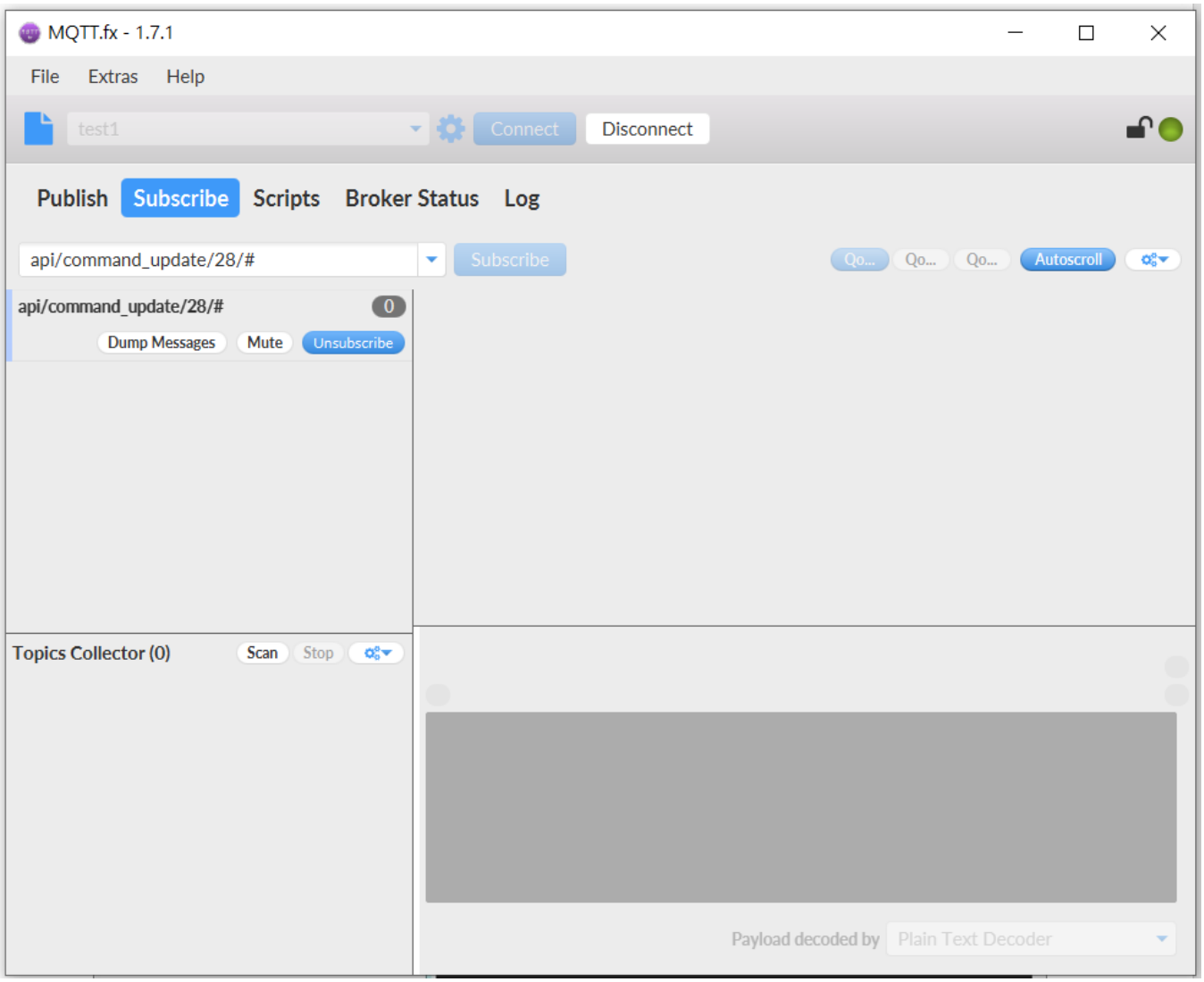
When Publish is executed, the received format is as follows:

{"action":"command/insert","command":{"id":<commandId>,"command":"<Command_Name>","timestamp":"<Time_Stamp>","lastUpdated":"<Time_Stamp>","userId":6,"deviceId":"<Device_ID>","networkId":6,"deviceTypeId":4,"parameters":{"<Parameter1>":"<Parameter2>"},"lifetime":null,"status":"<Status>","result":{"<Result1>":"<Result2>"}},"subscriptionId":<subscriptionId>}

