

Article

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

Reading JSON

🕒 Aug 4, 2022 • Knowledge

Product Type
FME Desktop

FME Version
2022.0

Tutorial: [Tutorial: Getting Started with JSON \(/s/article/tutorial-getting-started-with-json\)](#) | **Next:** [Extracting Location from JSON \(/s/article/converting-from-json-to-a-spatial-format-gis\)](#)

Introduction

This article covers reading JSON data. FME has a few ways to read JSON data into your workspace. Each has different scenarios where they are more useful. Some are specifications of JSON and have their own reader formats, including GeoJSON, Esri JSON, and IMDF, summarized in this table:

Method	Common Scenario	Tutorial
JSON Reader	Reading JSON from a known file path or URL	This page

FeatureReader	Reading JSON from a known or attribute-based file path or URL in the middle of a translation	The FeatureReader Transformer (/s/article/the-featurereader-transformer) .
JSON transformers (JSONExtractor, JSONFlattener, and JSONFragmenter)	Reading JSON directly into a transformer from a file path or URL, controlling how it is turned into FME features	Transforming JSON using the JSONExtractor, JSONFlattener, and JSONFragmenter (/s/article/json-transformations) .
HTTPCaller	Reading JSON from an API call	Tutorial: Getting Started with APIs (/s/article/getting-started-with-apis-1) .
GeoJSON Reader	The source JSON data follows the GeoJSON specification	GeoJSON Reader/Writer Documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/geojson/geojson.htm) .
Esri JSON Reader	The source JSON data follows the Esri JSON specification	Esri-JSON Reader/Writer Documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/esrijson/esrijson.htm?) .
IMDF Reader	The source JSON data follows the IMDF GeoJSON specification	IMDF Reader/Writer Documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/imdf/imdf.htm) .

This article shows how to read JSON directly into your workspace using the JSON reader. It shows two different reader Schema Scan Modes: Automatic and JSON Query. For other scenarios, see the links above.

Content Overview

- [Part 1: Inspect the Data](#)
- [Part 2: The JSON Reader Auto Mode](#)
- [Part 3: The JSON Reader JSON Query Mode](#)

Step-by-Step Instructions

Part 1: Inspect the Data

We access the FAA Airport Status Web Service API in this tutorial. You can access the [API documentation at SwaggerHub](#) (<https://app.swaggerhub.com/apis/FAA/ASWS/1.2.0#/FAA%20Airport%20Status%20Service/getAirportStatus>). `FAA-airport-status-example.json` is a sample extract of the data in case the API is unavailable.

Note: if the URL for this API changes, you can just view the attached example data instead, which is available in the files section of this article.

1. View JSON Data from a URL

Let's start by taking a look at our source JSON data. The example data contains status information about a US airport, including attributes such as the current visibility and weather. We want to read this live data feed for the San Francisco airport (code SFO), which is available from:

<https://soa.smext.faa.gov/asws/api/airport/status/SFO> (<https://soa.smext.faa.gov/asws/api/airport/status/SFO>).

