Εργαστηριακή άσκηση 6: MQTT services

1 Create a Java Maven Project

```
Name = gr.upatras.mqtt
groupId = gr.upatras
artifactId = gr.upatras.mqtt
```

2 Edit pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>gr.upatras</groupId>
        <artifactId>gr.upatras.mqtt </artifactId>
        <version>0.0.1-SNAPSHOT</version>


        <dependencies>
                <dependency>
                        <groupId>org.eclipse.paho</groupId>
                        <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
                        <version>1.2.5</version>
                </dependency>

                <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-api -->
                <dependency>
                        <groupId>org.slf4j</groupId>
                        <artifactId>slf4j-api</artifactId>
                        <version>1.7.36</version>
                </dependency>
                <dependency>
                        <groupId>org.slf4j</groupId>
                        <artifactId>slf4j-simple</artifactId>
                        <version>1.7.36</version>
                </dependency>
        </dependencies>
</project>
```

3 Java program

Create a class `SimpleMqttClient`

```
package gr.upatras.mqtt.publisher;

import java.util.Random;
import java.util.UUID;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
```

```java
import org.eclipse.paho.client.mqttv3.MqttCallback;
import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
import org.eclipse.paho.client.mqttv3.MqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.MqttTopic;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class SimpleMqttClient implements MqttCallback {

        MqttClient myClient;
        MqttConnectOptions connOpt;

//      IMqttClient publisher = new MqttClient("tcp://iot.eclipse.org:1883",publisherId);

        static final String M2MIO_THING = UUID.randomUUID().toString();
        static final String BROKER_URL = "tcp://test.mosquitto.org:1883";
//      static final String M2MIO_DOMAIN = "<Insert m2m.io domain here>";
//      static final String M2MIO_STUFF = "things";
//      static final String M2MIO_USERNAME = "<m2m.io username>";
//      static final String M2MIO_PASSWORD_MD5 = "<m2m.io password (MD5 sum of password)>";

        // the following two flags control whether this example is a publisher, a
        // subscriber or both
        static final Boolean subscriber = true;
        static final Boolean publisher = true;

        private Random rnd = new Random();

        private static final Logger log = LoggerFactory.getLogger(SimpleMqttClient.class);
        public static final String TOPIC = "grupatras/lab/engine/temperature";

        /**
         *
         * connectionLost This callback is invoked upon losing the MQTT connection.
         *
         */
        public void connectionLost(Throwable t) {
                log.info("Connection lost!");
                // code to reconnect to the broker would go here if desired
        }

        /**
         *
         * deliveryComplete This callback is invoked when a message published by this
         * client is successfully received by the broker.
         *
         */
        public void deliveryComplete(IMqttDeliveryToken token) {

        }

        /**
         *
         * messageArrived This callback is invoked when a message is received on a
         * subscribed topic.
         *
         */
        public void messageArrived(String topic, MqttMessage message) throws Exception {
                log.info("\n");
                log.info("-------------------------------------------------");
                log.info("| Topic:" + topic);
                log.info("| Message: " + new String(message.getPayload()));
                log.info("-------------------------------------------------");
                log.info("\n");
        }
        /**
         *
         * MAIN
         *
         */
        public static void main(String[] args) {
                SimpleMqttClient smc = new SimpleMqttClient();
                smc.runClient();
        }

        /**
         *
         * runClient The main functionality of this simple example. Create a MQTT
         * client, connect to broker, pub/sub, disconnect.
         *
         */
        public void runClient() {
                // setup MQTT Client
                String clientID = M2MIO_THING;
                connOpt = new MqttConnectOptions();

                connOpt.setCleanSession(true);
                connOpt.setKeepAliveInterval(30);
//              connOpt.setUserName(M2MIO_USERNAME);
//              connOpt.setPassword(M2MIO_PASSWORD_MD5.toCharArray());

                // Connect to Broker
                try {
                        myClient = new MqttClient(BROKER_URL, clientID);
                        myClient.setCallback(this);
                        myClient.connect(connOpt);
                } catch (MqttException e) {
                        e.printStackTrace();
                        System.exit(-1);
                }

                log.info("Connected to " + BROKER_URL);

                String myTopic = TOPIC;
                MqttTopic topic = myClient.getTopic(myTopic);

