SPRC
Laboratorul #5 Protocoale pentru IoT (MQTT)

Versiunea 2

Responsabili: **Dorinel FILIP**

Obiectivele laboratorului

În urma parcurgerii laboratorului, studenții:

- Vor putea descrie caracteristicile necesare unui protocol de comunicație folosit în contextul IoT și vor putea identifica protocoalele care sunt potrivite pentru o aplicație descrisă pentru Internet of Things;
- Vor înțelege modul de funcționare al protocolului MQTT;
- Vor fi capabili o aplicație de rețea folosind primitivele specifice unui client de MQTT.

Noțiuni teoretice

Pentru realizarea laboratorului sunt necesare noțiunile de bază despre funcționarea protocoalelor de tipul Publisher/Subscriber, în speță a caracteristicilor MQTT.

Asistanți cu atenție la prezentarea asistentului pentru a vă însuși principalele noțiuni.

Asigurați-vă că v-ați lămurit cu referire la următoarele aspecte:

1. Tipurile de entități din cadrul protocolului (broker și clienți);
2. Componentele mesajelor (topic și payload);
3. Principalele operații din protocol (publish și subscribe);
4. Nivelurile de QoS și garanțiile oferite de ele;
5. Potrivirea topic-urilor folosind caracterul # wildcard.

Ulterior prezentării, vă recomandăm următoarele resurse pentru reactualizare/aprofundare:

- MQTT.org - Documentația oficială a protoclului.;
- HiveMQ.com - Introducere în MQTT (cel puțin primele 3 părți);
- Steves Guide - Overview asupra mesajelor MQTT.

Aplicații

Exercițiul 1 - brokerul de MQTT

Folosind imaginea de Docker eclipse-mosquitto porniți un broker de MQTT care să asculte pe portul 1883 al mașinii locale.

HINT: Începând cu versiunea 2 a Eclise Mosquitto, implicit, daemon-ul face bind pe interfața de loopback, iar autentificarea este obligatorie, atât timp cât fișierele de configurare nu specifica altfel. Așadar, un fișier de configurare minimal pentru acest exercițiu este cel din Listing 1.

```
listener 1883
allow_anonymous true
```

Listing 1: Exemplu de fisier mosquitto.conf.

Fișierul de configurație `mosquitto.conf` trebuie legat la `/mosquitto/config/mosquitto.conf` în container.

Exercițiul 2 - aplicație de chat

La acest exercițiu, ne propunem să creăm o aplicație de chat folosind protocolul MQTT.

În acest sens, aplicația voastră trebuie:

1. să se conecteze la brokerul de MQTT pornit la exercițiul 1;
2. să facă subscribe topicurilor de forma `sprc/chat/#`;
3. La primirea oricărui mesaj la care este abonat să afișeze pe ecran topicul exact de pe care a venit mesajul și conținutul acestuia;
4. Să citească mesaje de la tastatură, și să trimită conținutul lor pe un topic de forma `sprc/chat/NUME` unde NUME este un acronim al numelui studentului.

În rezolvarea exercițiului, nu există nicio restricție privind limbajul de programare sau bibliotecile folosite.

În Python, pentru rezolvarea exercițiului recomandăm biblioteca paho-mqtt. Informații despre instalarea bibliotecii și un exemplu de utilizare se găsesc la adresa `https://pypi.org/project/paho-mqtt/`.

Hint! Fiind vorba despre o aplicație ce procesează mesajele individual, cea mai naturală abordare este cea cu funcții callback, care este folosită și în exemplul de Getting Started de pe site, însă veți avea nevoie de o metodă diferită de **loop_forever()** care ar bloca programul și astfel nu ați mai putea citit mesaje de la tastatură. O sugestie ar fi metoda **loop_start** care nu este blocantă.

Exercițiul 3 - Going global

Modificați programul anterior astfel încât să se conecteze la brokerul public **broker.hivemq.com**. Asistentul va monitoriza mesajele primite pe `sprc/chat/#`.

Pentru a vi se puncta acest exercițiu, realizați o conversație de minim 3 replici cu unul din colegi, apoi publicați un mesaj cu conținutul. ***** DONE - Nume - Grupă**

Hints and tips:

- Pentru a vedea traficul realizat pe broker-ul de mai sus puteți accesa adresa `http://www.mqtt-dashboard.com`.
- Pentru debugging-ul aplicațiilor voastre puteți folosi clientul de MQTT cu interfață MQTT-Spy: `https://www.hivemq.com/blog/mqtt-toolbox-mqtt-spy/`.

Exercițiul vă va fi punctat atunci când mesajul ajunge la asistent.