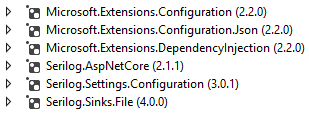
# Framework & Libraries used:

1. Target Framework: .NET Core 2.1
2. Libraries:

**Prudential.DailyWeatherModule**



**Prudential.DailyWeatherModule.Tests**



# Assumptions made:

1. Input file will be in JSON format.
2. Output received from Weather API is to be saved as is i.e. without any other transformations as a **json** file.
3. Application will be single threaded and input file is to be processed synchronously.
4. The format of the input file is expected to be **yyyy-dd-MM\_cities.json** where **yyyy-dd-MM** will be the date (current) when the weather service will run. For e.g. if the weather service is executed on 2nd May 2019, it will expect the input folder to contain a file by the name **2019-02-05\_cities.json**

Sample input files have been copied to **Artifacts\SampleInputFiles** folder.

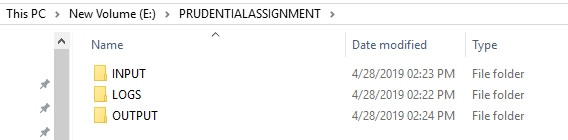
# Other considerations:

1. The main business logic lies in **Prudential.DailyWeatherModule** library. The current client is a .NET Core Console application. However, the same can be easily switched to a Windows service as well.
2. The console application is expected to be executed daily via **Windows Scheduler**.
3. The logic to read from input file could also have been delivered via dependency injection. I chose not to do it as reading data from a large file can prove to be expensive. Hence I opted for **StreamReader** approach instead of **File**.**ReadAllText** approach.

# Instructions before running the project:

1. Ensure **Prudential.DailyWeatherClient** is set as the startup project.
2. Create a folder with the name **‘PRUDENTIALASSIGNMENT’** in any directory.
3. Create 2 folders within **‘PRUDENTIALASSIGNMENT’** folder by the names **‘INPUT’** and **‘OUTPUT’**.

*‘LOGS’ folder will be auto created by Serilog (i.e. the logging framework being used) if it does not exist.*

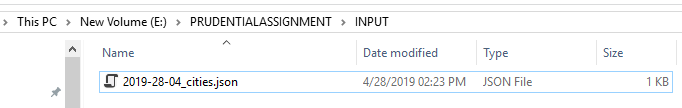


***Note:*** *While it is not necessary to follow naming convention as advised in steps 1 and 2, following the same will help with setting up the project faster.*

1. As seen below, please make sure that all paths (**highlighted below**) are **valid** and point to the **right** folders.



It is expected that the input folder will keep receiving a new file daily with a list of cities in **json** format. The format of this file is expected to be **yyyy-dd-MM\_cities.json** where **yyyy-dd-MM** will be the date (current) when the weather service will run. For e.g. if the weather service is executed on 2nd May 2019, it will expect the input folder to contain a file by the name **2019-02-05\_cities.json**. Sample input files have been copied to **Artifacts\SampleInputFiles** folder.



1. Post successful execution of the program, a new folder (with name being the current date) will be created in the OUTPUT folder and json files with weather information of all cities (mention in the input file) will be created as see under:

