

Article

← [FME Desktop \(/S/Topic/0TO4Q000000QL9uWAG/Fme-...\)](#)

Transforming JSON using the JSONExtractor, JSONFlattener, and JSONFragmenter

🕒 Aug 4, 2022 • Knowledge

Product Type

FME Desktop

FME Version

2022.0

Tutorial: [Tutorial: Getting Started with JSON \(/s/article/tutorial-getting-started-with-json\)](/s/article/tutorial-getting-started-with-json) | **Previous:** [Writing JSON with JSONTemplater \(/s/article/json-writing-with-jsontemplater\)](/s/article/json-writing-with-jsontemplater)

Introduction

Within FME, there are many ways to manipulate JSON to extract the desired data. In this article, we will cover three different transformers and highlight their uses: the JSONFlattener, the JSONExtractor, and the JSONFragmenter. In the first example, we will extract weather information from a JSON API feed from [MetaWeather \(https://www.metaweather.com/\)](https://www.metaweather.com/) using the JSONFlattener and JSONExtractor. In the second example, we will take a JSON file and fragment it into individual attributes containing multiple records using the JSONFragmenter.

Content Overview

- [Part 1: Reading a Simple Message](#)
- [Part 2: Extracting a Complex Message](#)

Step-by-Step Instructions

Part 1: Reading a Simple Message

We will read a JSON message from the [MetaWeather \(https://www.metaweather.com/\)](https://www.metaweather.com/) API using the HTTPCaller and then extract the weather status. This example will show two ways to create attributes from a JSON message. First, we will flatten the JSON using the JSONFlattener to expose all the attributes. Second, we will use the JSONExtractor to extract a particular part of the JSON using a query.

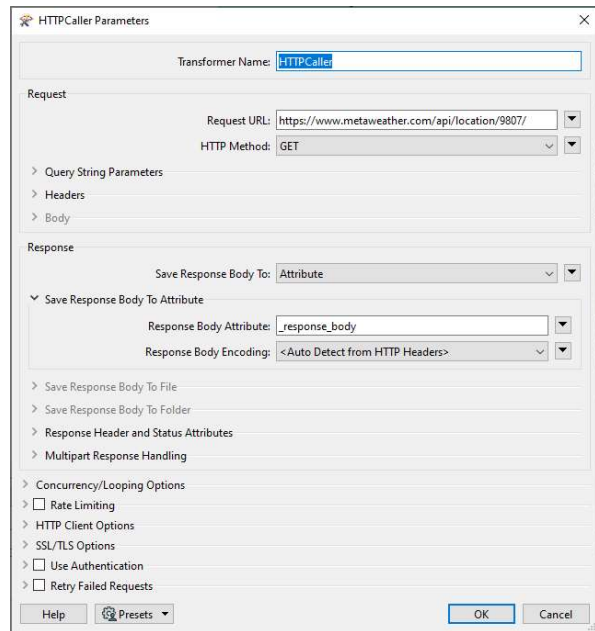
1. Start FME Workbench

Open FME Workbench and start a blank workspace. Add a Creator transformer to the canvas.

Click on the Creator and add an HTTPCaller transformer. The Creator will trigger the HTTPCaller to get information from the URL. In the HTTPCaller parameters, paste in the following URL for the Request URL and set the HTTP Method to GET:

<https://www.metaweather.com/api/location/9807/> (<https://www.metaweather.com/api/location/9807/>).

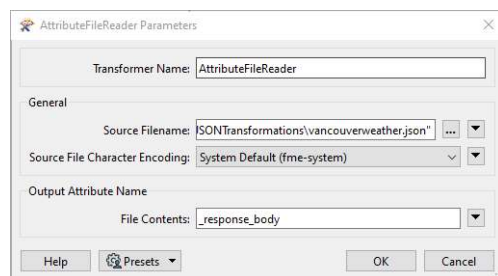
This API call is for Vancouver (location 9807); if you would like to follow this example with your own city, you can [look up the WOEID](http://woeid.rosselliot.co.nz/) (<http://woeid.rosselliot.co.nz/>) and then replace the string of numbers at the end of the URL. Your dialog should look like this:



