MQTT

دید کلی

مقدمه

MQTT که سرواژه شده Message Queuing Telemetry Transport میباشد، یکی از سریع ترین پروتکل هایی میباشد که در اینترنت اشیا مورد استفاده قرار میگیرد.

این پروتکل اولین بار توسط شرکت IBM در سال 1999 برای استفاده در خط لوله های نفتی توسعه پیدا کرد و در سال 2010 به صورت open source انتشار و در سال 2013 به یک پروتکل ارتباطی استاندارد تبدیل شد.

[source]

<http://www.steves-internet-guide.com/mqtt/>

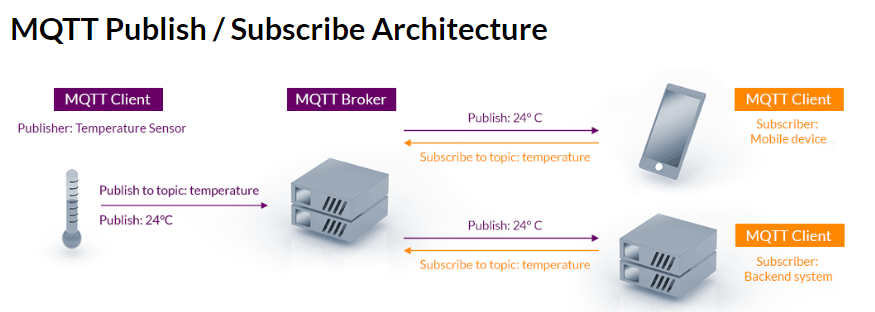
نحوه کارکرد

سیستم MQTT به صورت کلی از Client و Broker تشکیل شده است، به این صورت که Client ها میتوانند با ارسال و ارسال packet های MQTT، پیام ها و داده ها را ارسال و دریافت کنند، و Broker نیز نقش Server را برای ما ایفا خواهد کد.

نحوه کارکرد MQTT بسیار مشابه با نحوه کارکرد Instagram است و فقط در نامگذاری تفاوت هایی وجود دارد. اگر خود را در اینستاگرام در نظر بگیریم، به عنوان یک کاربر میتوانیم کاربران دیگر را دنبال کنیم و کاربران دیگر نیز میتوانند ما را دنبال کنند. در عین حال هر فردی میتواند چندین اکانت اینستاگرام داشته باشد

حال تفاوت های این نامگذاری ها را میتوانیم در جدول زیر خلاصه کنیم

|  |  |
| --- | --- |
| Instagram | MQTT |
| account | Topic |
| Post | Publish |
| Follow | Subscribe |



برای مثال، همان طور که از تصویر بالا مشخص است، سنسور دما در این قسمت دما اندازه گیری شده را به یک Topic خاص (در اینجا temperature) منتشر میکند (Publish)، سپس Broker به عنوان Server این پیام را دریافت کردن و برای تمامی Subscriber ها ارسال میکند.

[source]

<https://www.hivemq.com/mqtt-essentials/>

QoS (Quality of Service)

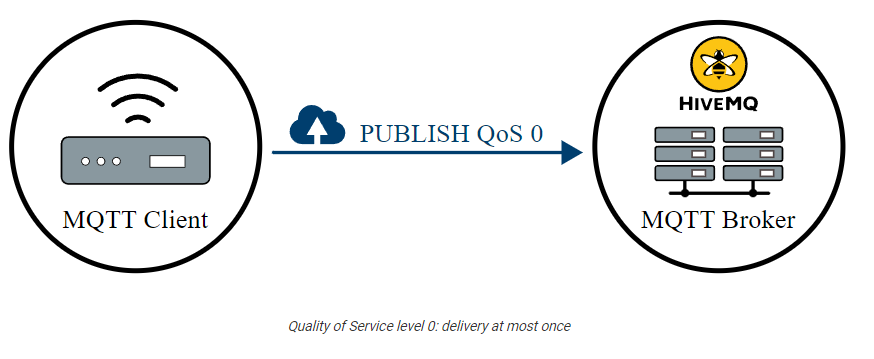
یکی از مهمترین پارامترهای استفاده شده در MQTT، QoS میباشد، که عددی است بین 0و1و2

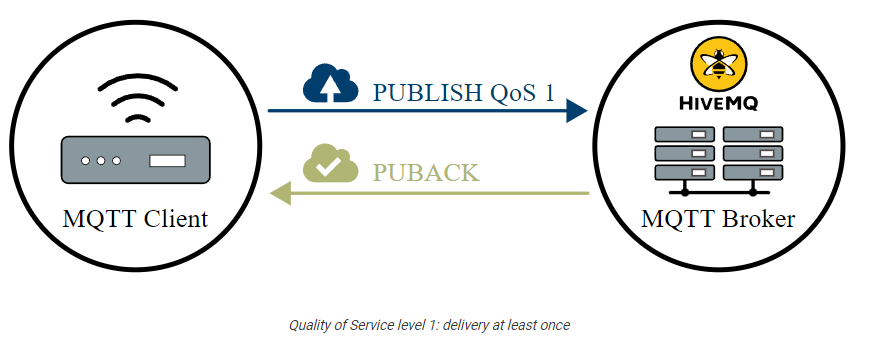
این عدد در اصل نحوه کارکرد کاربر را تعیین میکند،

