



Kostenloses eBook

LERNEN MQTT

Free unaffiliated eBook created from
Stack Overflow contributors.

#mqtt

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit MQTT	2
Bemerkungen	2
Examples	2
Einführung	2
Kapitel 2: Eigenschaften von MQTT	3
Einführung	3
Examples	3
Ein einfaches Public / Subscribe-Modell in MQTT	3
Kapitel 3: Implementierung von MQTT	5
Examples	5
Beispiel für Publish / Subscriber in Java	5
Kapitel 4: Installation und Einrichtung	7
Einführung	7
Examples	7
MQTT-Bibliotheken und MQTT-Broker	7
Schritte zum Installieren von ActiveMQ-Broker	8
Credits	11



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [mqtt](#)

It is an unofficial and free MQTT ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official MQTT.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit MQTT

Bemerkungen

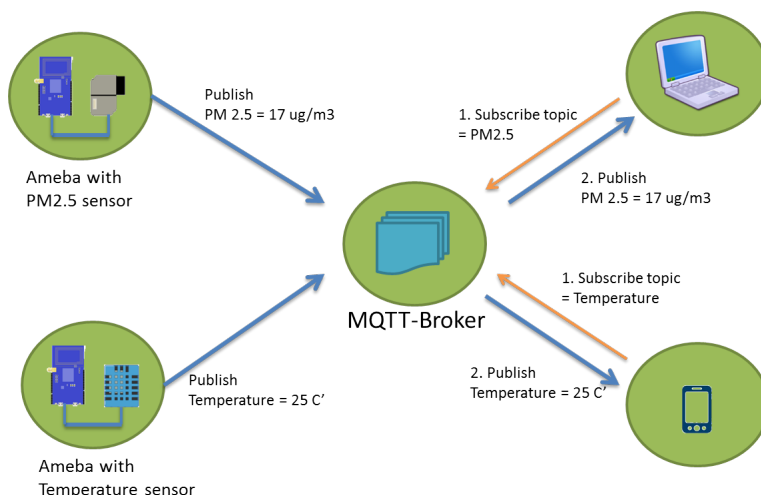
In diesem Abschnitt erhalten Sie einen Überblick darüber, was mqtt ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in mqtt erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für mqtt neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Einführung

MQTT (Message Queue Telemetry Transport) ist ein **Publish-Subscribe**-basiertes "leichtes" Messaging-Protokoll zur Verwendung auf dem **TCP / IP**- Stack.



Dies ist sehr nützlich für Verbindungen mit Remote-Standorten, an denen ein geringer Code-Footprint erforderlich ist und / oder die Netzwerkbandbreite von Bedeutung ist.

Es gibt viele verschiedene Broker und Clients, die das MQTT-Protokoll implementieren.

Eine Liste von Broker, Clients und Werkzeugen finden Sie auf der Website zu finden mqtt.org [hier](#), während es nicht definitiv ist es eine repräsentative Stichprobe bietet. Auf github.com gibt es auch eine kuratierte Liste zu MQTT.

Nachrichten zu den MQTT-Spezifikationen finden Sie unter mqtt.org.

MQTT v3.1.1 ist ein [hier](#) verfügbarer Oasis-Standard

Erste Schritte mit MQTT online lesen: <https://riptutorial.com/de/mqtt/topic/8187/erste-schritte-mit-mqtt>

Kapitel 2: Eigenschaften von MQTT

Einführung

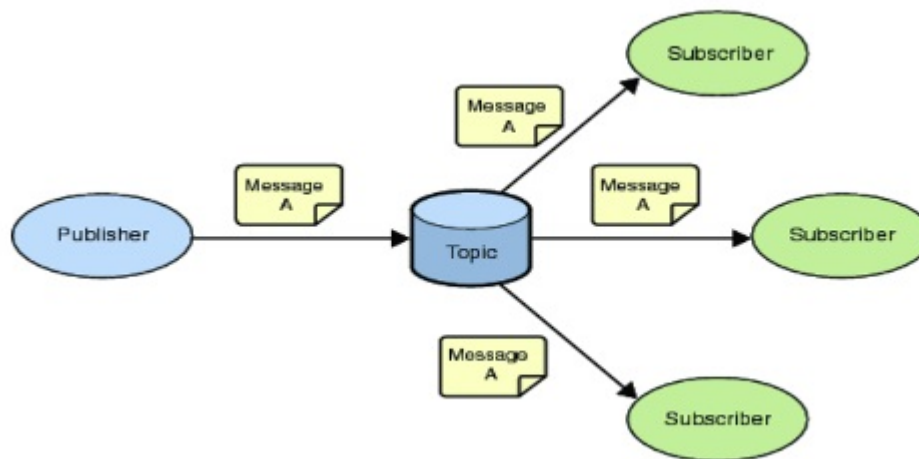
Das Protokoll läuft über TCP / IP oder über andere Netzwerkprotokolle, die geordnete, verlustfreie bidirektionale Verbindungen bereitstellen.