If you open this URL in a browser, you'll see you receive an XML response, as this API provides XML by default. If you wanted to receive JSON, you have to request it in an HTTP header.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<AirportStatus>
  <Name>San Francisco Intl</Name>
  <City>San Francisco</City>
  <State>CA</State>
  <ICAO>KSFO</ICAO>
  <IATA>SFO</IATA>
  <SupportedAirport>true</SupportedAirport>
  <Delay>false</Delay>
  <DelayCount>0</DelayCount>
  <Status>
    <Reason>No known delays for this airport</Reason>
  </Status>
  <Weather>
    <Weather>Mostly Cloudy</Weather>
    <Visibility>10.00</Visibility>
    <Meta>
      <Credit>NOAA's National Weather Service</Credit>
      <Url>https://weather.gov/</Url>
      <Updated>Last Updated on Jul 13 2022, 7:56 am PDT</Updated>
    </Meta>
    <Temp>62.0 F (16.7 C)</Temp>
    <Wind>Southwest at 4.6</Wind>
  </Weather>
</AirportStatus>
```

2. View JSON Data from an HTTP Request with Headers

We can make a proper HTTP request with headers requesting JSON in the browser by using [SwaggerHub](#) (<https://app.swaggerhub.com/apis/FAA/ASWS/1.1.0>), which is an API platform used by the FAA to design and document their API.

Go to the [API's getAirportStatus operation on SwaggerHub](#) (<https://app.swaggerhub.com/apis/FAA/ASWS/1.2.0#/FAA%20Airport%20Status%20Service/getAirportStatus>). Note that it may take a while to load. Then:

(1) Expand the GET `/api/airport/status/{airportCode}` section

FAA Airport Status Service ^

GET `/api/airport/status/{airportCode}` Get airport status based on path parameter provided on the API call. The path parameter is an IATA airport code. ✓ ↶

1

(2) Click the “Try it Out” button, which lets you provide parameters to the `getAirportStatus` operation.

GET `/api/airport/status/{airportCode}` Get airport status based on path parameter provided on the API call. The path parameter is an IATA airport code. ^ ↶

The airport status is retrieved from fly.faa.gov and is provided in a format that can be used by applications that require airport status data. Available airports include the following: BOS, LGA, TEB, EWR, JFK, PHL, PIT, IAD, BWI, DCA, RDU, CLT, ATL, MCO, TPA, MCO, FLL, MIA, DTW, CLE, MDW, ORD, IND, CVG, BNA, MEM, STL, MCI, MSP, DFW, IAH, DEN, SLC, PHX, LAS, SAN, LAX, SJC, SFO, PDX, SEA

Parameters Try it out

Name	Description
airportCode * required	airportCode
string	
(path)	

Responses Response content type `application/json` v

2

(3) Type “SFO” into the `airportCode` parameter field. (4) Click the “Execute” button, which will make an API request using the entered parameter. Note that the default value for “Response content type” is “application/json,” meaning that this API request will provide a header requesting JSON as the response.

GET `/api/airport/status/{airportCode}` Get airport status based on path parameter provided on the API call. The path parameter is an IATA airport code. ^ ↶

The airport status is retrieved from fly.faa.gov and is provided in a format that can be used by applications that require airport status data. Available airports include the following: BOS, LGA, TEB, EWR, JFK, PHL, PIT, IAD, BWI, DCA, RDU, CLT, ATL, MCO, TPA, MCO, FLL, MIA, DTW, CLE, MDW, ORD, IND, CVG, BNA, MEM, STL, MCI, MSP, DFW, IAH, DEN, SLC, PHX, LAS, SAN, LAX, SJC, SFO, PDX, SEA

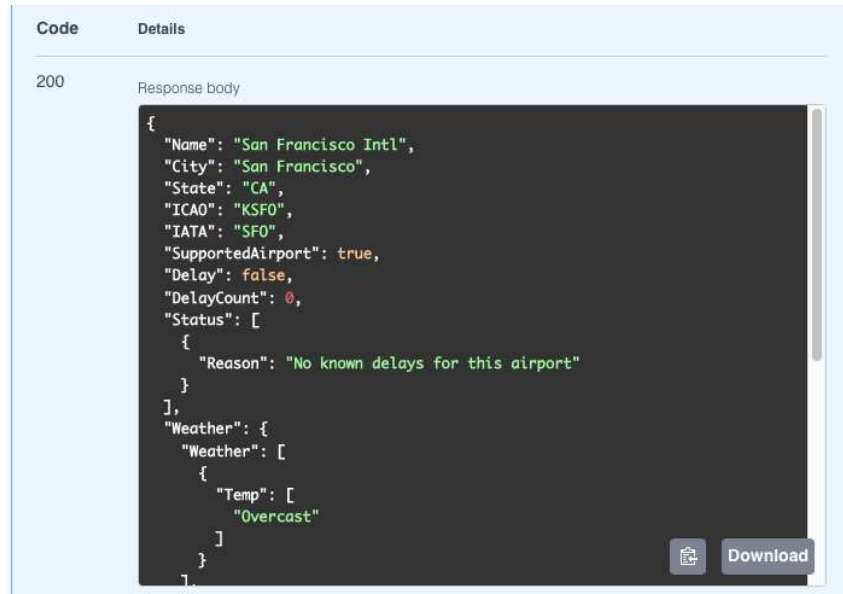
Parameters Cancel

Name	Description
airportCode * required	SFO
string	
(path)	

Execute

3 4

After clicking “Execute,” you will see some JSON appear under the response body for the HTTP 200 code:



The screenshot shows a web interface with two tabs: "Code" and "Details". The "Details" tab is active, displaying a "Response body" section. The response body contains a JSON object with the following structure:

```
{
  "Name": "San Francisco Intl",
  "City": "San Francisco",
  "State": "CA",
  "ICAO": "KSFO",
  "IATA": "SFO",
  "SupportedAirport": true,
  "Delay": false,
  "DelayCount": 0,
  "Status": [
    {
      "Reason": "No known delays for this airport"
    }
  ],
  "Weather": {
    "Weather": [
      {
        "Temp": [
          "Overcast"
        ]
      }
    ]
  }
}
```

At the bottom right of the JSON viewer, there are two buttons: a copy icon and a "Download" button.

This JSON is the data about SFO’s status we’ll be using in the exercise. It should look something like this:

```
{
  "Name": "San Francisco Intl",
  "City": "San Francisco",
  "State": "CA",
  "ICAO": "KSFO",
  "IATA": "SFO",
  "SupportedAirport": true,
  "Delay": false,
  "DelayCount": 0,
  "Status": [
    {
      "Reason": "No known delays for this airport"
    }
  ],
  "Weather": {
    "Weather": [
      {
        "Temp": [
          "Fair"
        ]
      }
    ],
    "Visibility": [
      10
    ],
    "Meta": [
      {
        "Credit": "NOAA's National Weather Service",
        "Url": "http://weather.gov/" (http://weather.gov/),
        "Updated": "Last Updated on Jul 7 2022, 6:56 am PDT"
      }
    ],
    "Temp": [
      "53.0 F (11.7 C)"
    ],
    "Wind": [
      "West at 11.5"
    ]
  }
}
```

JSON Code example

Part 2: The JSON Reader Auto Mode

1. Open FME Workbench and add a JSON Reader

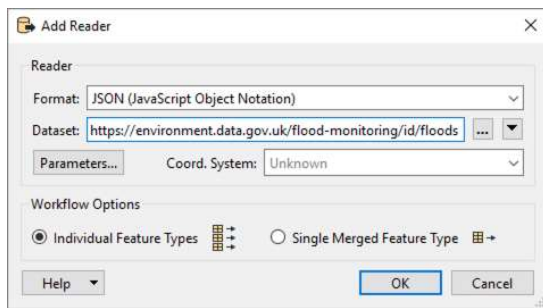
In FME Workbench, start a new workspace. Click Reader > Add Reader. Choose JSON (JavaScript Object Notation) as the Format.

You can supply a path to a local file (example available in the Files) or a URL for the Dataset parameter.

This article will assume you are using a URL. Paste the following URL into the Dataset Parameter:

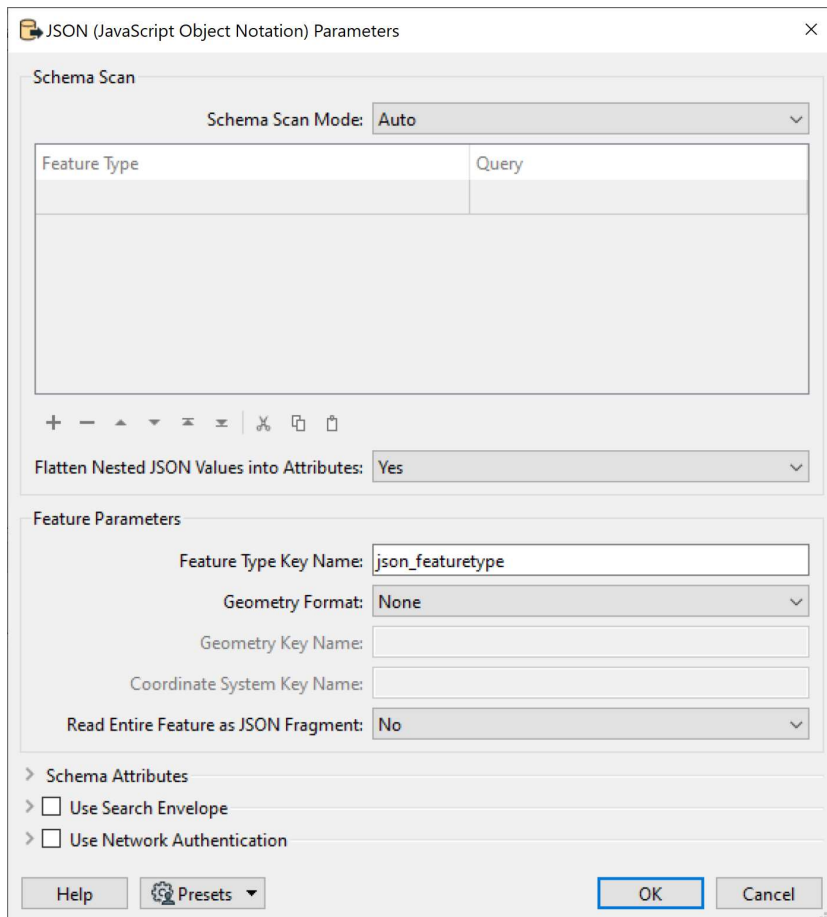
<https://soa.smext.faa.gov/asws/api/airport/status/SFO> (<https://soa.smext.faa.gov/asws/api/airport/status/SFO>).

Although we know this URL responds with XML, FME will automatically include the correct headers to request JSON when using the JSON Reader.



Click the Parameters button. Setting JSON reader parameters lets you control how FME turns the nested structure of JSON into features. There are two options for Schema Scan Mode: Auto and JSON Query. Auto mode simply takes the JSON parent attributes and turns those into FME attributes and turns the rest of the nested structure into FME list attributes. JSON Query mode, which we will examine later in the article, lets you write a JSON Query to pick which part of the JSON data structure you wish to use as the top level to define FME attributes.