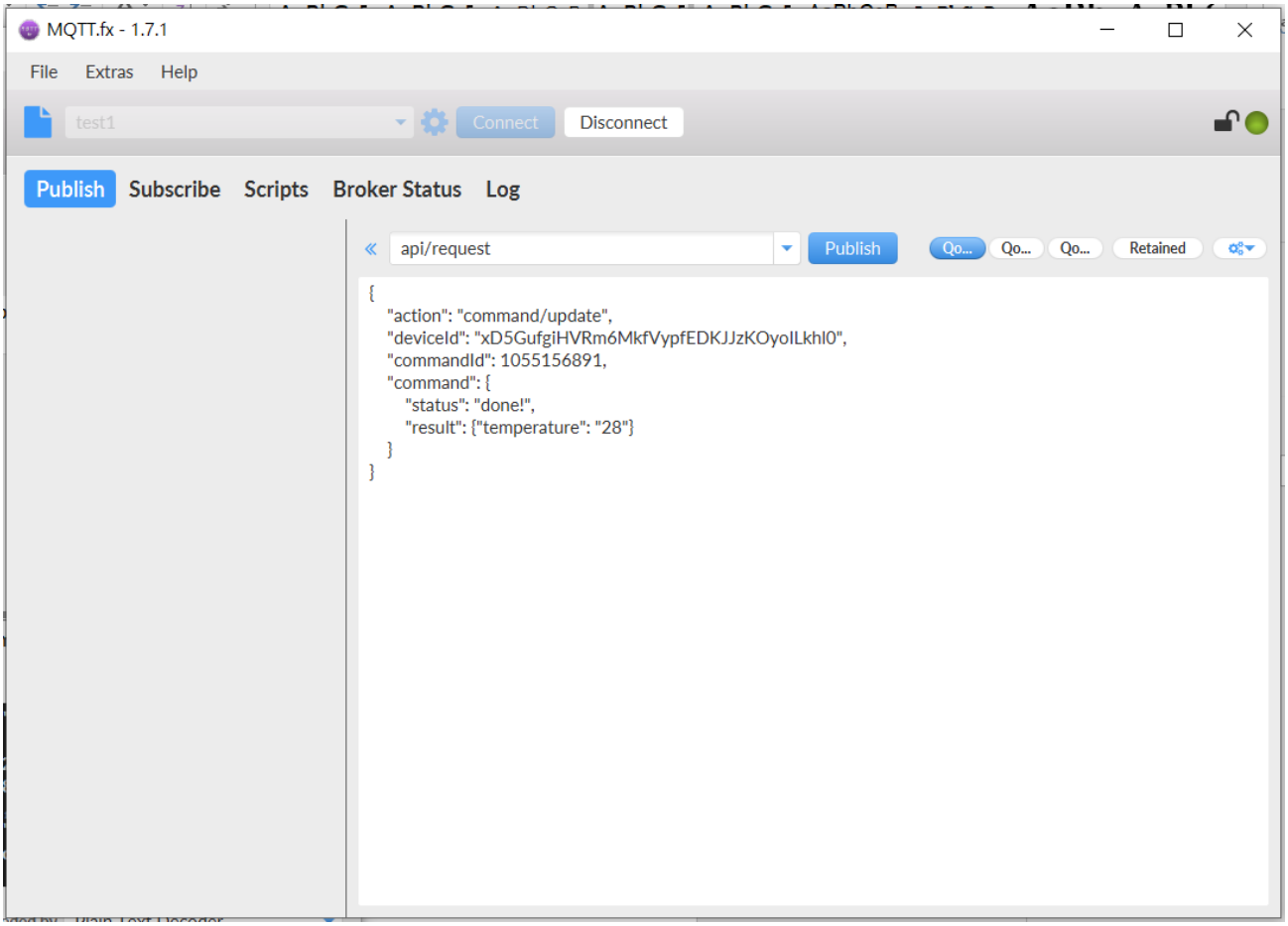
EX:
{"action":"command/insert","command":{"id":1055156891,"command":"Control Temperature","timestamp":"2020-11-02T14:01:38.441","lastUpdated":"2020-11-02T14:01:38.441","userId":28,"deviceId":"xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0","networkId":28,"deviceTypeId":5,"parameters":{"Control":"Temperature"},"lifetime":null,"status":"to do","result":{"temperature":"27"}},"subscriptionId":1604325662966499}

3.1 Subscribe command_update message

Subscribe to the topic "api/command_update/<Network_ID>/#":



3.2 Publish command_update message



Publish the command_update message to the "api/request" topic: The format of the command_update message is defined by this platform. The message will be stored in the database of this platform.