Examples

Ein einfaches Public / Subscribe-Modell in MQTT

Die wichtigsten Merkmale sind:

- Verwendung des Publish / Subscribe-Nachrichtenmusters, das die Verteilung von Nachrichten an mehrere Nachrichten und die Entkopplung von Anwendungen ermöglicht.
- Ein Nachrichtentransport, der den Inhalt der Nutzdaten nicht beeinflusst. Drei Servicequalitäten für die Zustellung von Nachrichten
- Ein kleiner Transportaufwand und Protokollaustausch werden minimiert, um den Netzwerkverkehr zu reduzieren



Im Allgemeinen gibt es zwei Arten von Messaging-Diensten.

- Warteschlange (Einzelverbindung)
- Thema (eins zu eins / eins zu vielen)

MQTT unterstützt keine Warteschlangen, die zuverlässig sind, aber MQTT unterstützt Topic. Standardmäßig ist Topic unzuverlässig. Wir können jedoch MQTT-Funktionen und -Methoden verwenden, um es zuverlässig zu machen.

Unterschied zwischen Thema und Warteschlange

Warteschlange:

- Punkt-zu-Punkt-Modell
- Nur ein Verbraucher erhält die Nachricht
- Nachrichten müssen in der gesendeten Reihenfolge zugestellt werden
- Eine Warteschlange garantiert nur, dass jede Nachricht nur einmal verarbeitet wird.
- Die Warteschlange weiß, wer der Consumer oder der JMS-Client ist. Das Ziel ist bekannt.
- Der JMS-Client (der Consumer) muss nicht immer aktiv sein oder mit der Warteschlange verbunden sein, um die Nachricht zu empfangen oder zu lesen.
- Jede erfolgreich verarbeitete Nachricht wird vom Verbraucher bestätigt.

Thema:

- Modell veröffentlichen / abonnieren
- Mehrere Kunden abonnieren die Nachricht
- In der gesendeten Bestellung müssen keine Nachrichten übermittelt werden
- Es kann nicht garantiert werden, dass jede Nachricht nur einmal verarbeitet wird. Da kann dies vom Modell abgefühlt werden
- Das Thema hat mehrere Abonnenten und es besteht die Möglichkeit, dass das Thema nicht alle Abonnenten kennt. Das Ziel ist unbekannt
- Der Abonnent / Client muss aktiv sein, wenn die Nachrichten vom Produzenten erstellt werden, es sei denn, das Abonnement war ein dauerhaftes Abonnement.
- Nein, jede erfolgreich verarbeitete Nachricht wird vom Verbraucher / Teilnehmer nicht bestätigt.

Mit MQTT können wir jedoch die Nachteile des Themas reduzieren. Das Thema kann zuverlässig sein und die Duplikate in den MQTT-Funktionen steuern

Eigenschaften von MQTT online lesen: <https://riptutorial.com/de/mqtt/topic/9129/eigenschaften-von-mqtt>

Kapitel 3: Implementierung von MQTT

Examples

Beispiel für Publish / Subscriber in Java

Erstellen Sie ein dynamisches Webprojekt in sts / eclipse. Laden Sie das Eclipse-Paho-Jar von [hier herunter](#) , um die JAR-Datei in Webcontent-> Webinf-> Ordner-> Lib herunterzuladen

Veröffentlichen Sie Beispiel

```
String broker = "tcp://localhost:1883";
String topicName = "test/topic";
int qos = 1;

MqttClient mqttClient = new MqttClient(broker, String.valueOf(System.nanoTime()));
//Mqtt ConnectOptions is used to set the additional features to mqtt message

MqttConnectOptions connOpts = new MqttConnectOptions();

connOpts.setCleanSession(true); //no persistent session
connOpts.setKeepAliveInterval(1000);

MqttMessage message = new MqttMessage("Ed Sheeran".getBytes());
```

// hier ist ed sheeran eine nachricht

```
message.setQos(qos); //sets qos level 1
message.setRetained(true); //sets retained message

MqttTopic topic2 = mqttClient.getTopic(topicName);

mqttClient.connect(connOpts); //connects the broker with connect options
topic2.publish(message); // publishes the message to the topic(test/topic)
```

Beispiel abonnieren

```
//We're using eclipse paho library so we've to go with MqttCallback
MqttClient client = new MqttClient("tcp://localhost:1883", "clientid");
client.setCallback(this);
MqttConnectOptions mqOptions=new MqttConnectOptions();
mqOptions.setCleanSession(true);
client.connect(mqOptions); //connecting to broker
client.subscribe("test/topic"); //subscribing to the topic name test/topic

//Override methods from MqttCallback interface
@Override
public void messageArrived(String topic, MqttMessage message) throws Exception {
    System.out.println("message is : "+message);
}

.
//other override methods
```

Implementierung von MQTT online lesen:

<https://riptutorial.com/de/mqtt/topic/9132/implementierung-von-mqtt>

Kapitel 4: Installation und Einrichtung

Einführung

Um MQTT zu implementieren

Wir benötigen MQTT Broker und MQTT Client Library

Examples

MQTT-Bibliotheken und MQTT-Broker

Für die Verwendung von MQTT in der Anwendung stehen verschiedene Bibliotheken für verschiedene Programmiersprachen zur Verfügung.