اگر کاربری خود را با QoS=0 به Broker معرفی کند، زمانی که Broker دیتای جدیدی را دریافت میکند، دیتا را بدون توجه به اینکه آیا کاربر به اینترنت متصل است ارسال میکند، و در این صورت ممکن است اگر کاربر به اینترنت متصل نباشد، دیتا را از دست دهد. (این مورد در حالتی استفاده میشود که دیتای دریافتی اهمیت چندانی ندارد و در صورت از دست رفتن مهم نیست)

اگر کاربر خود را با QoS=1,2 به Broker معرفی کند، در زمانی که کاربر به اینترنت متصل نباشد، دیتا توسط Broker ذخیره شده و هر زمانی که کاربر دوباره به Broker متصل شود، دیتاها دوباره توسط Broker به آن ارسال میشود.

نحوه کارکرد QoS=0, 1 در شکل های زیر مشخص است





[source]

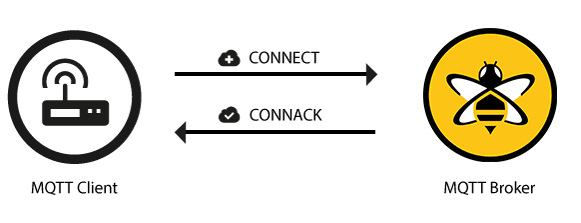
<https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>

جزئیات کارکرد

اتصال به Broker – (Connect)

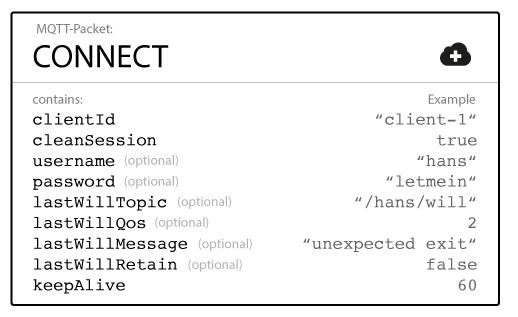
برای اتصال به Broker دو مرحله وجود دارد

1. ارسال Packet Connect - CONNECT
2. دریافت Connect Acknowledge – CONNACK



CONNECT

اطلاعاتی که باید در این Packet ارسال شود، به صورت زیر است



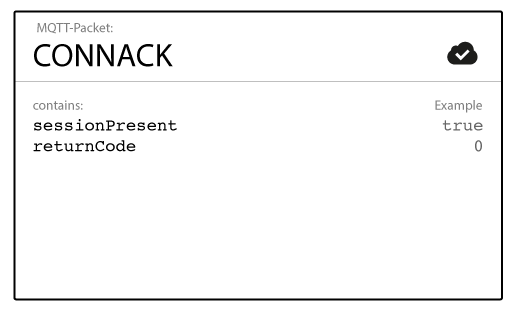
نکات

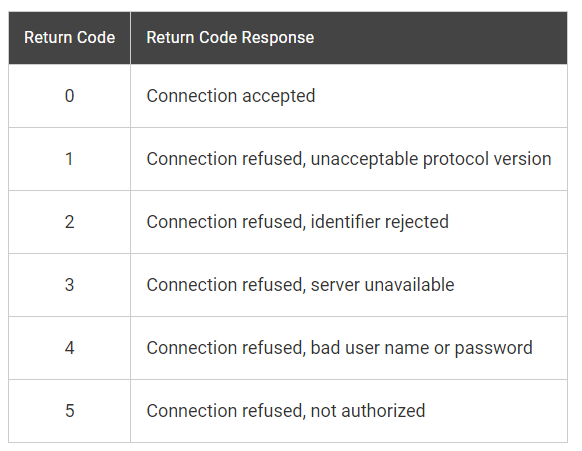
اگر از QoS=1,2 استفاده شود باید cleanSession برابر false قرار گیرد.

keepAlive زمانی است به صورت پریودیک که باید PING Request به سرور ارسال شود، در غیر این صورت ارتباط قطع خواهد شد. (در اصل برای این است که Broker متوجه شود هنوز Client به اینترنت متصل است)

CONNACK

اگر Packet اتصال به صورت موفقیت آمیز ارسال شده باشد، باید این Packet را به صورت صحیح دریافت کنیم



