

```

package main.java.com.wfdai.weatherforecastdai.main;

import org.eclipse.paho.client.mqttv3.MqttClient; import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
import org.eclipse.paho.client.mqttv3.MqttException; import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;

/** * Envia mensagens para o Broker  @author daniel */ public class Publisher
{

    String topic;
    String content;
    MemoryPersistence persistence;

    public Publisher() {
    }

    /**
     * Recebe uma mensagem e envia-a para o broker
     *
     * @param mensagem String com o conteudo da mensagem.
     * @param topico String com o tópic do Broker.
     */
    public void publish(String topico, String mensagem) {
        this.publish(topico, mensagem, true);
    }

    /**
     * Recebe uma mensagem e envia-a para o broker
     *
     * @param mensagem String com o conteudo da mensagem.
     * @param topico String com o tópic do Broker.
     * @param retained boolean com opção de reter a mensagem
     */
    public void publish(String topico, String mensagem, boolean retained) {
        int qos = 2;
        String broker = "tcp://127.0.0.1:9001";
        String clientId = "javaServer";
        this.topic = topico;
        this.content = mensagem;
        this.persistence = new MemoryPersistence();

        try {
            MqttClient sampleClient = new MqttClient(broker, clientId, persistence);
            MqttConnectOptions connOpts = new MqttConnectOptions();
            connOpts.setCleanSession(true);
            System.out.println("Connecting to broker: " + broker);

```

```

        sampleClient.connect(connOpts);
        System.out.println("Connected");
        System.out.println("Publishing message: " + content);
        MqttMessage message = new MqttMessage(content.getBytes());
        message.setQos(qos);
        message.setRetained(retained);
        sampleClient.publish(topic, message);
        System.out.println("Message published");
        sampleClient.disconnect();
        System.out.println("Disconnected");
        //System.exit(0);
    } catch (MqttException me) {
        System.out.println("reason " + me.getReasonCode());
        System.out.println("msg " + me.getMessage());
        System.out.println("loc " + me.getLocalizedMessage());
        System.out.println("cause " + me.getCause());
        System.out.println("excep " + me);
    }
}
}

```