import java.io.IOException;

import java.io.Reader;

import java.io.StringReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.ArrayList;

import java.util.Scanner;

import javax.swing.Box;

import javax.swing.BoxLayout;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.net.MalformedURLException;

import org.json.simple.JSONObject;

import org.json.simple.JSONArray;;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JSeparator;

import javax.swing.JTextField;

import javax.swing.JTextArea;

/\*\*

\* Project

\* The Java Project as a part of JAVA submission at VJTI

\*

\* This project searches for the book from google books

\* @author Pranay Gupta

\* @version 2.0

\*/

public class Project {

static JFrame window;

static JScrollPane bookPanel;

static JPanel books;

static JTextField searchBar;

static JButton searchBtn;

static JLabel status;

static JPanel topPanel;

public void initComponents(){

topPanel = new JPanel();

topPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 20, 20));

window = new JFrame();

bookPanel = new JScrollPane();

bookPanel.setBackground(Color.RED);

bookPanel.setForeground(Color.RED);

status = new JLabel("Status");

searchBar = new JTextField();

searchBar.setPreferredSize(new Dimension(200, searchBar.getPreferredSize().height));

searchBar.setToolTipText("Search book name");

searchBtn = new JButton("Search");

searchBtn.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

searchBtn\_onClick(e);

}

});

}

public Project(){

initComponents();

window.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

window.setSize(400,600);

window.setTitle("Search Books");

window.setVisible(true);

window.setLayout(new BorderLayout());

window.add(topPanel, BorderLayout.NORTH);

topPanel.add(searchBar);

window.add(bookPanel,BorderLayout.CENTER);

topPanel.add(searchBtn);

window.add(status,BorderLayout.SOUTH);

}

/\*\*

\* The code for the search button

\* @param e The event object that calls the event. Specified Implicitly

\*/

public void searchBtn\_onClick(ActionEvent e){

String query = searchBar.getText().replaceAll(" ", "%20");

Fetch fetch = new Fetch((Object a) ->{

ArrayList<String> arr = (ArrayList<String>) a;

books = new JPanel();

books.setLayout(new BoxLayout(books, BoxLayout.Y\_AXIS ));

bookPanel.getViewport().add(books);

for( int i=0;i< arr.size();i++){

System.out.println(arr.get(i));

JTextArea area = new JTextArea(arr.get(i));

area.setLineWrap(true);

books.add(area);

books.add(new JSeparator());

books.add(Box.createVerticalGlue());

window.repaint();

}

}, query);

fetch.start();

status.setText("Fetching results for "+ searchBar.getText());

}

/\*\*

\* The thread Class for fetching details over HTTP

\* @author Pranay

\*

\* @see Callback

\*

\*/

static class Fetch extends Thread{

Callback call;

String query ;

/\*\*

\* The thread that make request over HttpUrlConnection

\*

\* <br>Create object of this thread by passing the instance of Callback with overriden

\* run function and the search query

\* @param call The callback object executed once result is obtained

\* @param query the bookname to be searched

\*/

public Fetch(Callback call, String query){

this.call = call;

this.query = query;

}

/\*\*

\* The function that actually calls the api and pass the result to callback function

\*/

public void run(){

try {

HttpURLConnection conn;

ArrayList<String> ans = new ArrayList<String>();

URL url = new URL("https://www.googleapis.com/books/v1/volumes?q="+query );

conn = (HttpURLConnection) url.openConnection();

Scanner sc = new Scanner(conn.getInputStream());

String JSON = "";

while(sc.hasNext())

JSON +=sc.nextLine();

System.out.println(JSON);

JSONArray arr = (JSONArray) ( (JSONObject)new JSONParser().parse(JSON)).get("items");

for (Object ar : arr){

JSONObject el = (JSONObject) ((JSONObject) ar).get("volumeInfo");

ans.add( el.get("title").toString() + "\n" +el.get("authors") + "\n"+ ((String)el.get("description")) );

}

status.setText("Showing results for "+ query.replaceAll("%20", " "));

call.run(ans);

} catch (MalformedURLException ex) {

status.setText(ex.getMessage());

} catch (IOException ex) {

status.setText(ex.getMessage());

} catch (ParseException ex) {

status.setText(ex.getMessage());

}

}

}

/\*\*

\* The interface for Callback

\* @author Pranay Gupta

\* @see Fetch

\*/

static interface Callback{

/\*\*

\* Override this function for callback

\* @param result Contains the result of Books passed by Fetch thread

\*/

public void run(Object result);

}

public static void main(String[] args) {

Project app = new Project();

}

}