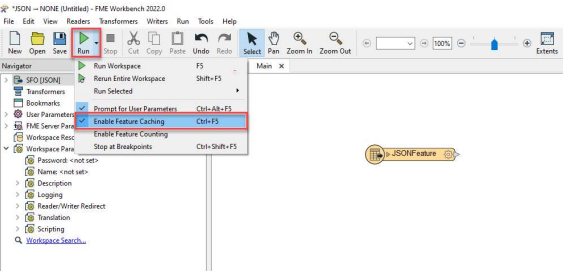
For now, make sure that Schema Scan Mode is set to Auto.



Click OK twice to add the reader.

3. Run Workspace

Ensure that Feature Caching is enabled then click the Run button to run the workspace and create a local feature cache of the JSON data.



4. Inspect the JSON Using Visual Preview

Click on the JSON reader feature type (or its green cache icon) to inspect it using the Table View of Visual Preview. You should see that the JSON data has been flattened into a table, with all of the top-level JSON attributes becoming FME attributes: Name, City, State, etc.

Visual Preview								
Table								
JSONFeature								
	Name	City	State	ICAO	IATA	SupportedAirport	Delay	DelayCount
1	San Francisco Intl	San Francisco	CA	KSFO	SFO	Yes	No	0

You will notice that no attributes with children are included, like “Status” and “Weather.” These attributes are still available on FME features, but because of the nested structure of JSON, they are read as lists.

Note: if you have not worked with unexposed attributes or lists in FME before, working with JSON might be challenging. You can read [Tutorial: Getting Started with List Attributes \(/s/article/working-with-list-attributes-tutorial\)](https://community.safe.com/s/article/working-with-list-attributes-tutorial) for a quick introduction to these topics.

5. View Features in the Feature Information Window

To find these unexposed attributes, click on the single feature in the Table View to select it. Then look in the Feature Information Window in Visual Preview. You can see all the attributes with children are stored on the feature as lists. They use [dot notation \(https://codeburst.io/javascript-quickie-dot-notation-vs-bracket-notation-333641c0f781\)](https://codeburst.io/javascript-quickie-dot-notation-vs-bracket-notation-333641c0f781), where their children are attributes and FME list bracket notation where there is more than one attribute-value pair in the array:

Feature Information

Property	Value
json_type (strim...)	json_no_geom
multi_reader_ju...	0
multi_reader_ke...	0
multi_reader_by...	JSON_1
multi_reader_by...	JSON
Name (strim...)	San Francisco Intl
State (strim...)	CA
Status() Reas...	No known delays for this airport
SupportedAirp...	Yes
Weather.Metal...	NOAA's National Weather Service...
Weather.Metal...	Last Updated on Jun 14 2022, 10...
Weather.Temp[...	https://weather.gov/
Weather.Temp[...	72.0 F (22.2 C)
Weather.Visib...	10
Weather.Wreath...	Fair
Weather.Wind[...	Northwest at 12.7
IFMENull	

Visual Preview

Table								
JSONFeature								
