MQTT Client Application Example for Visual studio C#

www.broker4iot.com

04.04.21

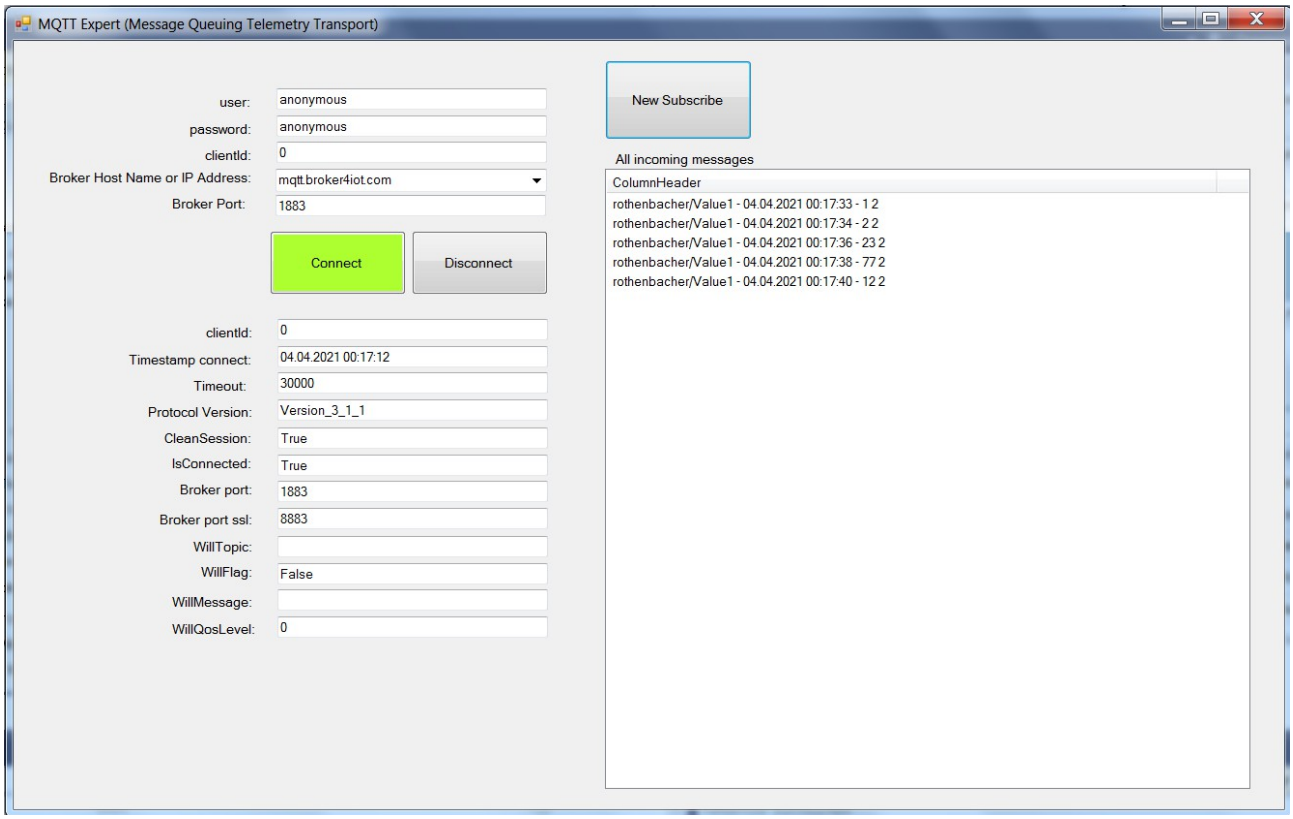
VERSION: 1.1

Inhaltsverzeichnis

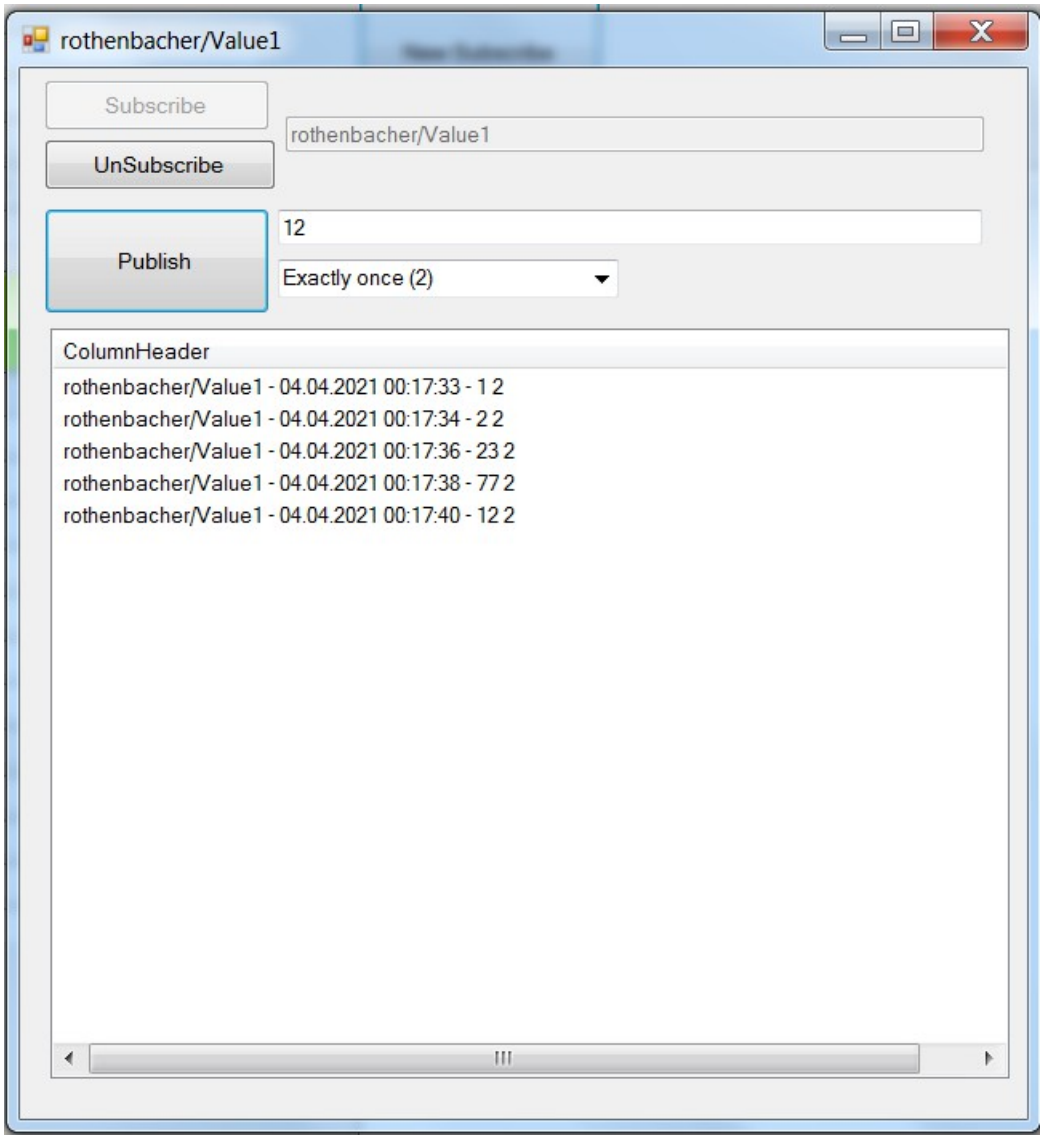
HISTORIE

Autor	Version	Datum	Änderung
@AR	01.01.00	03.04.21	Ersterstellung

Connection form



Subscribe form



Implementierung von Clients

Dieses Kapitel befasst sich mit der Implementierung von MQTT Clients.
Die Implementierung der Clients erfolgt in der Programmiersprache C#,
basierend auf dem .Net Framework. Zur Erstellung der Clients habe ich die Version 4.3.0
der M2Mqtt Bibliothek von Pablo Patierno verwendet [22].
Zunächst muss das passende NuGet Paket installiert und die zugehörige using Directive
im Projekt hinzugefügt werden. Als erstes wird mit Hilfe des Hostnamens oder alternativ
dessen IP Adresse ein neues MqttClient-Objekt erstellt. Im nächsten Schritt kann eine
Verbindung zum Broker4iot.com hergestellt werden. Falls keine persistente Session verwendet
wird, muss zunächst eine Client-ID erzeugt werden. Andernfalls kann auch eine statische
Client-ID übergeben werden. Folgendes Codebeispiel zeigt den Verbindungsaufbau
ohne persistente Session.

```
// Create Client instance
myClient = new MqttClient(txt_brokerurl.Text);

// Register to message received event
myClient.MqttMsgPublishReceived += client_recievedMessage;

string clientId = txtclientid.Text;
string tuser = txtuser.Text;
string tpassword = txtpassword.Text;