                // subscribe to topic if subscriber
                if (subscriber) {
                        try {
                                int subQoS = 0;
                                myClient.subscribe(myTopic, subQoS);
                                if (!publisher) {
                                        while (true) {
                                                Thread.sleep(1000);
                                        }
                                }
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }

                // publish messages if publisher
                if (publisher) {
                        while (true) {
                                double temp = 80 + rnd.nextDouble() * 20.0;
                                String val = String.format("T:%04.2f", temp);
                                String pubMsg = "{\"value\":" + val + "}";
                                int pubQoS = 0;
                                MqttMessage message = new MqttMessage(pubMsg.getBytes());
                                message.setQos(pubQoS);
                                message.setRetained(false);

                                // Publish the message
                                log.info("Publishing to topic \"" + topic + "\" qos " + pubQoS + "\" value " + val);
```

```
                    MqttDeliveryToken token = null;
                    try {
                            // publish message to broker
                            token = topic.publish(message);
                            // Wait until the message has been delivered to the broker
                            token.waitForCompletion();
                            Thread.sleep(1000);
                    } catch (Exception e) {
                            e.printStackTrace();
                    }
            }
    }

    // disconnect
    try {
            // wait to ensure subscribed messages are delivered
            if (subscriber) {
                    Thread.sleep(5000);
            }
            myClient.disconnect();
    } catch (Exception e) {
            e.printStackTrace();
    }
        }

}
```

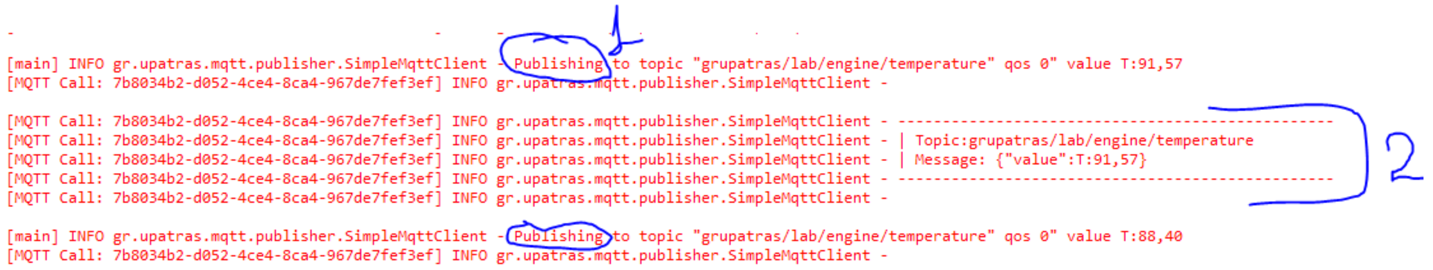
4 Both publisher and subscriber

Watch the lines:

```
static final Boolean subscriber = true;
static final Boolean publisher = true;
```

The program is both publisher and subscriber.

Execute the program. The output is like the following:



```
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:91,57
[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -

[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------------------------
[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Topic:grupatras/lab/engine/temperature
[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Message: {"value":T:91,57}
[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------------------------
[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -

[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:88,40
[MQTT Call: 7b8034b2-d052-4ce4-8ca4-967de7fef3ef] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -
```

It is both publishing a message and also retrieves it.

5 Separate publisher and subscriber

5.1 Run subscriber only

Change the lines:

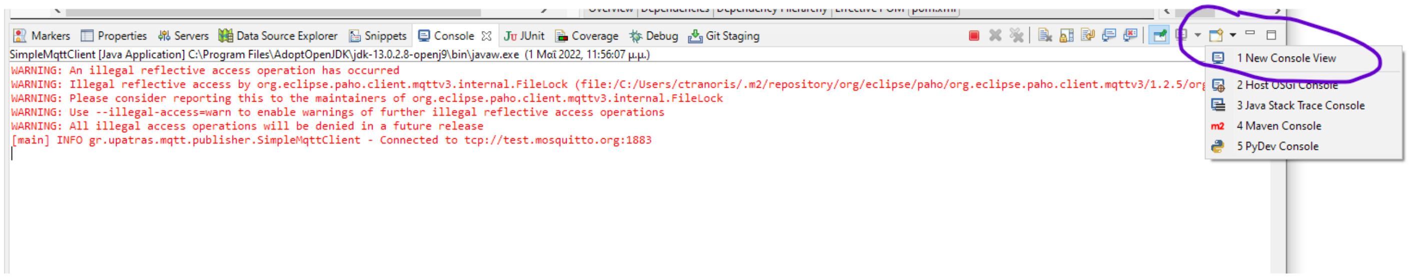
```
static final Boolean subscriber = true;
static final Boolean publisher = false;
```

and run the program. The program waits there:

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.eclipse.paho.client.mqttv3.internal.FileLock (file:/C:/Users/ctranoris/.m2/repository/org/eclipse/paho/org.ecli
WARNING: Please consider reporting this to the maintainers of org.eclipse.paho.client.mqttv3.internal.FileLock
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Connected to tcp://test.mosquitto.org:1883
```