2. Add an AttributeFileReader

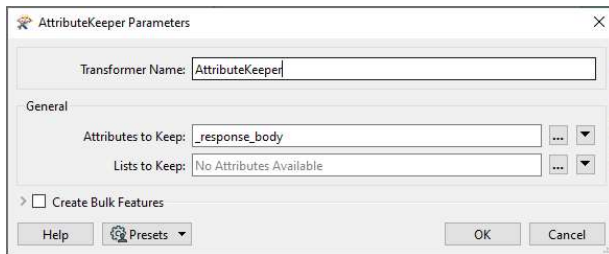
In the event the API is down, add an AttributeFileReader to the Rejected port of the HTTPCaller. This will read the local vancouverweather.json file provided in the Files section of this article.

Open the AttributeFileReader parameters, set the Source File Name to the vancouverweather.json path and then change the File Contents to _response_body. We will use System Default encoding.



3. Select the Attributes to Keep

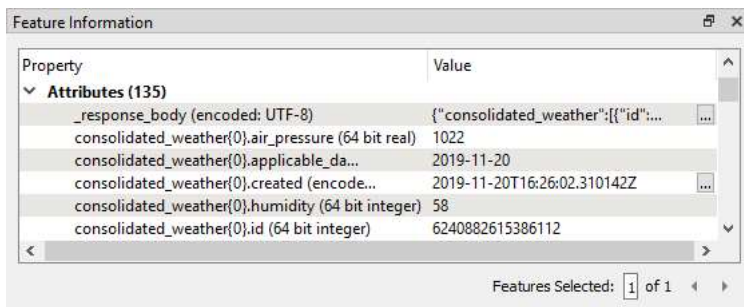
We are only interested in keeping the _response_body attribute from the HTTPCaller or AttributeFileReader. Add an AttributeKeeper transformer to the HTTPCaller Output port and AttributeFileReader Output port. In the parameters, set the Attributes to Keep to _response_body and click OK.



4. Create Attributes with the JSONFlattener

To access individual attributes, e.g., weather status and temperature, we need to flatten the JSON. Add a JSONFlattener transformer and connect it to the AttributeKeeper.In (<http://AttributeKeeper.In>), the parameters, select `_response_body` for the JSON Document.

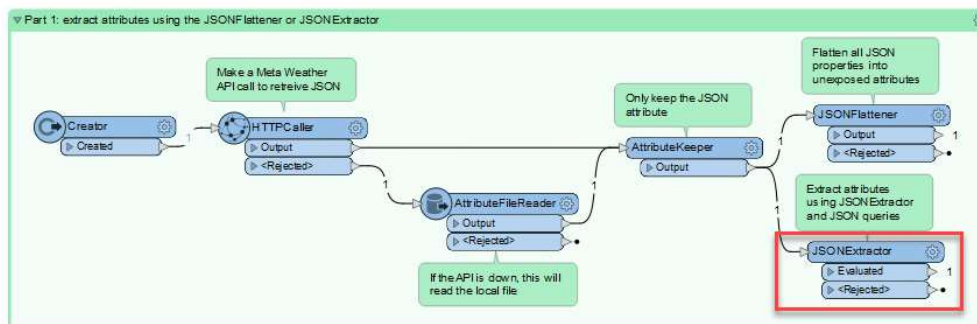
Run the workspace with Feature Caching enabled and select the single feature in Visual Preview. The Feature Information Window shows that `_response_body` has many unexposed attributes (the amount will vary based on the response), containing all the information we need.



The JSONFlattener workflow helps turn all JSON values into FME attributes. The challenge then becomes choosing which attributes to expose using an AttributeExposer.

5. Extract the JSON to Find Today's Weather State Name

If we are interested in one element or property, we can extract that directly into an attribute using the JSONExtractor, instead of flattening the JSON with the JSONFlattener. To find today's weather, we will need to extract the parent element and expose the child element. Connect a JSONExtractor transformer to the AttributeKeeper.