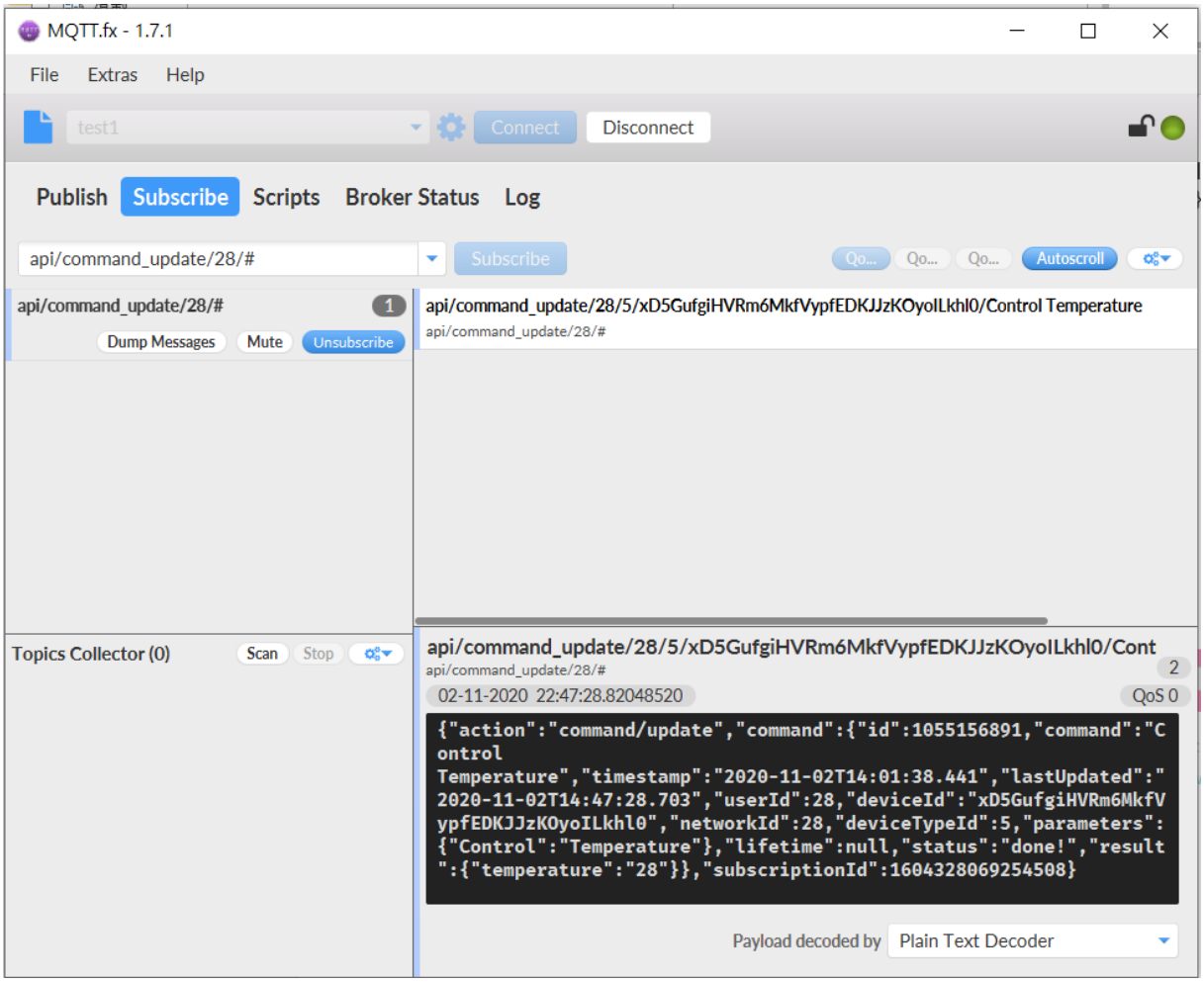
The command format is as follows:

{"action":"command/update","deviceId":"<Device_ID>","commandId":<commandId>,"command":{"status":"<Status>","result":"<Result1>":"<Result2>"}}

Check your commandId from your screen then replaced the below Id with yours. EX:

```
{
   "action": "command/update",
   "deviceId": "xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0",
   "commandId": 1055156891,
   "command": {
      "status": "done!",
      "result": {"temperature": "28"}
   }
}
```