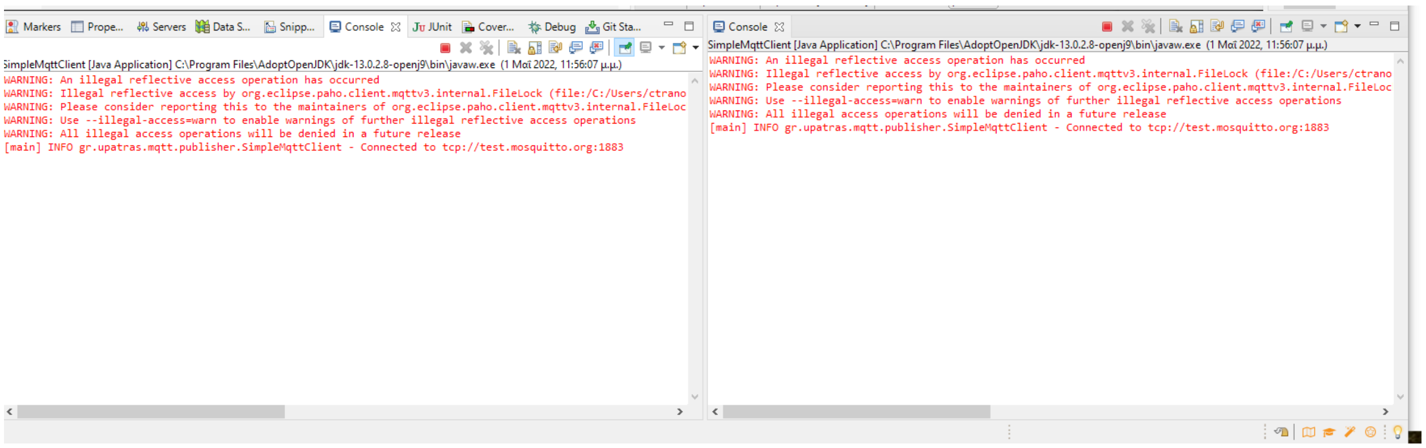
5.2 Run publisher

Open a second console in Eclipse



```
SimpleMqttClient [Java Application] C:\Program Files\AdoptOpenJDK\jdk-13.0.2.8-openj9\bin\javaw.exe (1 Μαΐ 2022, 11:56:07 μ.μ.)
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.eclipse.paho.client.mqttv3.internal.FileLock (file:/C:/Users/ctranoris/.m2/repository/org/eclipse/paho/org.eclipse.paho.client.mqttv3/1.2.5/org
WARNING: Please consider reporting this to the maintainers of org.eclipse.paho.client.mqttv3.internal.FileLock
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Connected to tcp://test.mosquitto.org:1883
```

Split the console like that:



```
SimpleMqttClient [Java Application] C:\Program Files\AdoptOpenJDK\jdk-13.0.2.8-openj9\bin\javaw.exe (1 Μαΐ 2022, 11:56:07 μ.μ.)
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.eclipse.paho.client.mqttv3.internal.FileLock (file:/C:/Users/ctrano
WARNING: Please consider reporting this to the maintainers of org.eclipse.paho.client.mqttv3.internal.FileLoc
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Connected to tcp://test.mosquitto.org:1883
```

```
SimpleMqttClient [Java Application] C:\Program Files\AdoptOpenJDK\jdk-13.0.2.8-openj9\bin\javaw.exe (1 Μαΐ 2022, 11:56:07 μ.μ.)
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.eclipse.paho.client.mqttv3.internal.FileLock (file:/C:/Users/ctrano
WARNING: Please consider reporting this to the maintainers of org.eclipse.paho.client.mqttv3.internal.FileLoc
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Connected to tcp://test.mosquitto.org:1883
```

