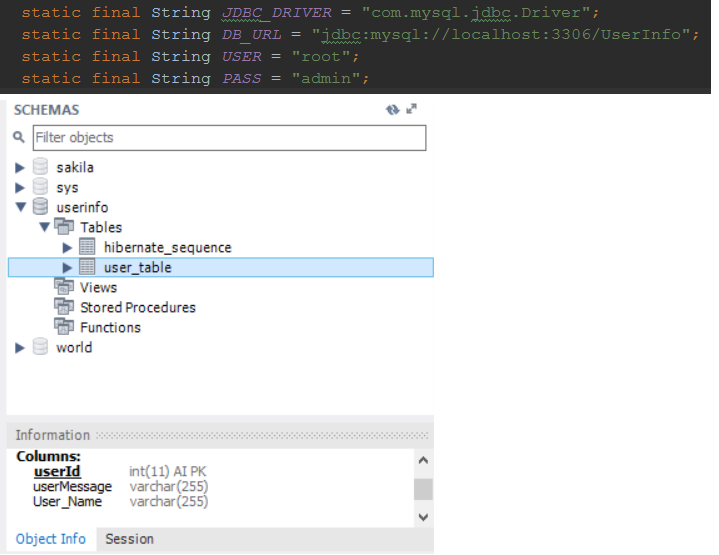
# **XML2DB : Simple Java app using Swing, JAXB and JDBC for persisting XML fields into DB.**

**Installation:**

1. Import project from 7zip archive into InteliJ Idea.
2. Install MySQL Workbench (client and server) and create a new local database with default name and “root”/”admin” as credentials.

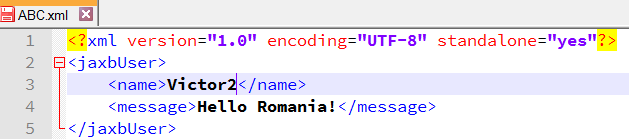


1. Create schema “userinfo” and import the “user\_table” from supplied CSV file.
2. Copy ABC.XML file to local disk.

**Use Case:**

1. **Edit the saved XML file and enter 2 values for the Name and Message:**

**Example:**

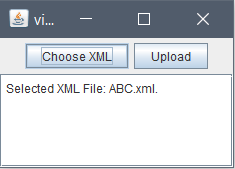
****

1. **Run GUI class from the Project.**

**Expected: A Swing windows appears.**

1. **Click “Choose XML”, navigate to the location where you saved the XML file. Select it and click “Open”**

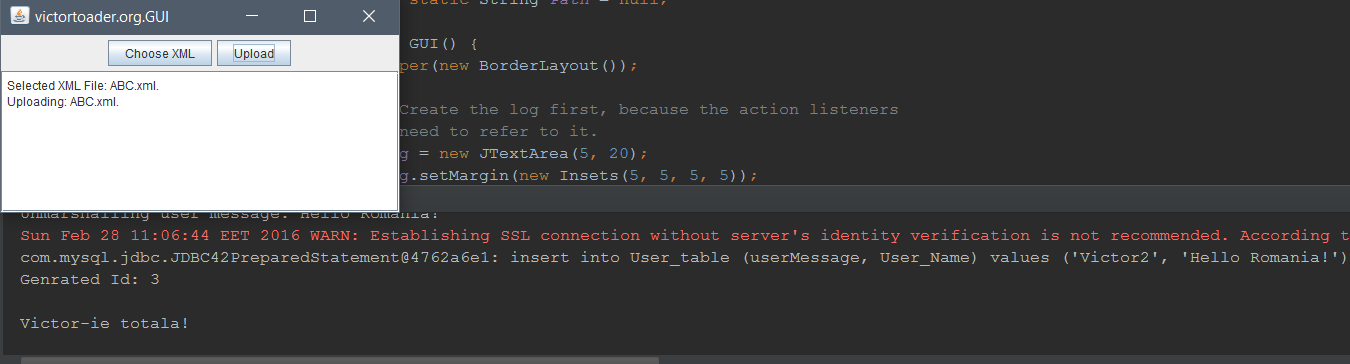
**Expected: The UI confirms the file selected.**

****

1. **In UI click the “Upload” button.**

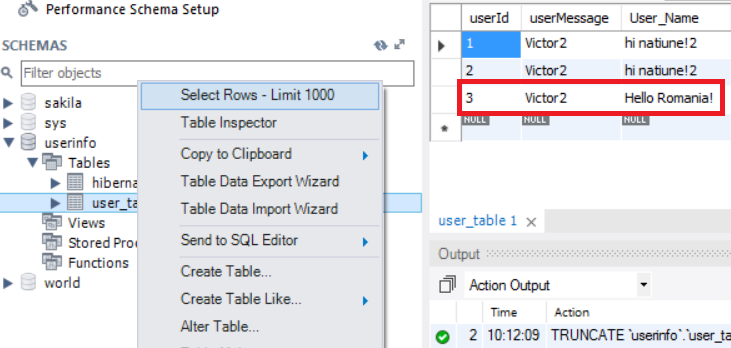
**Expected: UI confirms the uploading of the file.**

**In the InteliJ Console you can see the insert sql statement and the generated ID.**

****

1. **Go to MySQL Workbench and run a select \* on the “user\_table” table.**

**Expected: You can see that both fields that were entered at first step in the XML file are now a row in the DB.**

****

**Technologies used:**

* Swing with BorderLayout as Layout Manager and FileChooser component.
* JAXB for Unmarhsalling the XML file.
* JDBC using PreparedStatement with “?” wildcards and .setString(<position>, <value to replace>) methods in order to protect against SQL Injection attacks.

**Inspired by the following:**

* **FileChooser tutorial by Oracle:** [**https://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html**](https://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html)
* **JAXB Tutorials by Mykong.com:**

**http://www.mkyong.com/java/jaxb-hello-world-example/**