3.3 Received command_update message



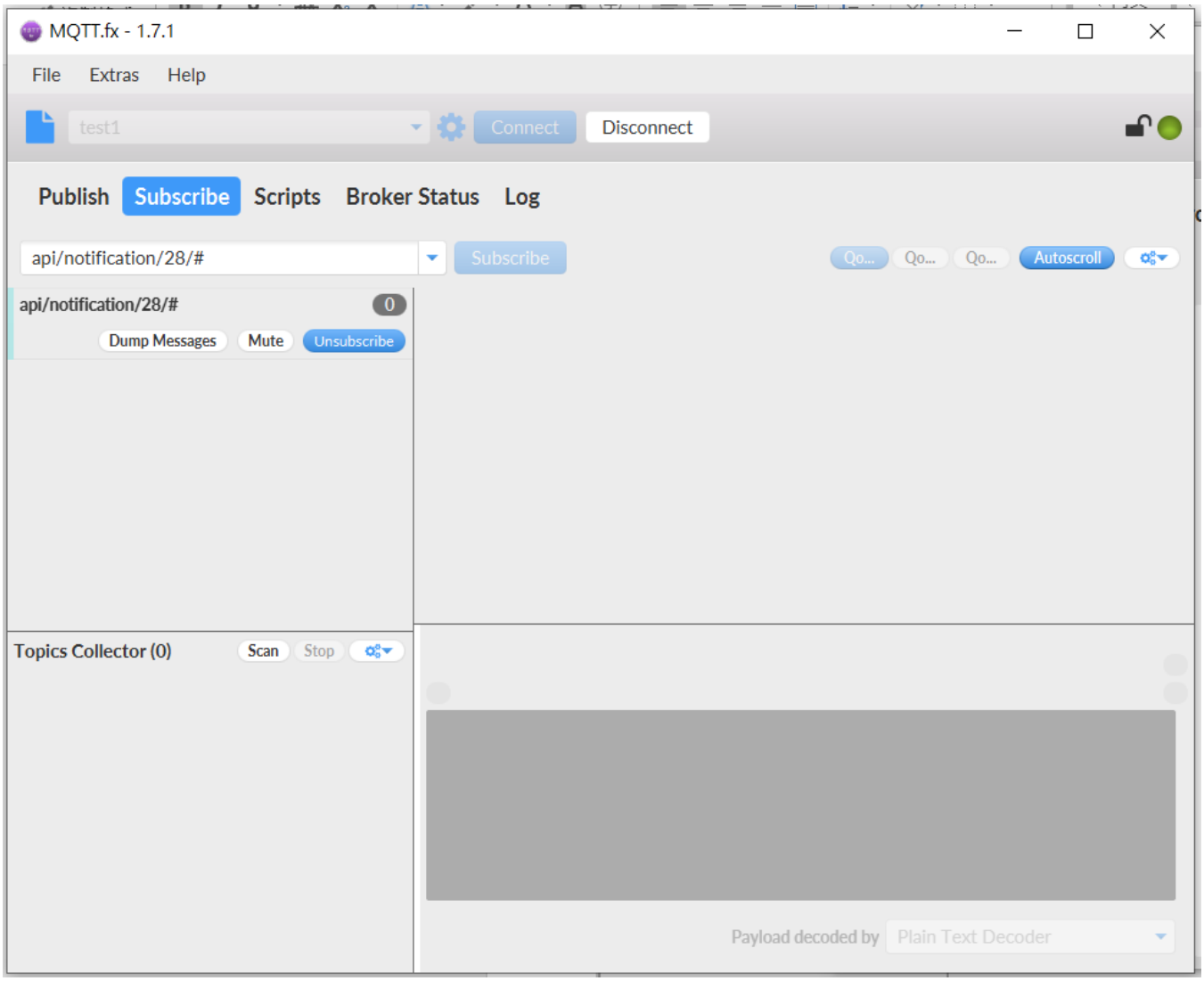
When Publish is executed, the received format is as follows:

{"action":"command/update","command":{"id":<commandId>,"command":"<Command_Name>","timestamp":"<Time_Stamp>","lastUpdated":"<Time_Stamp>","userId":6,"deviceId":"<Device_ID>","networkId":6,"deviceTypeId":4,"parameters":{"<Parameter1>":"<Parameter2>"},"lifetime":null,"status":"Done","result":{"<Result1>":"<Result2>"}},"subscriptionId":<subscriptionId>}

EX:
{"action":"command/update","command":{"id":1055156891,"command":"Control Temperature","timestamp":"2020-11-02T14:01:38.441","lastUpdated":"2020-11-02T14:47:28.703","userId":28,"deviceId":"xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0","networkId":28,"deviceTypeId":5,"parameters":{"Control":"Temperature"},"lifetime":null,"status":"done!","result":{"temperature":"28"}},"subscriptionId":1604328069254508}

4.1 Subscribe notification message

Subscribe to the topic "api/notification/<Network_ID>/#":



4.2 Publish notification message

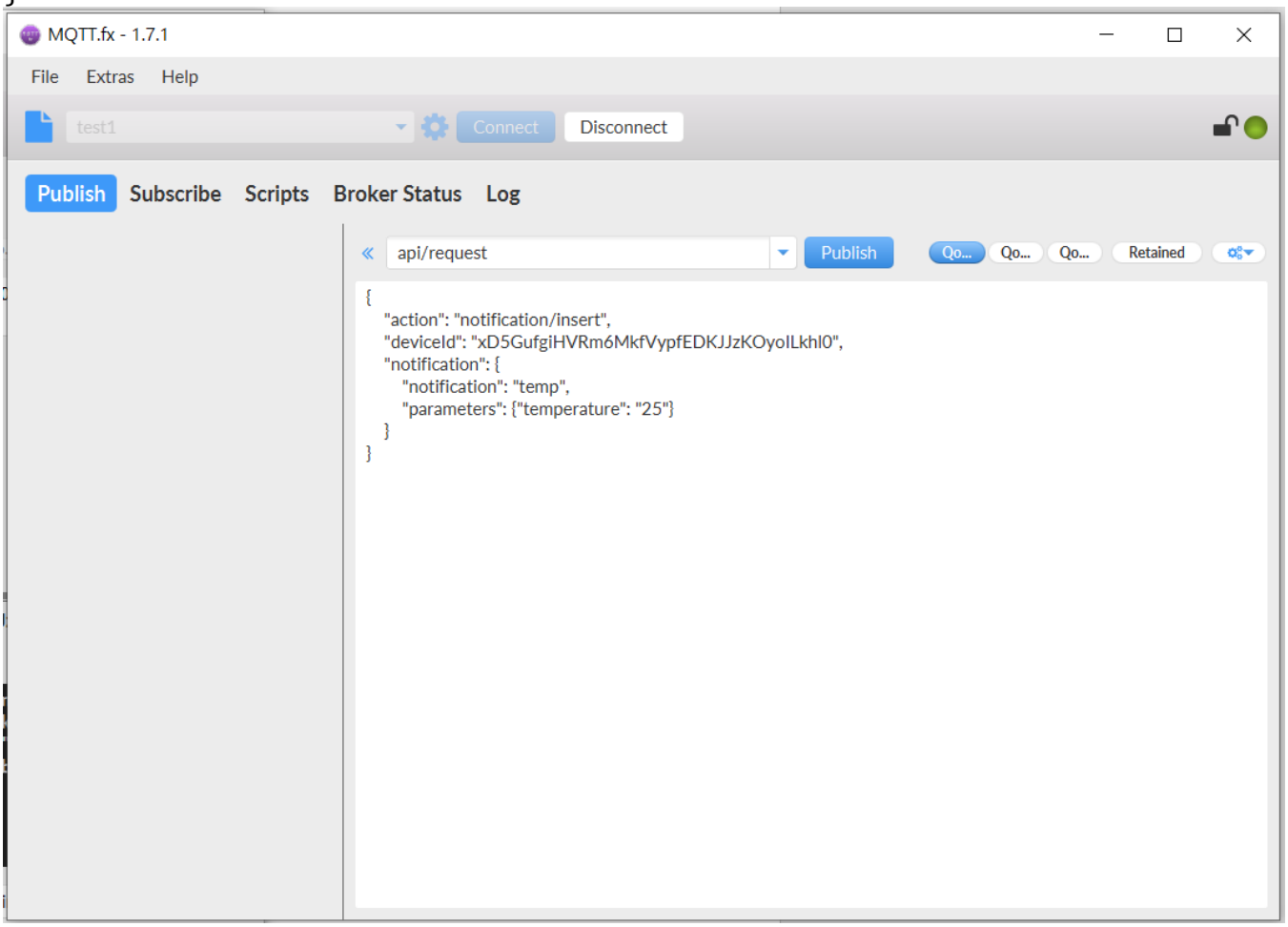
Publish notification messages to the "api/request" topic: The format and content of notification messages are defined by this platform. The message will be stored in the database of this platform.