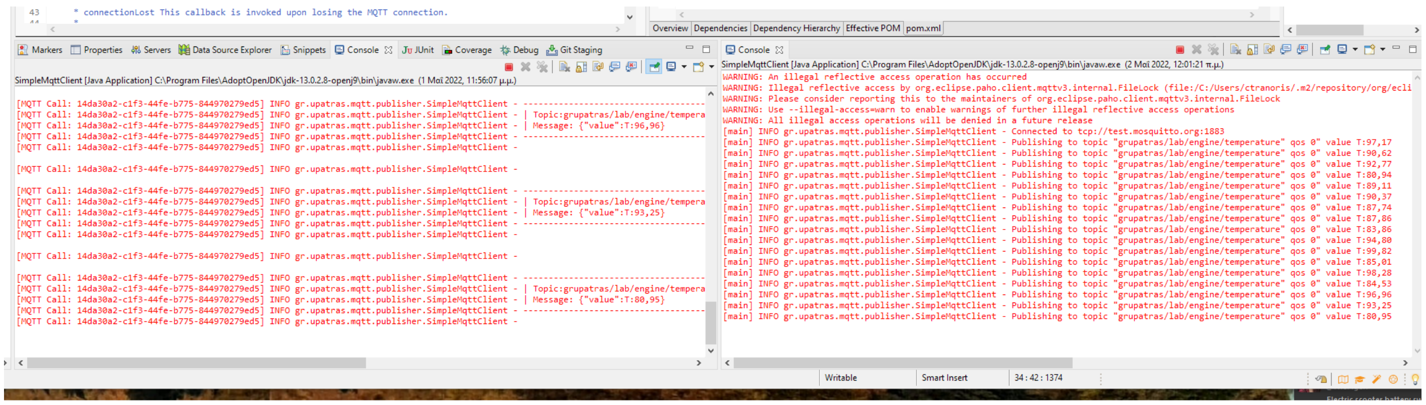
Change the lines:

```
static final Boolean subscriber = false;
static final Boolean publisher = true;
```

DO NOT STOP THE PREVIOUS PROGRAM.

Run the program again.

It is like this. The right console is the publisher of values. The left console show the values that receives:



```
     * connectionLost This callback is invoked upon losing the MQTT connection.
     *
```

SimpleMqttClient [Java Application] C:\Program Files\AdoptOpenJDK\jdk-13.0.2.8-openj9\bin\javaw.exe (1 Μαΐ 2022, 11:56:07 μ.μ.)

```
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Topic:grupatras/lab/engine/tempera
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Message: {"value":T:96,96}
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -

[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -

[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Topic:grupatras/lab/engine/tempera
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Message: {"value":T:93,25}
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -

[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -

[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Topic:grupatras/lab/engine/tempera
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - | Message: {"value":T:80,95}
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - ----------------------------------
[MQTT Call: 14da30a2-c1f3-44fe-b775-844970279ed5] INFO gr.upatras.mqtt.publisher.SimpleMqttClient -
```

SimpleMqttClient [Java Application] C:\Program Files\AdoptOpenJDK\jdk-13.0.2.8-openj9\bin\javaw.exe (2 Μαΐ 2022, 12:01:21 π.μ.)

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.eclipse.paho.client.mqttv3.internal.FileLock (file:/C:/Users/ctranoris/.m2/repository/org/ecli
WARNING: Please consider reporting this to the maintainers of org.eclipse.paho.client.mqttv3.internal.FileLock
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Connected to tcp://test.mosquitto.org:1883
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:97,17
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:90,62
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:92,77
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:80,94
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:89,11
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:90,37
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:87,74
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:87,86
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:83,86
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:94,80
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:99,82
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:85,01
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:98,28
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:84,53
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:96,96
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:93,25
[main] INFO gr.upatras.mqtt.publisher.SimpleMqttClient - Publishing to topic "grupatras/lab/engine/temperature" qos 0" value T:80,95
```