In the JSONExtractor parameters, set the Source JSON Document to `_response_body` and then set the Target Attribute to `Weather`. For JSON Query, enter:

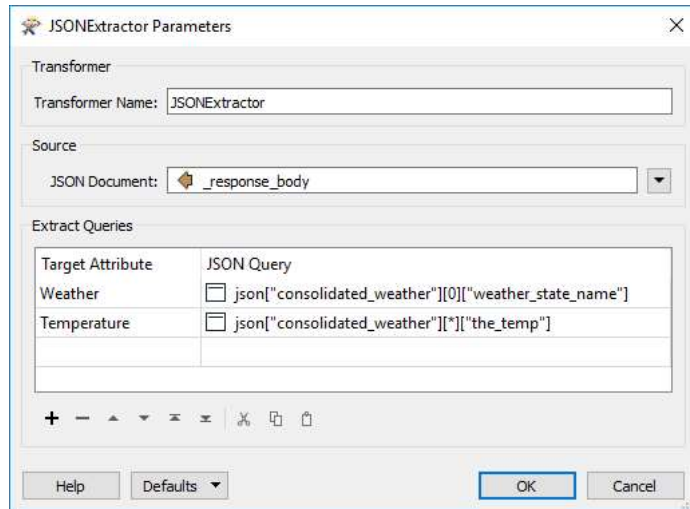
```
json["consolidated_weather"][0]["weather_state_name"]
```

This query will extract the parent element called `consolidated_weather`, then extract the first element in the array (which is 0), then extract the value for the name `weather_state_name`. This API gives us the 5 day forecast, but we are only interested in today's weather. That's why only the first array is queried. It can be challenging to write a JSON query without looking at the JSON structure; we'll look at strategies for writing the query in a later step.

6. Extract the JSON to find today's temperature

Next, we need to extract the temperature. In the same JSONExtractor, set another Target Attribute to Temperature and then set the JSON Query to:

```
json["consolidated_weather"][0]["the_temp"]
```



7. Strategies for Building JSON Queries for JSON Transformers

It can be challenging to find the query to access the weather information. There are two main options here: first, you can take a closer look at the JSON structure using Visual Preview and build your query based on that:

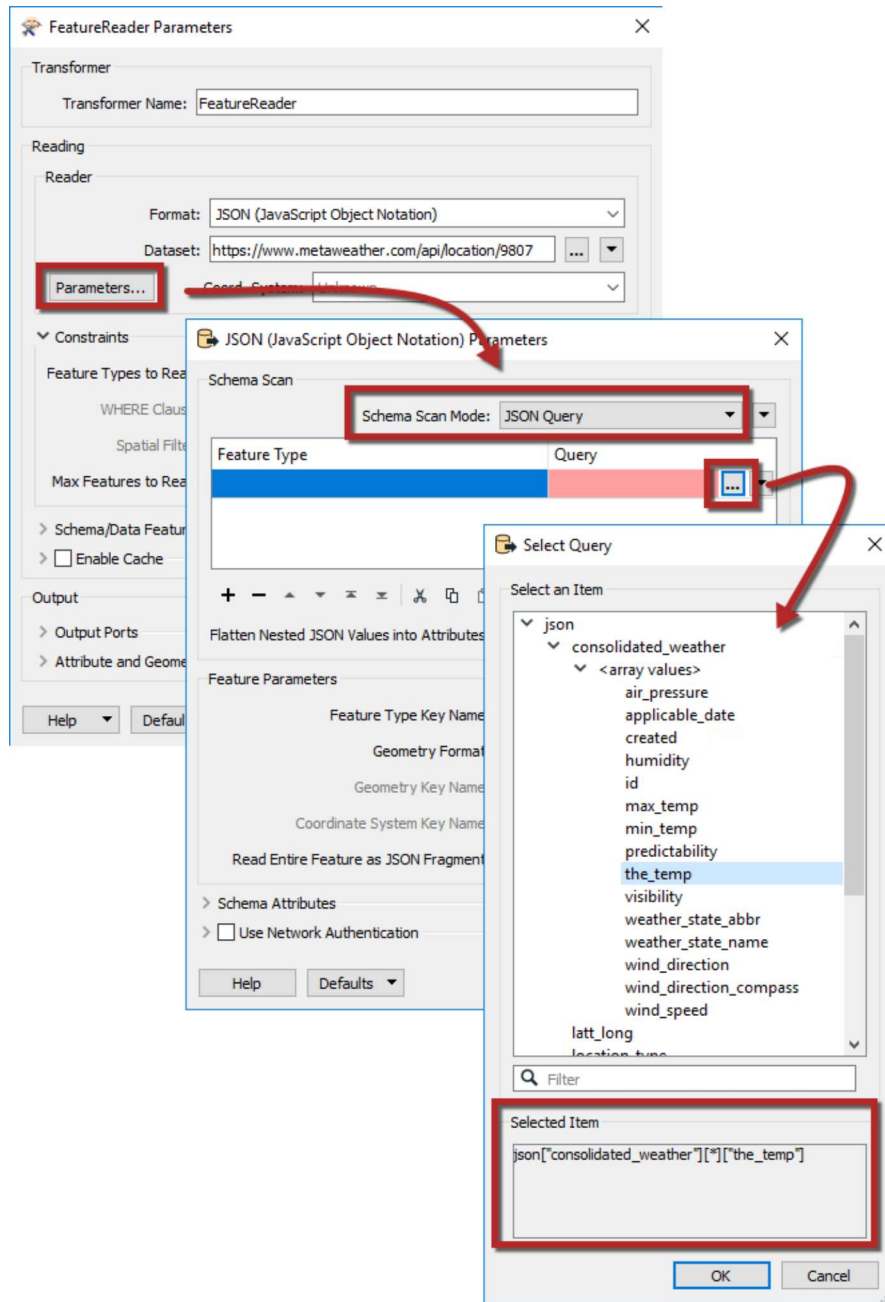
```

{
  "consolidated_weather": [{
    "id": 5097016155701248,
    "weather_state_name": "Heavy Cloud",
    "weather_state_abbr": "hc",
    "wind_direction_compass": "NW",
    "created": "2019-11-01T19:26:02.630529Z",
    "applicable_date": "2019-11-01",
    "min_temp": 1.0,
    "max_temp": 10.805,
    "the_temp": 9.33,
    "wind_speed": 2.5148912446088176,
    "wind_direction": 314.3123566281536,
    "air_pressure": 1030.5,
    "humidity": 64,
    "visibility": 16.250410104986877,
    "predictability": 71
  }, {
    "id": 5428890258374656,
    "weather_state_name": "Light Cloud",
    "weather_state_abbr": "lc",
    "wind_direction_compass": "NW",
    "created": "2019-11-01T19:26:05.847651Z",
    "applicable_date": "2019-11-02",
    "min_temp": 2.88,
    "max_temp": 11.56,
    "the_temp": 10.030000000000001,
    "wind_speed": 1.9563281953789868,
    "wind_direction": 315.1086944400913,
    "air_pressure": 1024.5,
    "humidity": 67,
    "visibility": 15.694282887934463,
    "predictability": 70
  }, {
    ...
  }
}

```

Hint: use a [JSONFormatter](https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/jsonformatter.htm) (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/jsonformatter.htm) to pretty-print your _response_body attribute to make it easier to read in Visual Preview.

The second option is to temporarily add a FeatureReader to the canvas and read in the JSON URL. Click on the parameters and set the Schema Scan Mode to 'JSON Query'. Then under Query, click the ellipses [...] to expose the JSON tree. In the tree, expand JSON, then expand consolidated_weather, and click on the child weather_state_name under. Copy the JSON query that is indicated in the Selected Item box, this query is the same query we input into the JSONExtractor. After you've copied this query you can delete the FeatureReader:



8. Inspect the Data

Run your workspace and inspect the JSONExtractor. You will see that Weather and Temperature attributes have been created and have values extracted from the JSON. The results will vary when you run this translation, as it is a live JSON feed.