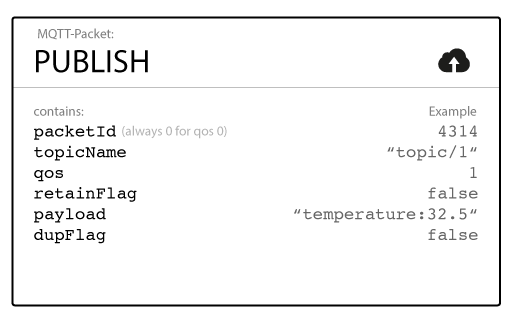


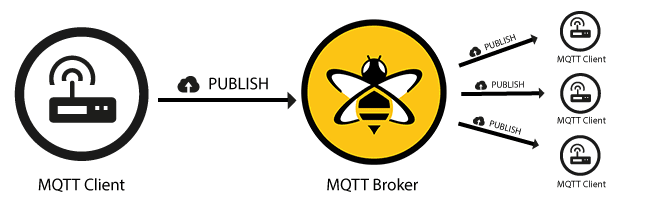
[source]

<https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>

منتشر کردن در Broker – (Publish)

این قسمت بنا به اینکه از چه QoS ای استفاده کرده ایم، متفاوت است، برای مثال اگر از QoS=0 استفاده کرده باشیم، فقط نیاز است Packet با مشخصات زیر را ارسال نماییم



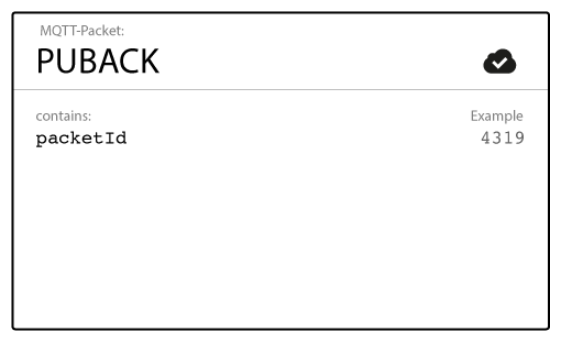


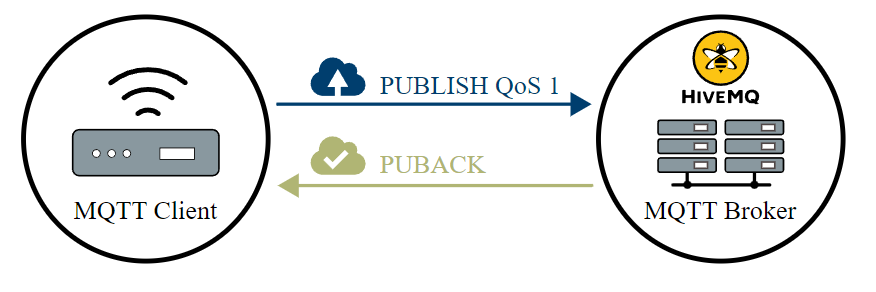
[source]

<https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe/>

اگر از QoS=1 استفاده کرده باشیم، نیاز است در صورت دریافت کردن هر PUBLIC در جواب یک PUBACK در پاسخ به فرستنده ارسال نماییم

این Packet به صورت زیر است





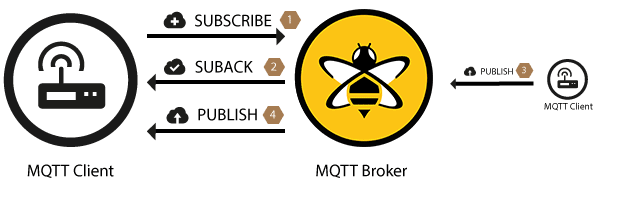
[source]

<https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>

عضویت در Topic – (Subscribe)

این مرحله مانند مرحله Connect میباشد و دو مرحله دارد

1. ارسال Subscribe توسط کاربر
2. پاسخ Suback توسط Broker

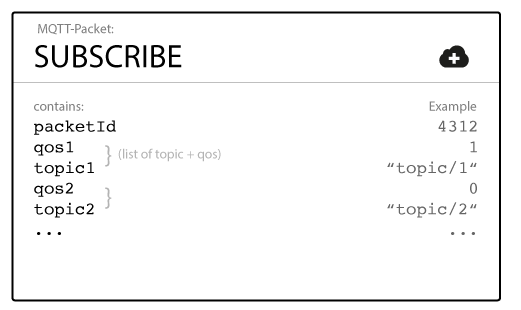


[source]

