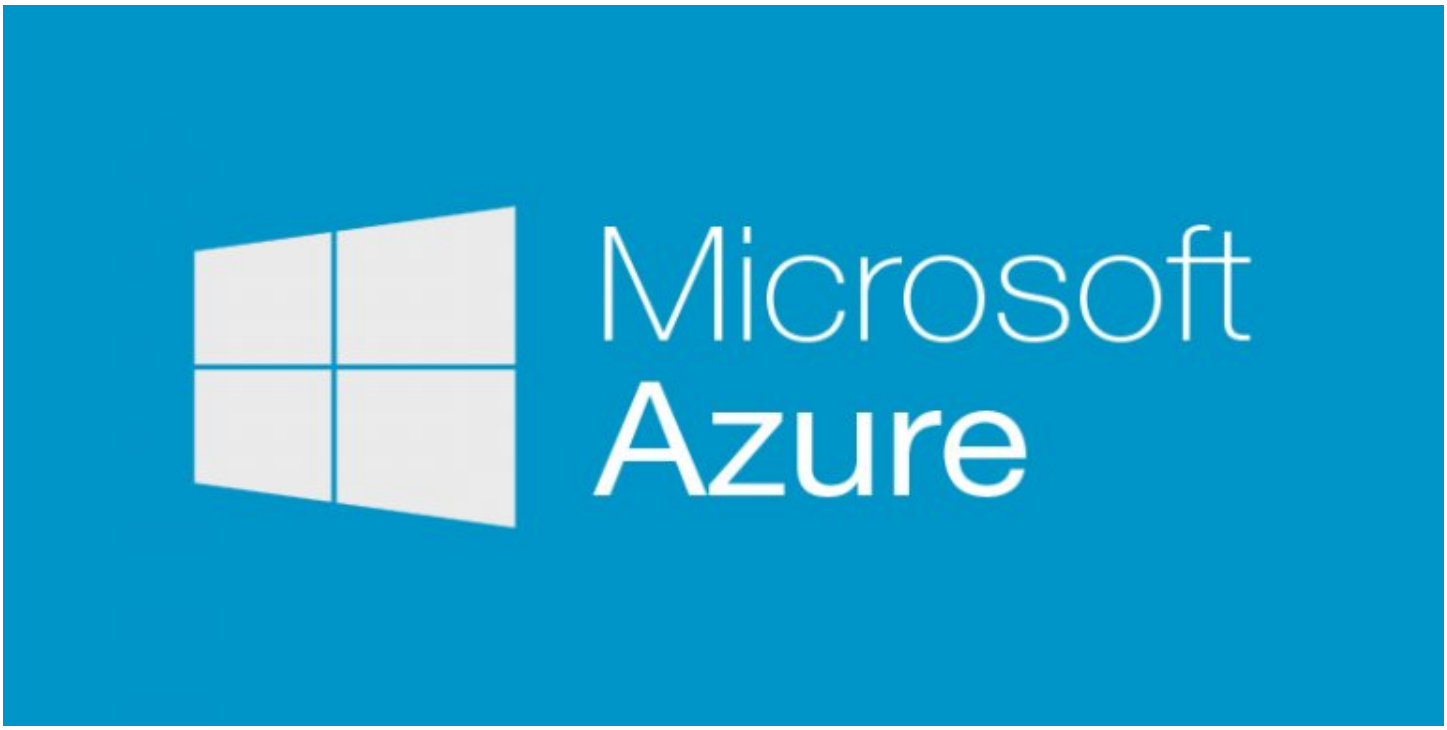
Azure Data Factory (ADF): How to extract JSON data from an API to Azure SQL Server



by Mik Panchal, Managing Consultant- Altis Sydney

This Blog is a step-by-step guide to build the integration pattern used to extract JSON data from an API using Azure Data Factory.

Using an API to extract data in a JSON format has become a common method for extracting data into Azure SQL Database or Azure Blob. As integration tools evolve, greater effort is placed on enabling the tools to extract data from various source systems in different formats, thus enabling Data Engineers to create integration patterns in the most efficient way possible.

WIIFM?

By using the ADF "**REST dataset"** I was able to successfully create a "**copy data**" task to directly map the JSON output to an Azure SQL server table, however, after inspecting the data I noticed ***NULL*** data returned from a **nested** JSON object while **non-nested** JSON object returned the right data.

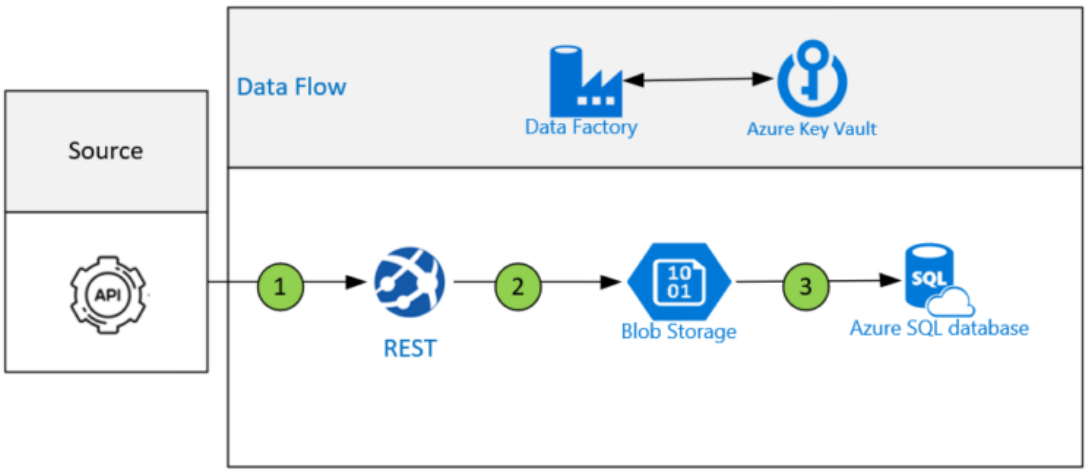
Even though the ADF REST dataset can read the API metadata, the values in these nested JSON objects/arrays are returned as *NULL* when you use a ***Copy data*** task to directly load JSON to Azure SQL Server.

