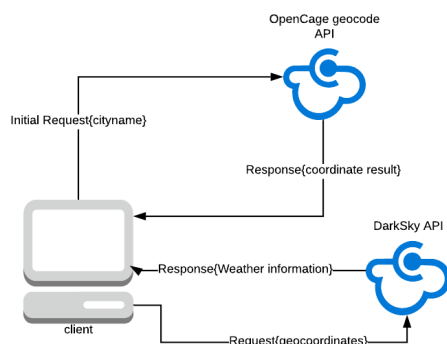# myWeatherApp
By: Mustafa Majeed

Description:

This is a console application that can be executed from the command terminal. It is built using C#/.NET to make http requests to 2 different APIs in order to display weather information about a user specified city in the United States. The city name must be passed to the program as an argument at the same time execution is called. In the occurrence of multiple cities with the same name, weather information about all the cities that share that name will be displayed.

Typing 'myWeatherApp.exe Seattle' into the command terminal when inside the correct directory will result in the following information to be displayed:

```
####################
Seattle, WA
####################
Saturday 10/19/2019
Summary of Today: Overcast
Current Temp: 50°F
High: 52°F
Low: 43°F
Cloud Cover: 89%
This Week's Summary: Rain today through Tuesday, with high temperatures rising to 63°F on Friday.
--------------------
Next 3 days forecast:
--------------------
Sunday 10/20/2019
Light rain in the morning and afternoon.
High: 53°F
Low: 48°F
Chances of rain: 83%
--------------------
Monday 10/21/2019
Light rain throughout the day.
High: 55°F
Low: 50°F
Chances of rain: 86%
--------------------
Tuesday 10/22/2019
Mostly cloudy throughout the day.
High: 57°F
Low: 44°F
Chances of rain: 53%
***********************************************
```

Design

Figure 1 myWeatherApp Design

myWeatherApp is designed to make calls to 2 different APIs to display weather information on the client where it is running. The first call is to OpenCage geocode API sending a request using an endpoint that requires a city name to be included. The response message will be in JSON format and contain multiple results using that city name. OpenCage will match the city name to multiple different types of geolocations which include roads, villages, hospitals, towns.

myWeatherApp is only going to be looking for results of type city AND results that also contain the city name.
As that information is parsed through, myWeatherApp will then make requests to DarkSky API using the endpoint that requires geo coordinates to retrieve and display weather information for each acceptable result from OpenCage API.

DEBUG NOTES FOR CURRENT ISSUES:
- I have not been able to figure our why OpenCage geocode will not recognize New York, or New York City as a city in USA. If you type in the names of any of the boroughs, you will get results. It might have something to do with how I am parsing through the information from the JSON response because I am specifically looking only for type: city AND must contain city: city name as a property. I can get the results that I want that way. I just don't understand why they do not return me a JSON that has the same format when I request geo information about New York or New York City.

Usage
How to use:
Build the app from source code:
- Navigate to the directory where weather_app_builder.bat, Program.cs, GeoLocation.cs, WeatherInfo.cs and myWeatherApp.csproj are stored in the command terminal. Make sure they are all in the same directory if they are not.
- Type 'weather_app_builder.bat' and press Enter.
- The script will try to build myWeatherApp.exe using dotnet build. (Make sure you have .NET installed on your machine)
- After build is successful, the script will navigate you to {currentDirectory}\bin\Debug\netcoreapp3.0
- Then you will be able to type 'myWeatehrApp.exe city name' to run the console application.
Run directly from myWeatherApp.exe:
- Unzip to the Full_weatherApp_Compiled_exe.zip into a separate location. Navigate to that directory in command terminal.
- Type 'cd bin\Debug\netcoreapp3.0' then type the command from the last step above.