**Task 2**

**write a java application that consumes a public rest API (e.g fetching weather data)and displays the data in a structured format**

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import org.json.JSONObject;

public class WeatherApp {

private static final String BASE\_URL = "https://api.openweathermap.org/data/2.5/weather?q=London&appid=89b323eafb36eb898b6fcd9d727b0dcb&units=metric";

private static final String API\_KEY = "89b323eafb36eb898b6fcd9d727b0dcb";

public static void main(String[] args) {

// Replace with the desired city

String city = "London";

try {

String apiUrl = String.format("%s?q=%s&appid=%s&units=metric", BASE\_URL, city, API\_KEY);

// Make the API request

String response = sendGetRequest(apiUrl);

// Parse and display the weather data

parseAndDisplayWeather(response);

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

// Method to send GET requests

private static String sendGetRequest(String apiUrl) throws Exception {

URL url = new URL(apiUrl);

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

connection.setRequestMethod("GET");

int responseCode = connection.getResponseCode();

if (responseCode == 200) {

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

StringBuilder response = new StringBuilder();

String line;

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

response.append(line);

}

in.close();

return response.toString();

} else {

throw new Exception("Failed to fetch data: HTTP code " + responseCode);

}

}

// Method to parse and display weather data

private static void parseAndDisplayWeather(String response) {

JSONObject jsonResponse = new JSONObject(response);

// Extracting weather data

String cityName = jsonResponse.getString("name");

JSONObject main = jsonResponse.getJSONObject("main");

double temperature = main.getDouble("temp");

int humidity = main.getInt("humidity");

JSONObject weather = jsonResponse.getJSONArray("weather").getJSONObject(0);

String description = weather.getString("description");

// Displaying data

System.out.println("Weather Data:");

System.out.println("-------------");

System.out.println("City: " + cityName);

System.out.println("Temperature: " + temperature + "°C");

System.out.println("Humidity: " + humidity + "%");

System.out.println("Condition: " + description);

}

}

**Dependacies:**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.mycompany</groupId>

<dependencies>

<!-- JSON library -->

<dependency>

<groupId>org.json</groupId>

<artifactId>json</artifactId>

<version>20210307</version>

</dependency>

</dependencies>

<artifactId>mavenproject3</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>jar</packaging>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>22</maven.compiler.source>

<maven.compiler.target>22</maven.compiler.target>

<exec.mainClass>com.mycompany.mavenproject3.Mavenproject3</exec.mainClass>

</properties>

</project>

**Execution Steps:**

1.Create maven project in Netbeans

2.Add Dependacies in pom.xml files located in maven project section

3.Create API key if dose not have from OpenWeatherMap

4. Write a code to display weather data

**Output:**

****