Visual Preview

Table

JSONExtractor_Evaluated

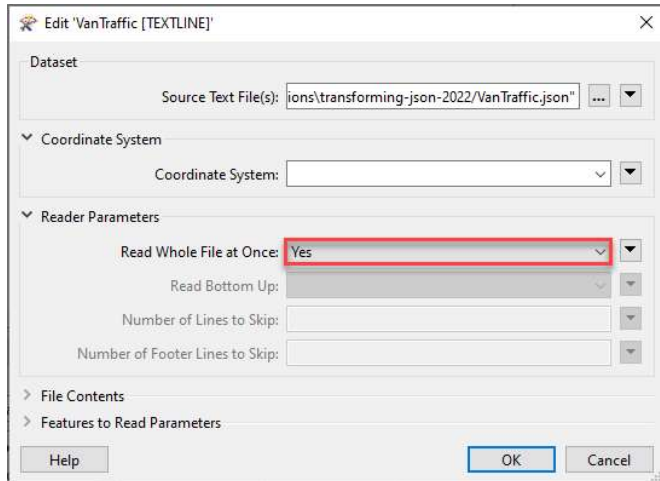
| | _response_body | Weather | Temperature |
|---|-------------------------|-------------|-------------|
| 1 | {\"consolidated_weat... | Heavy Cloud | 10.15 |

Part 2: Extracting a Complex Message

In the previous example, we read in JSON and examined the output using a JSONFlattener and a JSONExtractor. In this example, we will use a JSONFragmenter to work with JSON that is more nested. We will extract latitude and longitude values from JSON to create points.

1. Read in the JSON Using the Text File Reader

In a blank workspace, or in another section of the previous workspace, add a Text File reader. Read in the VanTraffic.json file. In the Parameters, set Read Whole File at Once to Yes, click OK.



Note: When browsing to the JSON file, change the file type in Windows Explorer to All Files (*) to see the JSON file.

The reason we are reading in the JSON data with a Text File reader instead of a JSON reader is because we want the JSON contained within a single attribute called text_line_data. If we read in the JSON using the JSON reader, each line will be a separate record contained within its own attribute, including the brackets.

2. Add a JSONFragmenter to Break the JSON Down into Individual Attributes

Connect a JSONFragmenter to the text_line reader feature type. In the parameters, set the JSON Attribute to text_line_data and set the JSON Query to:

```
json["features"][*]
```

If you prefer, you can build this query using the methods stated in the first part of the article (step 7). Then change the Flatten Query Result Into Attributes to Yes, and then ensure Recursively Flatten Objects/Arrays is set to Yes. Your dialog should look like this:

Transformer Name:

Source

Input Source:

JSON Attribute:

File/URL:

General

JSON Query:

Fragment as Format:

Reject Features which Produce No Fragments:

Result Attribute:

Flattening

Flatten Query Result into Attributes:

Recursively Flatten Objects/Arrays:

Prefix New Attributes With:

Attributes to Expose:

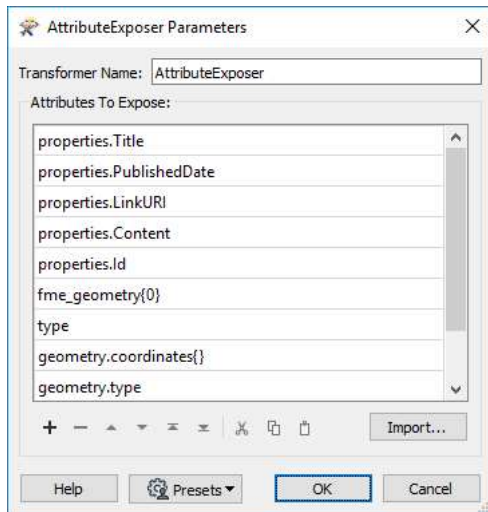
Help Presets OK Cancel

3. Expose Attributes of Interest

Add an AttributeExposer transformer. In its parameters, enter the following attributes.

