1. Project outline

To build a data pipeline, Nifi was chosen. Many reasons led to choosing Nifi, mainly it is an open source tool, user-friendly and the volume of data is limited as it is for personal use

1. Data source

The website <https://home.openweathermap.org/> allows streaming real-time weather data through an API, for any desired location. For this project, Montreal, Ottawa, Quebec, and Toronto were the chosen cities.it is possible to have a streaming pipeline but since weather data doesn’t change each second, the pipeline was set to batch with 5min interval.

1. Nifi pipeline

To start Nifi, run-nifi file is started from a cmd. With the credentials in the log file, Nifi is started online.



5 processors were configured

A screenshot of a computer

Description automatically generated

1. InvokeHTTP :

For this processor the parameters of importance to configure were the HTTP URL and the HTTP method.

HTTP URL = http://api.openweathermap.org/data/2.5/weather?lat=45.5019&lon=73.5674&appid=${token} &units=metric

HTTP Method = GET

For the relationships, only Original was set to Terminate.

The token in the HTTP URL is set as follows:

A screenshot of a computer

Description automatically generated

A yellow and grey box with white text

Description automatically generated

1. EvaluateJsonPath

In the properties section :

A screenshot of a computer

Description automatically generated

The destination was set to flowfile-attribute and all the parameters obtained from the website need to have a value configured, as it will help the search for this value in the received files.

As an example of a received input for this processor:

*{"coord":{"lon":73.5674,"lat":45.5019},"weather":[{"id":803,"main":"Clouds","description":"broken clouds","icon":"04n"}],"base":"stations","main":{"temp":19.77,"feels\_like":19.11,"temp\_min":19.77,"temp\_max":19.77,"pressure":1004,"humidity":50,"sea\_level":1004,"grnd\_level":961},"visibility":10000,"wind":{"speed":6.61,"deg":218,"gust":11.1},"clouds":{"all":58},"dt":1714843461,"sys":{"country":"KZ","sunrise":1714780155,"sunset":1714832138},"timezone":18000,"id":1520789,"name":"Mynaral","cod":200}*

1. ReplaceText

the input of this processor is :

{"coord":{"lon":73.5674,"lat":45.5019},"weather":[{"id":803,"main":"Clouds","description":"broken clouds","icon":"04n"}],"base":"stations","main":{"temp":14.77,"feels\_like":14.42,"temp\_min":14.77,"temp\_max":14.77,"pressure":1012,"humidity":81,"sea\_level":1012,"grnd\_level":968},"visibility":10000,"wind":{"speed":5.07,"deg":298,"gust":6.94},"clouds":{"all":80},"dt":1714933413,"sys":{"country":"KZ","sunrise":1714866470,"sunset":1714918615},"timezone":18000,"id":1520789,"name":"Mynaral","cod":200}

the only property that was modified is “Replacement value” under properties:

*city="Montreal",Temperature=${temperature},Wind=${wind},Tempfeeling=${temp\_feeling},Humidity=${humidity},Pressure=${pressure}*

1. ReplaceText

The input of this processor is:

city="Montreal”, Temperature=14.77,Wind=5.07,Tempfeeling=14.42,Humidity=81,Pressure=1012

As date and hour of the data gathered is not mentioned, a column date was added:

A screenshot of a computer

Description automatically generated

Search Value:

city="Montreal", Temperature=${temperature},Wind=${wind},Tempfeeling=${temp\_feeling},Humidity=${humidity},Pressure=${pressure}

was modified with :

city="Montreal", Temperature=${temperature},Wind=${wind},Tempfeeling=${temp\_feeling},Humidity=${humidity},Pressure=${pressure},Time=${now():format("dd/MM/yyyy HH:mm")}

1. InvokeHTTP

The input of this processor is:

city="Montreal", Temperature=14.77, Wind=5.07, Tempfeeling=14.42, Humidity=81,Pressure=1012, Time=05/05/2024 14:27

As for the first processor, the main properties to be configured are HTTP method and HTTP URL, which were set as follow:

HTTP Method: POST, as data will be transferred to a csv file

HTTP URL: <http://127.0.0.1:5000> : specifies the port of data gathering.

1. Collect data in a CSV file

To collect the data, a python file was written with the instructions on how to gather the data in csv file. From a cmd prompt, in the folder containing the python file, the following command is executed : python app-toronto.py.

A screen shot of a computer program

Description automatically generated

The processors setting and the python file were configured for each one of the four cities.

For each cities, a different port was specified in HTTP URL of the last processor, that matched the port specified in the python file

1. Visualize data with Power bi

This is the fun part! To prepare a weather dashboard, data was slightly cleaned by dividing the date column in two, one for the date and one for the hour. The data for the four cities was collected in one single table in power bi (they have separate CSVs)

A close-up of a graph

Description automatically generated