

```
2  * To change this license header, choose License Headers in Project Properties.
6 package it.uniroma1.msecs.ay2019.e3;
7
8 import java.sql.*;
17
18 /**
19  *
20  * @author biar
21  */
22 public class FlightListener implements MessageListener {
23     private Connection connection;
24     private TopicConnection jmsConnection;
25     private final static Pattern statusPattern = Pattern.compile(".* : (.*)");
26
27
28     public FlightListener() {
29         try {
30             Properties props = new Properties();
31             props.setProperty(Context.INITIAL_CONTEXT_FACTORY,
32 "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
33             props.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
34             Context ctx = new InitialContext(props);
35             ConnectionFactory connectionFactory = (ConnectionFactory)ctx.lookup
36 ("ConnectionFactory");
37             jmsConnection = (TopicConnection)connectionFactory.createConnection();
38             TopicSession session = (TopicSession) jmsConnection.createSession(false,
39 Session.AUTO_ACKNOWLEDGE);
40             Destination destination = (Destination)ctx.lookup("dynamicTopics/Flights");
41             TopicSubscriber subscriber = session.createSubscriber((Topic)destination);
42             subscriber.setMessageListener(this);
43
44             Class.forName("org.sqlite.JDBC");
45         } catch (JMSEException | NamingException | ClassNotFoundException err) {
46             err.printStackTrace();
47             System.exit(1);
48         }
49
50
51     public void onMessage(Message msg) {
52         try {
53             String flight = msg.getStringProperty("flight");
54             String text = ((TextMessage)msg).getText();
55
56             Matcher matcher = statusPattern.matcher(text);
57             if (matcher.find()) {
58                 String status = matcher.group(1);
59
60                 PreparedStatement statement = connection.prepareStatement("INSERT INTO
61 flights VALUES(?, ?)");
62                 statement.setQueryTimeout(30);
63
64                 statement.setString(1, flight);
65                 statement.setString(2, status);
66                 statement.executeUpdate();
67
68                 System.out.println(String.format("%s -> %s", flight, status));
69             }
70         } catch (JMSEException | SQLException err) {
71             err.printStackTrace();
72         }
73     }
74 }
```

```
71     }
72 }
73
74
75 public void start() {
76     try {
77         jmsConnection.start();
78         connection = DriverManager.getConnection("jdbc:sqlite:/home/biar/se-
2019_09.db");
79     } catch (JMSEException | SQLException err) {
80         err.printStackTrace();
81     }
82 }
83
84
85 public void stop() {
86     try {
87         jmsConnection.stop();
88         if (connection != null) {
89             connection.close();
90         }
91     } catch (JMSEException | SQLException err) {
92         err.printStackTrace();
93     }
94 }
95 }
96
```