The command format is as follows:

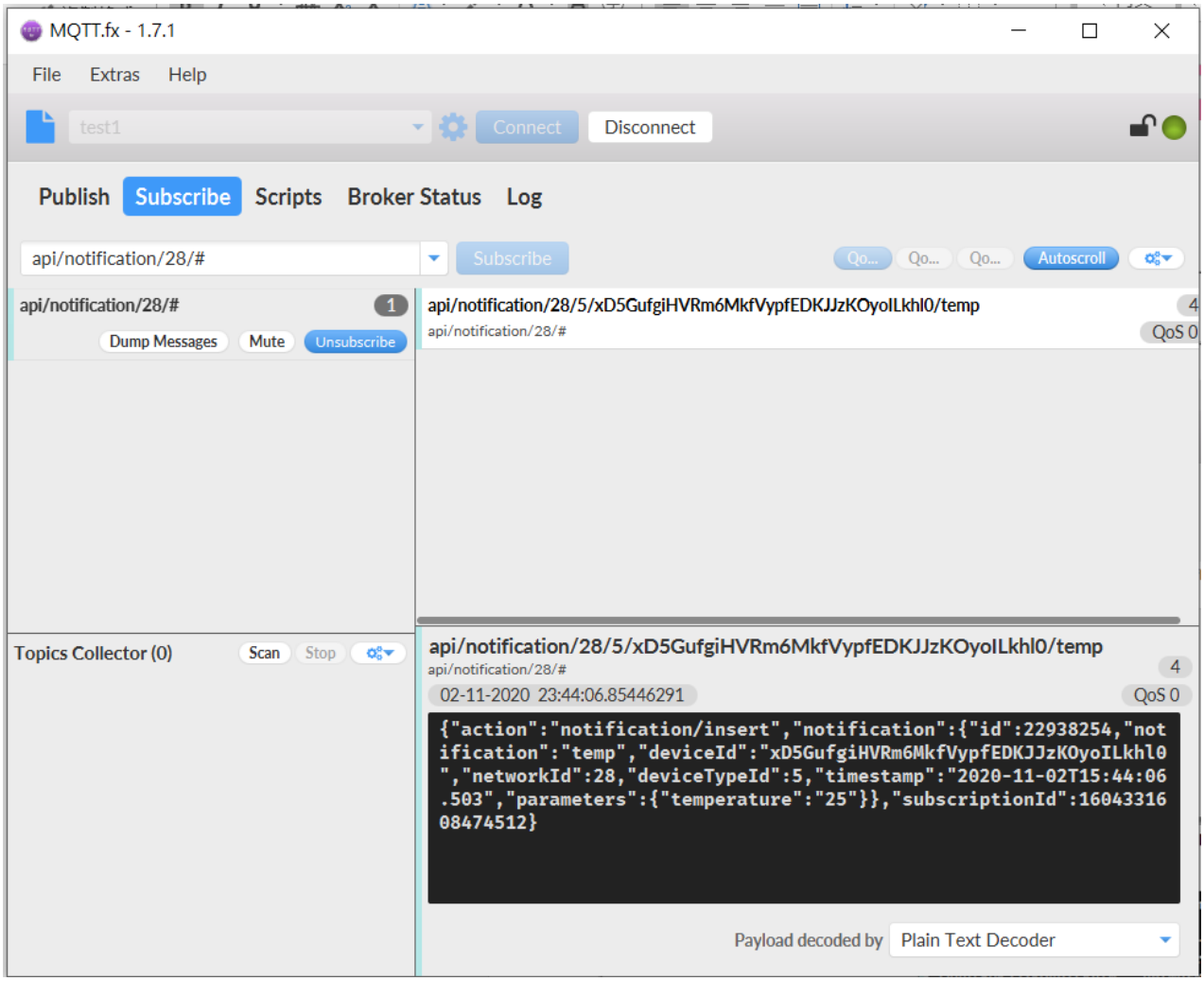
{"action":"notification/insert", "deviceId": "<Device_ID>","notification": {"notification": "<Notification_Name>", "parameters": {"<Parameter1>":"<Parameter2>"}}}"

EX:

```
{
   "action": "notification/insert",
   "deviceId": "xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0",
   "notification": {
      "notification": "temp",
      "parameters": {"temperature": "25"}
   }
}
```



4.3 Received notification message



Subscribe to the topic "api/notification/<Network_ID>/#": This topic is defined by the platform service to receive notification messages.

When Publish is executed, the received format is as follows:

{"action":"notification/insert","notification":{"id":<notificationId>,"notification":"<Notification_Name>","deviceId":"<Device_ID>","networkId":6,"<Time_Stamp>":"<TimeStamp>","parameters":{"<Parameter1>":"<Parameter2>"}},"subscriptionId":<subscriptionId>}

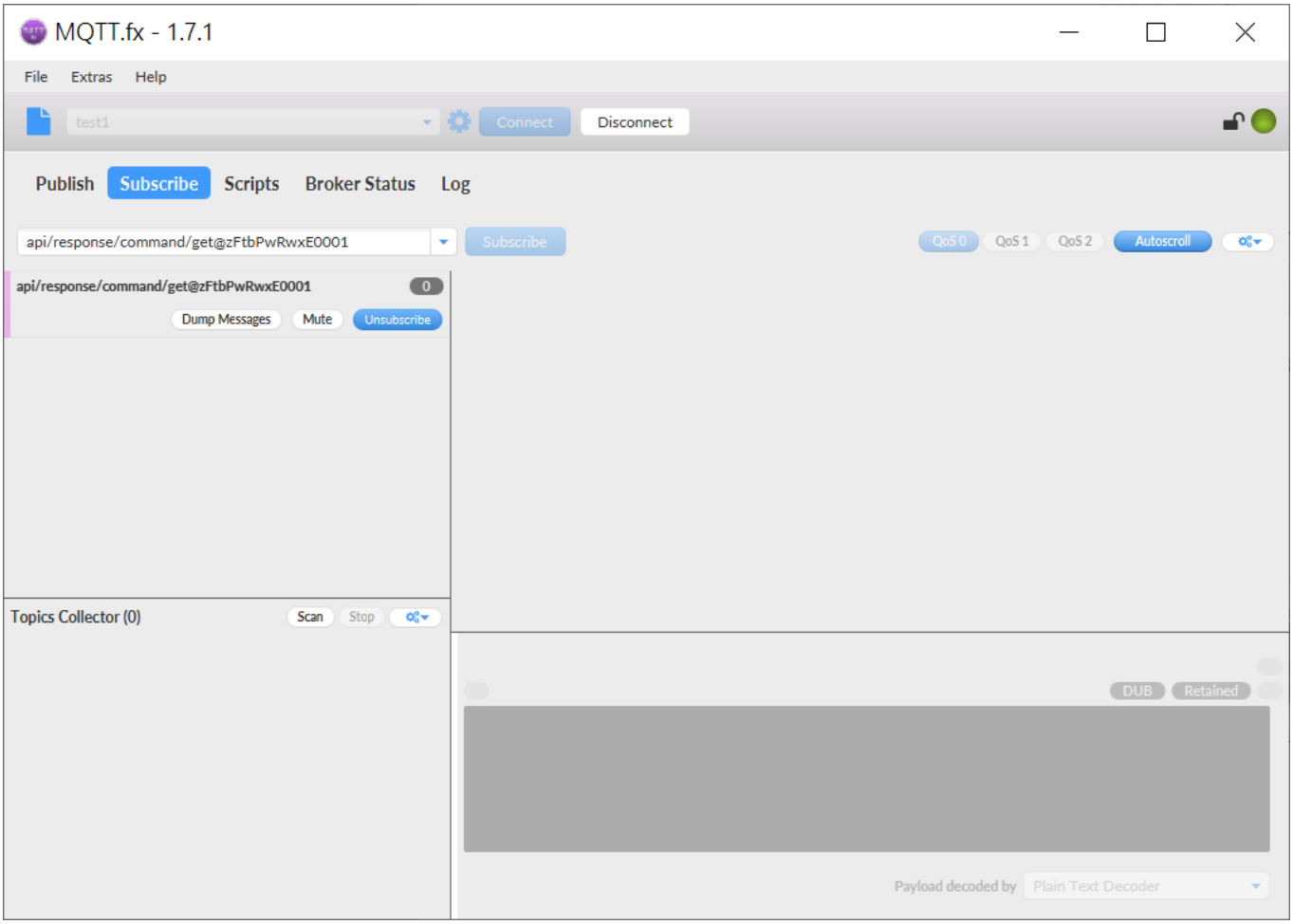
EX:

{"action":"notification/insert","notification":{"id":22938254,"notification":"temp","deviceId":"xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0","networkId":28,"deviceTypeId":5,"timestamp":"2020-11-02T15:44:06.503","parameters":{"temperature":"25"}},"subscriptionId":1604331608474512}

5.1 Subscribe command query message

Subscribe to the topic "api/response/command/get@<Clinet_ID>":

You can check your Clinet_ID from "Extras→Edit Connection Profiles"



5.2 Publish command query message

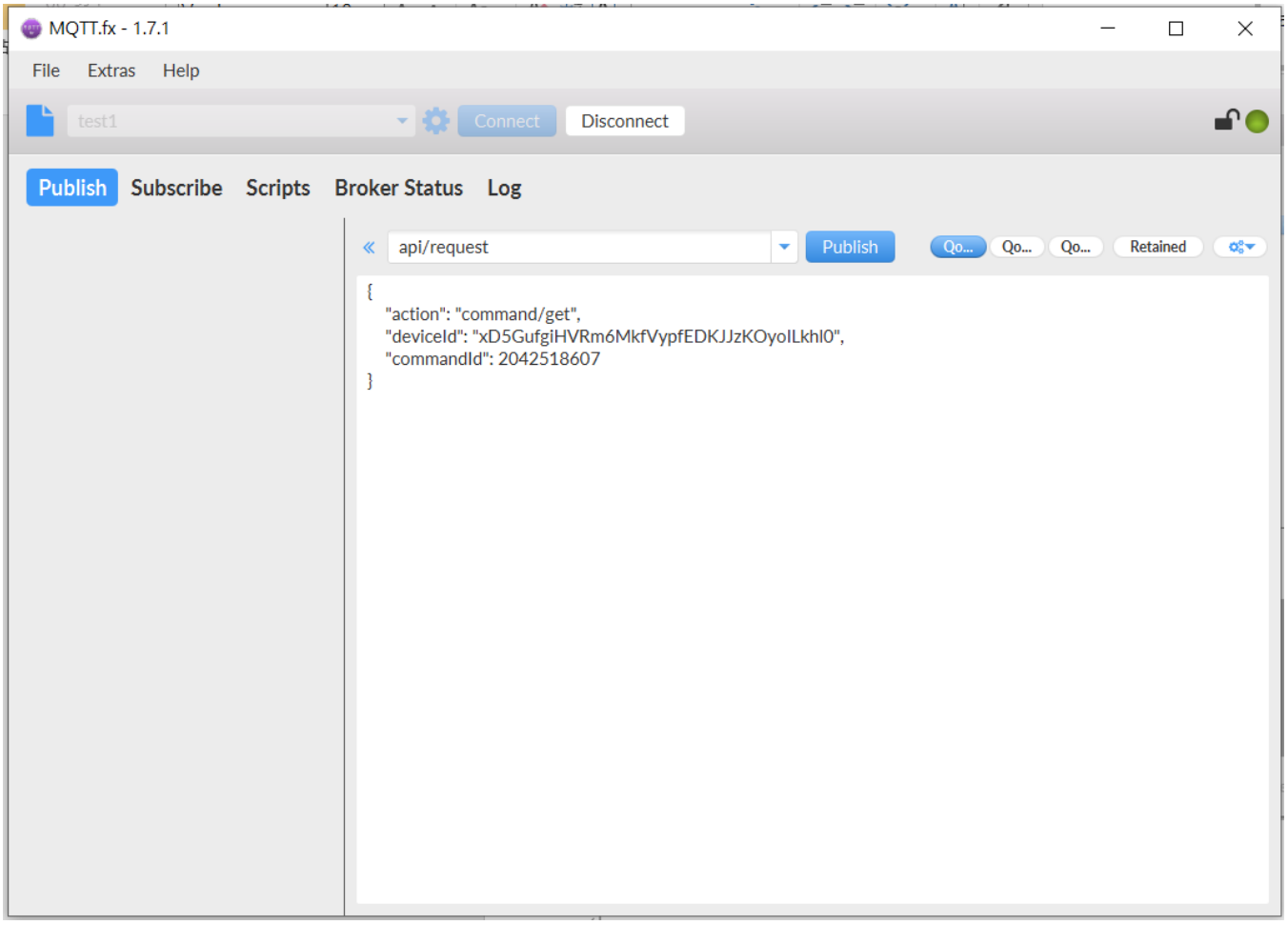
Publish Command_get message to "api/request" topic: The format and content of Command_get message are defined by this platform.