myClient.Connect(clientId, tuser, tpassword);
```

Die einzelnen Parameter der Verbindung werden bei diesem Vorgang festgelegt. Falls
kein Passwortschutz der eigenen Session gewünscht wird, kann als Username und
Passwort alternativ der Parameter NULL übergeben werden. Sollte ein Passwort übergeben
werden, darf der Username Parameter jedoch nicht NULL sein. Der Parameter
des WILL QoS kann außerdem alternativ als Integer übergeben werden.

Wenn der Client nicht nur Nachrichten veröffentlichen, sondern auch Nachrichten anderer Publisher empfangen will, muss er die gewünschten Topics, inklusive zugehörigem Quality of Service, abonnieren. Das folgende Codebeispiel stellt einen beispielhaften Subscribe dar.

```
//Topic Subscriben
ushort SubscribeID = myClient.Subscribe(new string[] { TB_Topic.Text }, new byte[]
{ MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE });
topic = TB_Topic.Text;

Bu_Subscribe.Enabled = false;
TB_Topic.Enabled = false;

Bu_Publish.Enabled = true;
TB_Data.Enabled = true;

this.Text = topic ;
```

Hierbei können mehrere Topics auf einmal abonniert werden. Der Client muss nicht für jedes Topic eine eigene Subscribe Nachricht verschicken. Um empfangene Nachrichten verarbeiten zu können, muss dem Client eine Methode hinzugefügt werden, die die Daten aus der Nachricht extrahiert. Zunächst muss in der Main Methode angegeben werden welche Methode die empfangene Nachricht erhalten soll. Dies geschieht mit folgendem Befehl:

```