MQTT-Bibliothek

BIBLIOTHEK	SPRACHE	BESCHREIBUNG
Eclipse Paho	C, C ++, Java, Javascript, Python, Los, C #	Paho-Clients gehören zu den beliebtesten Implementierungen von Client-Bibliotheken.
Sicherungsquelle-MQTT-Client	Java	Der MQTT-Client von Fusesource ist ein Java-MQTT-Client mit drei verschiedenen API-Stilen: Blocking, Future-based und Callback-based.
MQTT.js	Javascript	MQTT.js ist eine MQTT-Clientbibliothek für Node.js und Webanwendungen, die als npm-Modul verfügbar ist.
Ruby-Mqtt	Rubin	ruby-mqtt ist ein MQTT-Client, der als Ruby-Edelstein erhältlich ist. QoS> 0 wird nicht unterstützt.

MQTT-Broker

Der Broker ist in erster Linie dafür verantwortlich, alle Nachrichten zu empfangen (Broker ist wie ein Nachrichtenserver), zu filtern, zu entscheiden, wer daran interessiert ist, und dann die Nachricht an alle abonnierten Clients zu senden. MQTT-Broker-Implementierungen: Die folgende Tabelle zeigt einige der beliebtesten Open-Source- und kommerziellen Broker-Implementierungen.

Makler _____	Beschreibung
Apache	ActiveMQ ist ein Open-Source-Multi-Protocol-Nachrichtenbroker, dessen

Makler _____	Beschreibung
ActiveMQ	Kern um JMS geschrieben ist. Es unterstützt MQTT und bildet die MQTT-Semantik über JMS ab.
Moskito	
Kaninchen MQ	RabbitMQ ist eine skalierbare Open Source-Nachrichtenwarteschlangenimplementierung, die in Erlang geschrieben wird. Es ist ein AMQP-Nachrichtenbroker, verfügt jedoch über ein MQTT-Plugin. Unterstützt nicht alle MQTT-Funktionen (z. B. QoS 2).
HiveMQ	HiveMQ ist ein skalierbarer, leistungsstarker MQTT-Broker, der sich für missionskritische Implementierungen eignet. Es unterstützt vollständig MQTT 3.1 und MQTT 3.1.1 und verfügt über Funktionen wie Websockets, Clustering und ein Open-Source-Plug-In-System für Java-Entwickler.
WebsphereMQ / IBM MQ	Websphere MQ ist eine kommerzielle, nachrichtenorientierte Middleware von IBM. Unterstützt vollständig MQTT.

Schritte zum Installieren von ActiveMQ-Broker

Gehen Sie zur ActiveMQ-Website und laden Sie die neueste stabile Version von activeMQ herunter

[Klicken Sie hier, um ActiveMQ-Downloads herunterzuladen](#)

- entpacken Sie es nach dem Download

wenn Sie Windows 32 verwenden

- **Gehen Sie zu Apache-Activemq-5.14.3 \ bin \ win32**

wenn Windows 64

- **apache-activemq-5.14.3 \ bin \ win64**
- Führen Sie die **Activemq-** Batchdatei aus
- Das ist es, ActiveMQ-Server wird an der Eingabeaufforderung ausgeführt

wenn Sie die UI Console für activeMQ sehen möchten. um zu erfahren, wie Nachrichten organisiert und gesendet werden

zu [http: // localhost: 8161 / admin /](http://localhost:8161/admin/)

Authentication Required

http://localhost:8161 requires a username and password.

User Name:

Password:

Log In

Cancel

- standardmäßig

Benutzername = Admin

Passwort = Admin

- Klicken Sie dann auf die Registerkarte "Thema".



Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Topic Name

Topics

Name ↑	Number Of Consumers	Messages Enqueued	Messages Dequeued
ActiveMQ.Advisory.Connection	0	3	0
ActiveMQ.Advisory.MasterBroker	0	1	0
ActiveMQ.Advisory.Topic	0	10	0

Copyright 2005-2015 The Apache Software Foundation.

Die Registerkarte "Themen" enthält Informationen darüber, wie viele Themen vorhanden sind und ob aktive Verbraucher vorhanden sind, ob Nachrichten übermittelt werden oder nicht.

Installation und Einrichtung online lesen: <https://riptutorial.com/de/mqtt/topic/9130/installation-und-einrichtung>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit MQTT	Community , ed sheeran , hardillb , Ivo Limmen , Trishant Pahwa
2	Eigenschaften von MQTT	ed sheeran
3	Implementierung von MQTT	ed sheeran
4	Installation und Einrichtung	ed sheeran