The command format is as follows:

{"action":"command/get","deviceId":"<Device_ID>","commandId":<commandId>}

EX:

```
{
   "action": "command/get",
   "deviceId": "xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0",
   "commandId": 2042518607
}
```



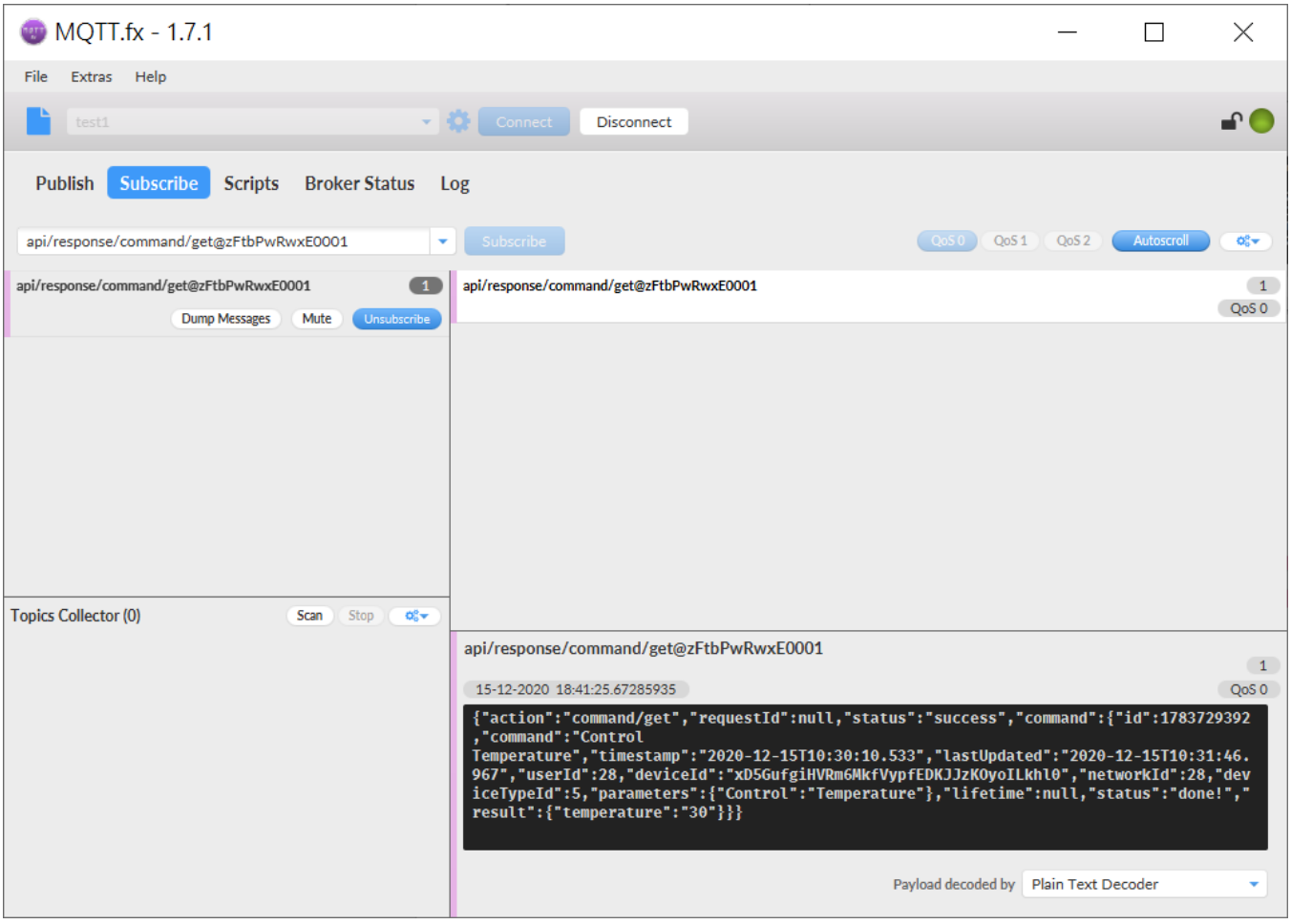
5.3 Received command query message

When Publish is executed, the received format is as follows:

{"action":"command/get","requestId:null,
"status":"success","command":{"id":<commandId>,"command":"<Command_Name>","timestamp":"<Time_Stamp>","lastUpdated":"<Time_Stamp>","userId":6,"deviceId":"<Device_ID>","networkId":6,"deviceTypeId":4,"parameters":{"<Parameter1>":"<Parameter2>"},"lifetime":null,"status":"<Status>","result":{"<Result1>":"<Result2>"}}}

EX:

{"action":"command/get","requestId":null,"status":"success","command":{"id":2042518607,"command":"Control Temperature","timestamp":"2020-11-03T09:56:27.588","lastUpdated":"2020-11-03T09:56:27.588","userId":28,"deviceId":"xD5GufgiHVRm6MkfVypfEDKJJzKOyoILkhl0","networkId":28,"deviceTypeId":5,"parameters":{"Control":"Temperature"},"lifetime":null,"status":"to do","result":{"temperature":"27"}}}



REYAX
TECHNOLOGY CORPORATION, LTD

Taiwan: sales@reyax.com
China: sales@reyax.com.cn
http://reyax.com