Weather Application Project Usage Guide

The Weather Application Project can calculate when a schedule delay will happen in the construction of the site at 280 Vanderbilt Beach Rd, Naples, FL 34108 due to heavy precipitation.

This project has two methods of informing team members as to when a schedule delay can be claimed.

1) The first way is a small report will be displayed when users log into the system. The welcome panel will display if the conditions are met for schedule delay. The screen will also display the evidence necessary to convince the client.

The tools necessary for the first approach to work is a working connected local msSql database so the users can log in to see the data and a working connection to the internet so that <https://open-meteo.com/> api can be accessed.

2) The Second way occurs after a user creates an account on this system. The system then puts the user in a list of people to email when the precipitation becomes high enough to claim a schedule delay. That email also contains a report with the evidence to convince the client that a schedule delay is necessary. The system is built to, by default, send out an email at 1:00AM everyday where the five day average precipitation meets the condition where schedule delay can be claimed.

The tools necessary for the second approach to work is a working connected local msSql database so the user email can be stored and a working connected SMTP server so the email can be sent. They will also need a working internet connection to be able to access the <https://open-meteo.com/> api.

Problems

Please note as well that it was very hard to find a working and free weather api’s to create this app. Most of the api’s out there that were free have switched to monetizing their information. <https://open-meteo.com/>was the only service I could find with fully free services.

Also note that the <https://open-meteo.com/> api seems to lag behind the present by almost twelve days in terms of its precipitation data. How this manifests in the data is there is a trailing list of nulls in the list of data. I am treating a null data input as a zero value in terms of a precipitation measurement and because of this its likely the last 5 days will always be zero up until the measurements catch up to the present date. This may mean that the email system will never be triggered under standard operating conditions.

Because of the above problems I’ve added a bunch of configuration options to the appsettings.json which will allow the developer to configure the auto emailer. Those settings are as follows:

AlwaysCallback: This condition if set to true will send the email irregardless of the precipitation level. The email itself will contain a message detailing weather a schedule delay can be claimed.

UseEmailDelay: If true then the app enforces the 1:00Am email time but if false then the app will attempt to email every 30 seconds. Please use with caution.

ChangeDay: Allows for the changing of the day the system thinks is the current day. If set to a negative number then the precipitation values will be used for a different day than the current day. This will be useful for checking if a schedule delay could occur on a given past date.

ChangeHour, ChangeMinute: Allows for the precise controlling of when on a particular day the auto emailer will attempt to send out an email. Only works if UseEmailDelay is true.