	Name	City	State	ICAO	IATA	SupportedAirport	Delay	DelayCount
1	San Francisco Intl	San Francisco	CA	KSFO	SFO	Yes	No	0

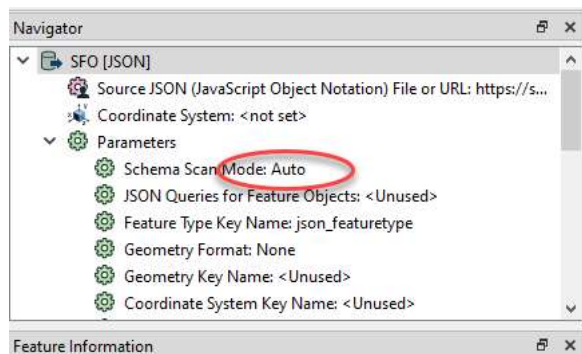
In the next part of this article, you will learn how the JSON structure becomes FME features.

Part 3: The JSON Reader JSON Query Mode

1. Change JSON Reader Schema Scan Mode to JSON Query

Now let's try reading in JSON using a JSON Query.

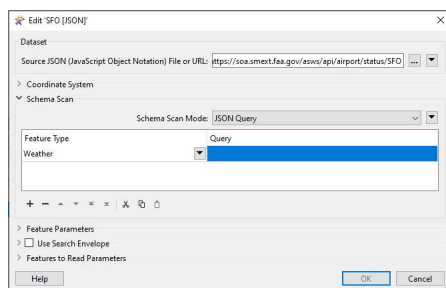
Expand the JSON Reader in the Navigator window, then Parameters, then double-click Schema Scan Mode



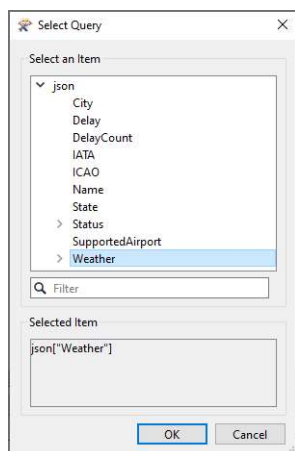
2. Create a JSON Query

To construct a JSONQuery:

1. Change Schema Scan Mode to JSON Query.
2. In the table under Feature Type, type "Weather"



3. Then click in the space under Query and then click on the ellipses button. This button opens the JSON data's nested structure, which you can use to build the query.
4. Click on the arrows that contain the nested list objects. Select Weather from the list; the Query will then display json["Weather"].



5. Click OK. You may have to confirm the Query by clicking any blank spaces in the dialog, for the OK button to enable. The parameters should look as follows:

Edit SFO [JSON]

Dataset

Source JSON (JavaScript Object Notation) File or URL:

Coordinate System

Coordinate System:

Schema Scan

Schema Scan Mode:

Feature Type	Query
<input type="checkbox"/> Weather	<input type="checkbox"/> json["Weather"]

Feature Parameters

Feature Type Key Name:

Geometry Format:

Geometry Key Name:

Coordinate System Key Name:

Read Entire Feature as JSON Fragment:

☐ Use Search Envelope

☐ Features to Read Parameters

Help OK Cancel

3. Run Workspace

Run the workspace and then inspect the new feature cache. The Table View will report the data has no schema, but using the Feature Information Window reveals list attributes. Only the child data of weather and not the parent data such as Airport, City, etc., are included:

Feature Information

Property	Value
Feature Type	Weather
Coordinate System	Unknown
Dimension	2D
Number of Vertices	0
Min Extents	nan, nan
Max Extents	nan, nan

Attributes (11)

fme_feature_ty...	Weather
fme_geometry (strin...	fme_undefined
fme_type (strin...	fme_no_geom
json_type (strin...	json_no_geom
Meta(0).Credit (strin...	NOAA's National Weather Service
Meta(0).Updat...	Last Updated on Jun 14 2012, 11:56 am PT
Meta(0).Url (strin...	https://weather.gov/
Temp(0) (strin...	77.0 F (25.0 C)
Visibility(0) (64 bit real)	10
Weather(0).Temp...	Fair
Wind(0) (strin...	Northwest at 6.9

Features Selected: 1 of 1

Visual Preview

Table

Weather

<no schema>

1