// Register to message received event
myClient.MqttMsgPublishReceived += client_recievedMessage;
```

Anschließend kann die Methode deklariert werden, welche den selben Methodennamen besitzen muss, wie zuvor in der Main Methode angegeben.

```
// Handle message received
var message = System.Text.Encoding.Default.GetString(e.Message);
if (InvokeRequired)
{
BeginInvoke(new Event_DataChange(client_recievedMessage), new object[] { sender, e });
return;
}
foreach (var TC in liTopicControls)
{
if(TC.Text == e.Topic)
{
TC.li_Msgs.Add(e.Topic + "-" + DateTime.Now.ToString() + "-" + message + "" + e.QosLevel);
TC.UpdateListView();
}
}
```

Um selbst als Publisher zu fungieren und Nachrichten zu veröffentlichen, müssen die dafür notwendigen Parameter der Publish Methode übergeben werden. Das Retain Flag

bezeichnet hierbei ob die Nachricht als Retained Message des Topics abgespeichert werden soll. Hierbei muss beachtet werden das Wildcards in einer Publish Nachricht nicht zulässig sind.

```
//Nachricht auf Topic schicken
ushort msgId = myClient.Publish(topic,Encoding.UTF8.GetBytes(TB_Data.Text), QoS, true);

//Publish besteht aus : 1. Der Topic(srting) 2. Der Nachricht(byte[]) 3.QoS als MsgBaseQoS
4. Der Retained Info
```

Um Topics deabonnieren zu können, müssen lediglich die gewünschten Topics der Unsubscribe Methode übergeben werden.

```
myClient.Unsubscribe( new String[] {TB_Topic.Text} );
button1.Enabled = false;
Bu_Subscribe.Enabled = true;
```