

# Übung 08

## Android MQTT

### Erstellen einer Chat-Software

### 4abHWII

April 20, 2022



Übungsleiter: Albert Greinöcker

Ziel der Übung:

- Verwendung von MQTT in Android
- Einrichten eines MQTT-Brokers in der Cloud

In dieser Aufgabe wollen wir uns die MQTT-Technologie zunutze machen und eine kleine Chat-Software schreiben, die ohne speziellen Server auskommt.

## 1 Einrichten eines MQTT-Brokers in der Cloud

Es gibt viele Möglichkeiten, einen MQTT-Broker in der Cloud einzurichten, der gängigste ist: <https://www.hivemq.com/mqtt-cloud-broker/>. Es wird verschlüsselte Übertragung sowie Absicherung durch Authorisierung mittels Benutzernamen und Passwort in der Gratis-Version angeboten.

Hier bitte:

- a. Einfach mal registrieren
- b. Einen neuen Cluster einrichten. Unter Overview kann man dann die Cluster-URL einsehen, die dann in den Android-Clients verwendet werden soll (man kann diese auch anpassen). Dieser Dienst läuft nicht unter den Ports 1883 bzw. 1884, sondern 8883 und 8884, da sie mit TLS verschlüsselt sind.
- c. Unter Access Management können Benutzer angelegt werden. Für unser Vorhaben reicht einer.
- d. Unter Getting Started bekommt man schon fix fertige Code-Segmente für die Verarbeitung mit HiveMQ in unterschiedlichen Technologien sowie weitere Tools bereitgestellt. Bitte mal einen Blick drauf werfen!

## 2 Einrichten des Vorlagen-Programms

In unseren Android-Examples auf github ([https://github.com/albertgreinoecker/Android\\_Examples](https://github.com/albertgreinoecker/Android_Examples)) ist ein Beispiel in dem grundsätzlich Nachrichten über MQTT ausgetauscht werden können. Folgende Komponenten sind in ein eigenes Projekt zu übernehmen bzw. anzupassen (nicht vom PDF, sondern vom Projekt übernehmen!):

a. res/layout/activity\_mqtt\_activity.xml

b. at.ac.androidexamples.mqtt.MQTTActivity.java (Wird im Unterricht ausführlich besprochen). Diese Werte einfach von HiveMQ übernehmen:

```
1 String serverUri; // Beispiel-Aufbau: ssl://xxxxxxxxxxx.s2.eu.hivemq.cloud
2 String user;
3 String password = "";
```

c. AndroidManifest.xml

- Setzen der Berechtigungen

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

- Unter den Tag application muss der Service registriert werden:

```
<service android:name="org.eclipse.paho.android.service.MqttService">
</service>
```

- Nicht vergessen die richtige Activity als Startactivity festzulegen!

d. build.gradle (Module):

```
implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.4'
implementation 'org.eclipse.paho:org.eclipse.paho.android.service:1.1.1'
```

Wenn alles korrekt eingerichtet ist, dann sollte ein einfacher Chatclient sichtbar sein (auf das Layout würde nicht geachtet). Folgendes ist zu beachten:

- Die Clientid (Variable in MQTTActivity.java) muss eindeutig sein.
- Zuerst Subscribe, dann Publish, sonst wird natürlich nichts übernommen.
- Im unteren Feld gibt man die Nachrichten ein, wenn diese ge-published werden, werden sie im oberen Textfeld angezeigt.

### 3 Funktionalität des Chat-Clients

Die bereits bestehende Funktionalität soll um Folgendes erweitert werden:

- a. Die Clientid muss man in einem Feld eingeben können. Diese soll dann auch angezeigt werden, wenn eine Nachricht gesendet wurde
- b. Der Channel soll einstellbar sein, über den man gerade chattet. Es ist auch möglich gleich in mehreren Channls zu subscriben, deshalb soll auch der Channel angezeigt werden, über den die Nachricht gekommen ist.
- c. Man soll sich auch von einem Topic wieder abmelden können. MqttAndroidClient bietet zu diesem Zweck eine Methode unsubscribe an.
- d. Jede weitere Idee ist natürlich super wenn sie eingebaut wird!

Bitte unbedingt mit mehreren Geräten ausprobieren (sonst macht ja der Chat keinen Sinn).

## 4 Ausblick

- Nach dieser Übung wird es keine mehr in diesem Fach geben, sondern "nur" ein Projekt, wo die MQTT-Funktionalität von Android einzubauen ist.