

Software Engineering - 2019.09.10

' Messages

A JMS producer publishes data about flights in `dynamicTopics/Flights` .

Each message has a `flight` property and a text body in the form `Item <number> >> flight <flight> has landed : <status>` . `number` is a sequential message id, `status` can either be `true` or `false` .

Example:

```
Item 11 >> flight AZ742 has landed : true
Item 12 >> flight JF384 has landed : true
Item 13 >> flight PH327 has landed : false
Item 14 >> flight FD902 has landed : true
```

' Client

You need to implement a client that inserts in table `flights(flight TEXT, status TEXT)` the flight id and the status of all received flights. Each message must be written separately, flights can not be memorized in batch.

' Database

The provided class `DatabaseManager` can be used to manage the database. You can implement your own class to perform the same actions, but it must use the database file specified in `DatabaseManager` .

If `DatabaseManager` is called with argument `create` , it creates and initialized the database. If it is called with argument `run` , it prints the content of table `flights` .

To perform operations on the database, you need to add the dependency:

```
<dependency>
  <groupId>org.xerial</groupId>
  <artifactId>sqlite-jdbc</artifactId>
  <version>3.7.2</version>
</dependency>
```

' DatabaseManager

```
import java.sql.*;
```

```
public class DatabaseManager {
    public static void main(String[] args) throws ClassNotFoundException {
        if (args.length < 1) {
            System.out.println("Pass 'create' to initialize the database, 'run' to print
the content of the database");
            System.exit(1);
        }

        Class.forName("org.sqlite.JDBC");
        Connection connection = null;

        try {
            connection = DriverManager.getConnection("jdbc:sqlite:/home/earendil/se-
2019_09.db");
            Statement statement = connection.createStatement();
            statement.setQueryTimeout(30);

            if (args[0].equals("create")) {
                statement.executeUpdate("DROP TABLE IF EXISTS flights;");
                statement.executeUpdate("CREATE TABLE flights (flight STRING, status
STRING);");
                statement.executeUpdate("INSERT INTO flights VALUES('AA123',
'landed');");
            } else{
                ResultSet rs1 = statement.executeQuery("SELECT * FROM flights");
                while (rs1.next()) {
                    System.out.println(String.format("%s : %s", rs1.getString("flight"),
rs1.getString("status")));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (connection != null) {
                    connection.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```