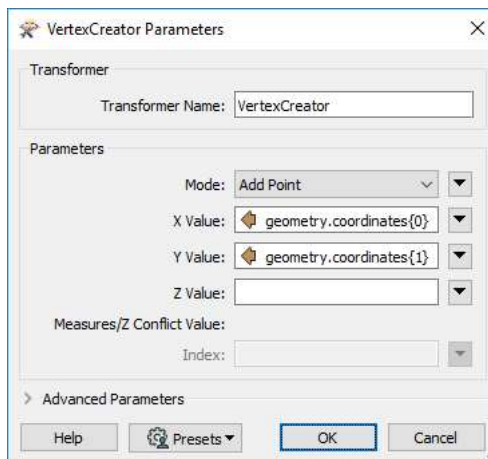
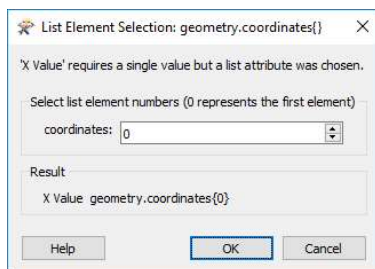
Alternatively, do a partial run and use the ellipses [...] to select the desired attributes.

| |
|--|
| properties.Title |
| properties.PublishedDate |
| properties.LinkURI |
| properties.Content |
| <u>properties.Id</u> (<u>http://properties.Id</u>) |
| fme_geometry{0} |
| type |
| geometry.coordinates{} |
| geometry.type |



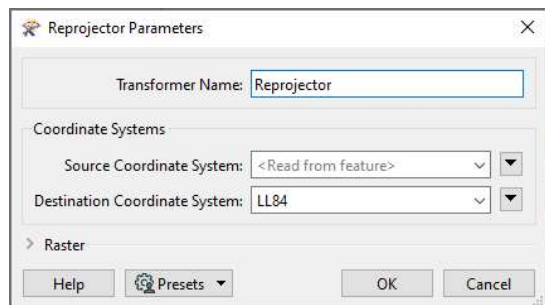
4. Create Points

Add a VertexCreator transformer after the AttributeExposer. In the parameters, set the X Value to the attribute `geometry.coordinates{}`, in the List Element Selection dialog, enter 0 for the coordinates. This will read the first coordinate in the array, which is the longitude value. Repeat this for the Y Value, entering 1 for the coordinates. This will read the second coordinate in the array, which is the latitude value. Your dialog should look like this:



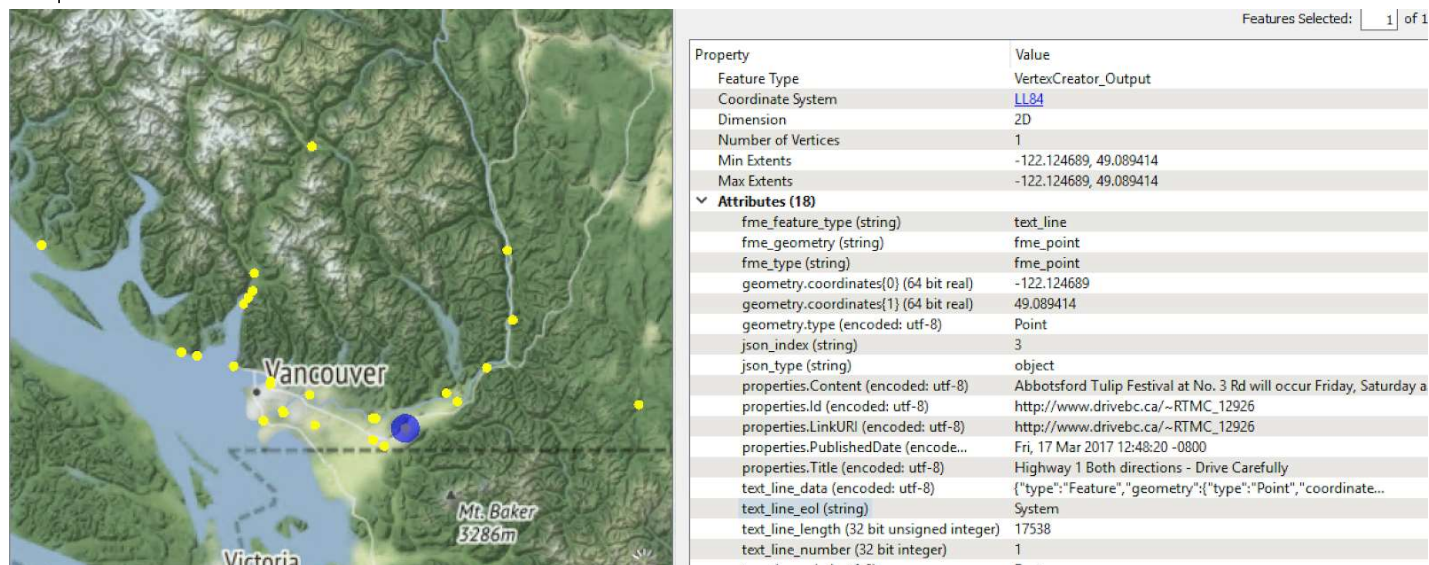
5. Project the Coordinates into LL84

Add a Reprojector transformer after the VertexCreator, and set the Destination Coordinates System to LL84 to reproject the points.



6. Inspect the Results

Connect an Inspector to the Reprojector and run the workspace. Inspecting the output, you can see that each of the attributes we exposed is contained within each point.



Return to the tutorial start: [Tutorial: Getting Started with JSON \(/s/article/tutorial-getting-started-with-json\)](#).

Data Attribution

Part 1: The data used here originates from open data made available by [MetaWeather \(https://www.metaweather.com/\)](https://www.metaweather.com/).

Part 2: The data used here originates from open data made available by the [Government of British Columbia - Drive BC \(http://www.drivebc.ca/\)](http://www.drivebc.ca/).

First Published Date

7/29/2020, 12:18 AM

Last Published Date

8/4/2022, 1:59 AM

Transformation
(/s/topic/OTO4Q000000...)

FME Desktop
(/s/topic/OTO4Q000000...)

Sort by:

Latest Posts ▼



LizAtSafe (/s/profile/0050c00000CeGV0AAN) (Employee) published a new version of this Knowledge.
August 4, 2022 at 1:59 AM (/s/feed/0D54Q00009ggWDySAM).

5 views



Like



Comment

Log In to Comment



kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
[July 19, 2022 at 8:57 PM \(/s/feed/0D54Q000009fDctxSAC\)](/s/feed/0D54Q000009fDctxSAC)

4 views



Like



Comment

Log In to Comment



kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
[July 19, 2022 at 8:35 PM \(/s/feed/0D54Q000009fDbvKSAC\)](/s/feed/0D54Q000009fDbvKSAC)

4 views



Like



Comment

Log In to Comment



kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
[June 30, 2022 at 9:43 PM \(/s/feed/0D54Q000009cP9izSAC\)](/s/feed/0D54Q000009cP9izSAC)

10 views



Like



Comment

Log In to Comment



kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
[June 30, 2022 at 9:00 PM \(/s/feed/0D54Q000009cP7RDSA0\)](/s/feed/0D54Q000009cP7RDSA0)

9 views



Like



Comment

Log In to Comment

Follow

[Files \(3\) \(/s/relatedlist/ka14Q000001DWycQAG/AttachedContentDocuments\)](/s/relatedlist/ka14Q000001DWycQAG/AttachedContentDocuments)



transforming-json-2022
Jul 19, 2022 • 37KB • fmwrt



vantraffic
Jul 28, 2020 • 17KB • json



vancouverweather

Related Articles

Extracting Location from JSON ([/s/article/converting-from-json-to-a-spatial-format-gis](#))



Safe Software respectfully acknowledges that we live, learn and work on the traditional and unceded territories of the Kwantlen, Katzie, and Semiahmoo First Nations.