Visual Preview Translation Log

Continue to the next article: [Extracting Location from JSON](https://community.safe.com/s/article/converting-from-json-to-a-spatial-format-gis)
(<https://community.safe.com/s/article/converting-from-json-to-a-spatial-format-gis>).

Data Attribution

The data used in this article come from the [Federal Aviation Administration \(https://www.faa.gov/\)](https://www.faa.gov/)'s [Airport Status Web Service \(ASWS\) API \(https://app.swaggerhub.com/apis/FAA/ASWS/1.1.0\)](https://app.swaggerhub.com/apis/FAA/ASWS/1.1.0), which is licensed [Creative Commons 0 \(CC0\) \(https://creativecommons.org/publicdomain/zero/1.0/legalcode\)](https://creativecommons.org/publicdomain/zero/1.0/legalcode).

First Published Date

7/29/2020, 12:18 AM

Last Published Date

8/4/2022, 1:58 AM


Transformation (/s/topic/0TO4Q000000...)

FME Desktop (/s/topic/0TO4Q000000...)

Sort by:

Latest Posts




 [LizAtSafe \(/s/profile/0050c00000CeGV0AAN\)](/s/profile/0050c00000CeGV0AAN) (Employee) published a new version of this Knowledge.
[August 4, 2022 at 1:58 AM \(/s/feed/0D54Q00009ggWCgSAM\)](#)

1 view

 Like

 Comment

Log In to Comment

 [kailinatsafe \(/s/profile/0054Q00000EwiZmQAJ\)](/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
[July 13, 2022 at 6:18 PM \(/s/feed/0D54Q00009e1wPbSAI\)](#)

1 view


 Like

 Comment

Log In to Comment

Follow

[Files \(1\) \(/s/relatedlist/ka14Q000001DWyIQAW/AttachedContentDocuments\)](/s/relatedlist/ka14Q000001DWyIQAW/AttachedContentDocuments)

 **faa-airport-status-example**
Jul 28, 2020 • 676B • json

[View All](#)

[\(/s/relatedlist/ka14Q000001DWyIQAW/AttachedContentDocuments\)](/s/relatedlist/ka14Q000001DWyIQAW/AttachedContentDocuments)

Related Articles

- Advanced JSON Reading (/s/article/json-reading-advanced)
- Extracting Location from JSON (/s/article/converting-from-json-to-a-spatial-format-gis)
- Tutorial: Getting Started with JSON (/s/article/tutorial-getting-started-with-json)
- Writing JSON with the JSONTemplater (/s/article/json-writing-with-jsontemplater)
- Transforming JSON using the JSONExtractor, JSONFlattener, and JSONFragmenter (/s/article/json-transformations)



SAFE SOFTWARE®

(<https://safe.com>).

[Forums \(/s/forums/\)](#)

[Groups](#)

[Knowledge Base \(/s/knowledge-base/\)](#), ([/s/group/CollaborationGroup/00Ba000000A0BxJEAV](#))

[Support \(/s/support/\)](#)

[Register / Log In \(/s/login/\)](#)



(<http://difficulttofind.com/links/SAFEsoftware/>)

© Safe Software Inc | [Legal \(https://www.safe.com/legal/\)](#)

Land Acknowledgement —

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