

```

package com.dai.webServer.Mqtt;

/*****
 * Licensed under the Apache License, Version 2.0 (the "License"); * you may not
 * use this file except in compliance with the License. * You may obtain a copy of
 * the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless re-
 * quired by applicable law or agreed to in writing, software * distributed under the
 * License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. * See the License for
 * the specific language governing permissions and * limitations under the License.
 *****/

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;

public class Publisher {

    /**
     * The main method.
     *
     * @param args the arguments
     */

    public static void sendMessage(String topic, String message, String id) { String
mqttTopic = topic; String content = message; int qos = 2; String broker =
"tcp://alvesvitor.ddns.net:80"; String clientId = id; MemoryPersistence persis-
tence = new MemoryPersistence();

        try {

            MqttClient sampleClient = new MqttClient(broker, clientId, persistence);
            MqttConnectOptions connOpts = new MqttConnectOptions();
            connOpts.setUsername("dai");
            String password = "12345678";

            connOpts.setPassword(password.toCharArray());
            connOpts.setCleanSession(true);
            System.out.println("Connecting to broker: " + broker);
            sampleClient.connect(connOpts);
            System.out.println("Connected");
            System.out.println("Publishing message: " + content);
            MqttMessage mqttMessage = new MqttMessage(content.getBytes());
            mqttMessage.setQos(qos);
            sampleClient.publish(mqttTopic, mqttMessage);
            System.out.println("Message published");
            sampleClient.disconnect();
            System.out.println("Disconnected");
        }
    }
}

```

```
    } 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);  
        me.printStackTrace();  
    }  
}  
}
```