<https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe/>

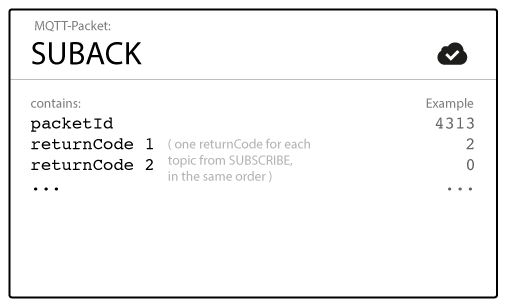
Subscribe

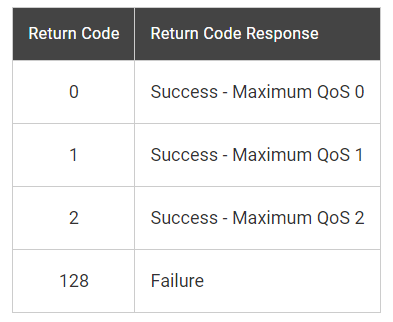
برای اینکار تنها نیاز است Packet زیر را برای Broker ارسال کنیم



Suback

پاسخ Broker به صورت زیر میباشد





مثال عملی با پایتون

ابتدا یکی از کدها را برای عنوان ارسال کننده با QoS=1 تنظیم میکنیم

from distutils.command.clean import clean

import paho.mqtt.client as mqtt

این تابع در زمانی که به Broker متصل شدیم، فراخوانی خواهد شد

def on\_connect(client, userdata, flags, rc):

    print("Connected with result code " + str(rc))

این تابع log های مربوطه را در خروجی چاپ خواهد کرد

def on\_log(client, userdata, level, buf):

    print(f"log - {buf}")

ابتدا یک کاربر با آیدی تصادفی میسازیم و logger را نیز برای آن فعال میکنیم

client = mqtt.Client(client\_id="12847720", clean\_session=False)

client.on\_connect = on\_connect

client.on\_log = on\_log

client.enable\_logger()

درخواست Connect را برای Broker مربوطه ارسال میکنیم و برنامه را فعال میکنیم.

client.connect("mqtt.eclipseprojects.io", 1883, 60)

client.loop\_start()

از طرفی دیگر یکی از کدها را به عنوان دریافت کننده با QoS=1 به صورت زیر تنظیم میکنیم

from pydoc import cli

import paho.mqtt.client as mqtt

def on\_connect(client, userdata, flags, rc):

    print("Connected with result code "+str(rc))

    client.subscribe("scooter/112", qos=1)

این تابع زمانی که یک پیام جدید از طرف Broker دریافت شد فراخوانی خواهد شد. {بقیه توابع تکراری میباشند، به همین دلیل از توضیح دوباره آن ها صرف نظر میشود}

def on\_message(client, userdata, msg):

    print(msg.topic+" "+str(msg.payload))

def on\_log(client, userdata, level, buf):

    print(f"log - {buf}")

client = mqtt.Client(client\_id="62991728", clean\_session=False)

client.on\_connect = on\_connect

client.on\_message = on\_message

client.on\_log  = on\_log

client.enable\_logger()

client.connect("mqtt.eclipseprojects.io", 1883, 60)

client.loop\_forever()

شروع ارسال

برای اینکار تکه کد زیر را به فرستنده خود باید اضافه نماییم، تا هر 10 ثانیه یکبار، یک پیام ارسال نماید

from time import sleep

sleep(10)

counter = 0

while True:

    client.publish("scooter/112", str(counter), qos=1)

    print(f"Publish {counter}")

    counter += 1

    sleep(10)

خروجی

در طرف فرستنده خود، خروجی زیر را داریم

log - Sending CONNECT (u0, p0, wr0, wq0, wf0, c0, k60) client\_id=b'12847720'

log - Received CONNACK (1, 0)

Connected with result code 0

log - Sending PUBLISH (d0, q1, r0, m1), 'b'scooter/112'', ... (1 bytes)

Publish 0

log - Received PUBACK (Mid: 1)

log - Sending PUBLISH (d0, q1, r0, m2), 'b'scooter/112'', ... (1 bytes)

Publish 1

log - Received PUBACK (Mid: 2)

در طرف گیرنده خود، خروجی به صورت زیر میشود

log - Sending CONNECT (u0, p0, wr0, wq0, wf0, c0, k60) client\_id=b'62991728'

log - Received CONNACK (1, 0)

Connected with result code 0

log - Sending SUBSCRIBE (d0, m1) [(b'scooter/112', 1)]

log - Received SUBACK

log - Received PUBLISH (d0, q1, r0, m3), 'scooter/112', ...  (1 bytes)

scooter/112 b'0'

log - Sending PUBACK (Mid: 3)

log - Received PUBLISH (d0, q1, r0, m4), 'scooter/112', ...  (1 bytes)

scooter/112 b'1'

log - Sending PUBACK (Mid: 4)

همان طور که واضح است، ترتیب Packet های ارسالی و دریافتی بسیار واضح هستند.