Name: Tan Le Xiong

Matric No: B032310533

**Swing Client App.java**

import java.awt.EventQueue;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.io.\*;

import java.net.HttpURLConnection;

import java.net.URL;

import org.json.JSONArray;

import org.json.JSONObject;

import java.awt.event.ActionListener;

public class JokeClientApp {

private JFrame frame;

private JTextField textFieldCategory;

private JTextField textFieldType;

private JTextField textFieldReligiousCount;

private JTextArea textArea;

private int religiousCount = 0;

private String lastCategory = "";

private String lastType = "";

public static void main(String[] args) {

EventQueue.*invokeLater*(() -> {

try {

JokeClientApp window = new JokeClientApp();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

});

}

public JokeClientApp() {

initialize();

}

private void initialize() {

frame = new JFrame();

frame.setTitle("Joke API Client");

frame.setBounds(100, 100, 600, 450);

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frame.getContentPane().setLayout(null);

JLabel lblJoke = new JLabel("Joke From Web Service:");

lblJoke.setBounds(10, 40, 200, 13);

frame.getContentPane().add(lblJoke);

JLabel lblCategory = new JLabel("Category Name:");

lblCategory.setBounds(50, 270, 100, 20);

frame.getContentPane().add(lblCategory);

textFieldCategory = new JTextField();

textFieldCategory.setBounds(170, 271, 200, 20);

frame.getContentPane().add(textFieldCategory);

JLabel lblType = new JLabel("Type:");

lblType.setBounds(50, 303, 100, 20);

frame.getContentPane().add(lblType);

textFieldType = new JTextField();

textFieldType.setBounds(170, 304, 200, 20);

frame.getContentPane().add(textFieldType);

JLabel lblReligiousCount = new JLabel("Religious Count:");

lblReligiousCount.setBounds(50, 336, 100, 20);

frame.getContentPane().add(lblReligiousCount);

textFieldReligiousCount = new JTextField();

textFieldReligiousCount.setBounds(170, 337, 80, 20);

frame.getContentPane().add(textFieldReligiousCount);

JButton btnGetJoke = new JButton("Get Joke From Web Service");

btnGetJoke.setBounds(163, 10, 250, 25);

frame.getContentPane().add(btnGetJoke);

JButton btnSave = new JButton("Save To Database");

btnSave.setBounds(345, 334, 150, 25);

frame.getContentPane().add(btnSave);

textArea = new JTextArea();

textArea.setBounds(10, 63, 523, 172);

frame.getContentPane().add(textArea);

// Action: Get joke

btnGetJoke.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Thread thread = new Thread(() -> {

try {

URL url = new URL("https://v2.jokeapi.dev/joke/Any");

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod("GET");

BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));

StringBuilder json = new StringBuilder();

String line;

while ((line = reader.readLine()) != null) {

json.append(line);

}

JSONObject response = new JSONObject(json.toString());

// Display the complete JSON object

String formattedJson = response.toString(4); // 4 spaces for indentation

// Process the joke

religiousCount = 0;

final boolean[] isFiltered = {false};

final String[] jokeText = {""};

// Check flags

JSONObject flags = response.getJSONObject("flags");

if (flags.getBoolean("nsfw")) {

religiousCount++;

}

// Check if joke should be filtered

if (!flags.getBoolean("religious") &&

!flags.getBoolean("racist") &&

!flags.getBoolean("sexist")) {

lastCategory = response.getString("category");

lastType = response.getString("type");

if (response.getString("type").equals("single")) {

jokeText[0] = response.getString("joke");

} else {

jokeText[0] = response.getString("setup") + "\n" + response.getString("delivery");

}

} else {

isFiltered[0] = true;

}

// Update UI

SwingUtilities.*invokeLater*(() -> {

textArea.setText(formattedJson +

"\n\n--- Filtered Joke ---\n" +

(isFiltered[0] ? "Joke was filtered due to content flags" :

"Category: " + lastCategory + "\n" +

"Type: " + lastType + "\n" +

"Joke: " + jokeText[0]));

textFieldCategory.setText(lastCategory);

textFieldType.setText(lastType);

textFieldReligiousCount.setText(String.*valueOf*(religiousCount));

});

} catch (Exception ex) {

ex.printStackTrace();

SwingUtilities.*invokeLater*(() -> {

JOptionPane.*showMessageDialog*(frame, "Failed to fetch jokes: " + ex.getMessage());

});

}

});

thread.start();

}

});

// Action: Save to database

btnSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (lastCategory.isEmpty() || lastType.isEmpty()) {

JOptionPane.*showMessageDialog*(frame, "No valid joke data to save!");

return;

}

Thread thread = new Thread(() -> {

try {

URL url = new URL("http://localhost:3306/JokeAPI/joke");

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setDoOutput(true);

conn.setRequestMethod("POST");

conn.setRequestProperty("Content-Type", "application/json");

JSONObject data = new JSONObject();

data.put("category", lastCategory);

data.put("type", lastType);

data.put("religiousCount", religiousCount);

OutputStream os = conn.getOutputStream();

os.write(data.toString().getBytes());

os.flush();

BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));

StringBuilder resp = new StringBuilder();

String line;

while ((line = in.readLine()) != null) {

resp.append(line);

}

SwingUtilities.*invokeLater*(() -> {

JOptionPane.*showMessageDialog*(frame, "Saved successfully!\nResponse: " + resp.toString());

});

} catch (Exception ex) {

ex.printStackTrace();

SwingUtilities.*invokeLater*(() -> {

JOptionPane.*showMessageDialog*(frame, "Failed to save to local API: " + ex.getMessage());

});

}

});

thread.start();

}

});

}

}

Output:

A screenshot of a computer

AI-generated content may be incorrect.