JAVA MINI Assignment 2

Problem Statement: Weather Forecast Aggregator

Description:

Your task is to build a Weather Forecast Aggregator application using Java 8, Spring Boot, REST. The application should fetch weather forecast data from multiple weather APIs concurrently and aggregate the results to provide a comprehensive weather forecast.

Retrieve Weather Info for a Specified City: Implement the below endpoint for the given City & Zip Code

- **URI: http://localhost:8090/weather?city={cityName}&zip={zip code},{country code}**
- **Sample URL: http://localhost:8090/weather?city=Noida&zip=110025,IN**

Requirements:

- Implement a Spring Boot application that acts as a weather forecast aggregator.
- Configure the application to fetch weather forecast data from multiple weather APIs (e.g., OpenWeatherMap, AccuWeather, etc.).
- Fetch weather forecast data for a specific location (e.g., city) from each weather API concurrently.
- Use RESTful endpoints to retrieve the weather forecast for a given location.
- Aggregate the weather forecast data from all the APIs into a single response.
- Provide comprehensive weather forecast information including temperature, humidity, wind speed, precipitation, etc.
- Implement error handling and graceful degradation for failed API calls.
- Utilize Java 8 features such as lambdas, streams, and Completable Future for concurrent processing.

Solution Approach:

Step 1:

Fetch weather forecast data from AccuWeather by calling the below APIs through Spring boot.

API KEY : O3pWXByyepG5BeukHHQAZ4wlZVtV32TC

To call the AccuWeather APIs, we require to pass the API-KEY, you can also use the above-one, or you can create a new developer account as well on AccuWeather to get the new API Key: https://developer.accuweather.com/apis

JAVA MINI Assignment 2

API 1: GET Location KEY (will pass the two params – City & Api-Key)

https://dataservice.accuweather.com/locations/v1/search?q=**Noida**&apikey=**O3pWXByyepG5BeukHHQAZ4wlZVtV32TC**

Response: Save the value of "KEY" param, as we must pass the same in NEXT API.


```
[
    {
        "Version": 1,
        "Key": "3146227",
        "Type": "City",
        "Rank": 35,
        "LocalizedName": "Noida",
        "EnglishName": "Noida",
        "PrimaryPostalCode": "",
        "Region": {
            "ID": "ASI",
            "LocalizedName": "Asia",
            "EnglishName": "Asia"
        },
        "Country": {
            "ID": "IN",
            "LocalizedName": "India",
            "EnglishName": "India"
```


API 2: Get Weather Information using Location KEY which we get from API 1.

https://dataservice.accuweather.com/currentconditions/v1/**3146227**?apikey=**O3pWXByyepG5BeukHHQAZ4wlZVtV32TC**

Response:


```
{
    "LocalObservationDateTime": "2023-06-29T13:13:00+05:30",
    "EpochTime": 1688024580,
    "WeatherText": "Clouds and sun",
    "WeatherIcon": 4,
    "HasPrecipitation": false,
    "PrecipitationType": null,
    "IsDayTime": true,
    "Temperature": {
        "Metric": {
            "Value": 32.1,
            "Unit": "C",
            "UnitType": 17
        },
        "Imperial": {
            "Value": 90.0,
            "Unit": "F",
            "UnitType": 18
```


Note: This API will have details like Weather, Temperature, precipitation and Is Daytime.

JAVA MINI Assignment 2

Step 2:

Fetch weather forecast data from Open Weather by calling the below APIs through Spring boot.

API-KEY: 69017e0b57a66b777aec1aec85cb05a0

To call the Open Weather APIs, we require to pass the API-KEY, you can also use the above-one, or you can create a new developer account as well on Open Weather:
https://openweathermap.org/api

API 1: Geocoder API for getting values for LAT & LONG with respect to provided City & Zip Code.

URI: http://api.openweathermap.org/geo/1.0/zip?zip={zip code},{country code}&appid={API key}

Sample URL:

https://api.openweathermap.org/geo/1.0/zip?zip=**201014,IN**&appid=**69017e0b57a66b777aec1aec85cb05a0**

Sample Response:

```
{
    "zip": "110025",
    "name": "Defence Colony Tehsil",
    "lat": 28.5672,
    "lon": 77.2725,
    "country": "IN"
}
```

API 2: GET Weather Details

URI: https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}

Parameters

lat, lon	required	Geographical coordinates (latitude, longitude). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use above API for the same

JAVA MINI Assignment 2

appid	required	Your unique API key (you can always find it on your account page under the "API key" tab or you can use given one in this document)

SAMPLE URL : -
https://api.openweathermap.org/data/2.5/weather?**lat=28.53&lon=77.39**&appid=**69017e0b57a66b777aec1aec85cb05a0**

Sample Response:

{

"coord": {

"lon": 77.39,

"lat": 28.53

},

"weather": [{

"id": 804,

"main": "Clouds",

"description": "overcast clouds",

"icon": "04d"

}],

"base": "stations",

"main": {

"temp": 302.34,

"feels_like": 301.86,

"temp_min": 302.34,

"temp_max": 302.34,

"pressure": 999,

"humidity": 39,

"sea_level": 999,

"grnd_level": 977

},

"visibility": 10000,

"wind": {

"speed": 1.06,

JAVA MINI Assignment 2

"deg": 82,

"gust": 3.13

},

"clouds": {

"all": 93

},

"dt": 1688025782,

"sys": {

"type": 1,

"id": 9165,

"country": "IN",

"sunrise": 1687996522,

"sunset": 1688046716

},

"timezone": 19800,

"id": 1278007,

"name": "Badarpur",

"cod": 200

}

Note: This API will have details like Wind speed, Visibility, Temperature Feels Like, Humidity along with Sun Rise and Sun Set Info.

Step 3: Get the required data from Step 1 and Step 2

Note: API calls in Step1 & Step2 should be executed in parallel using executor framework

1. Combine the response from Step 1 & Step 2 in Service Implementation Class.
2. Put the logic to filter the required fields only from both the responses.
    a. Step 1 - *Weather, Temperature, precipitation and Is Daytime.*
    b. Step 2 - *Wind speed, Visibility, Temperature Feels Like, Humidity along with Sun Rise and Sun Set Info.*
3. Transform the final payload as per the response given in point 6 (Use Functional Interfaces of Java8 for transformation).
4. Temperature should be in Celsius only.
5. **Write Junit Test Cases for above Endpoint, covering all the scenarios. Usage of Mockito for mocking the response is mandatory.**

JAVA MINI Assignment 2

6. **Here is the finalized expected response based on the provided data:**

{

"WeatherText": "Clouds and sun",

"HasPrecipitation": false,

"PrecipitationType": null,

"IsDayTime": true,

"Temperature": {

"Value": 32.9,

"Unit": "C"

},

"feels_like": {

"Value": 35.9,

"Unit": "C"

},

"pressure": 999,

"humidity": 39,

"visibility": 10000,

"wind": {

"speed": 1.06,

"deg": 82,

"gust": 3.13

},

"sunrise": 1687996522,

"sunset": 1688046716

}