It seems that there is a bug with ADF (v2) when it comes to directly extract a **nested JSON** to Azure SQL Server using the **REST dataset** and **Copy data** task.

Example of nested Json object

```
"latitude":1234567890",
"address":{
    "link":"https://goo.gl/maps/iTeYynDKY9VvkEQD9",
    "value":"Level 6/219 Castlereagh St, Sydney NSW 2000"
},
"street":"Castlereagh Street"
```

High-level data flow using Azure Data Factory



The process involves using ADF to extract data to Blob (.json) first, then copying data from Blob to Azure SQL Server. This additional step to Blob ensures the ADF dataset can be configured to traverse the nested JSON object/array.

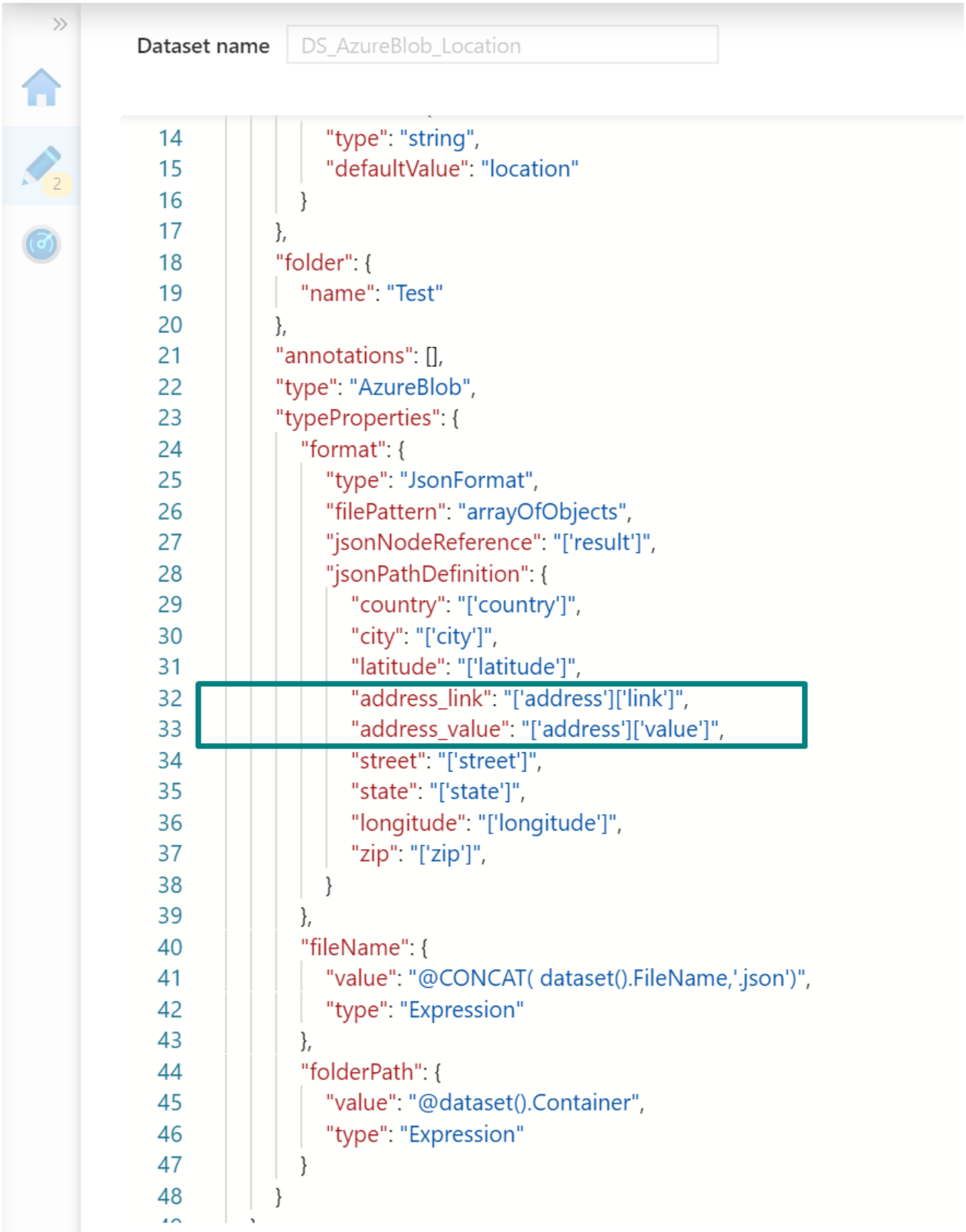
Below is a step-by-step guide to extracting complex JSON data in your Azure platform using Azure Data Factory (ADF).

Follow the steps outlined below:

1. Set up API linked service, and create a REST dataset to API
2. Set up **Azure Blob Storage** dataset to Blob storage "**DS_Source_Location**"
    a. Clear the Shema objects

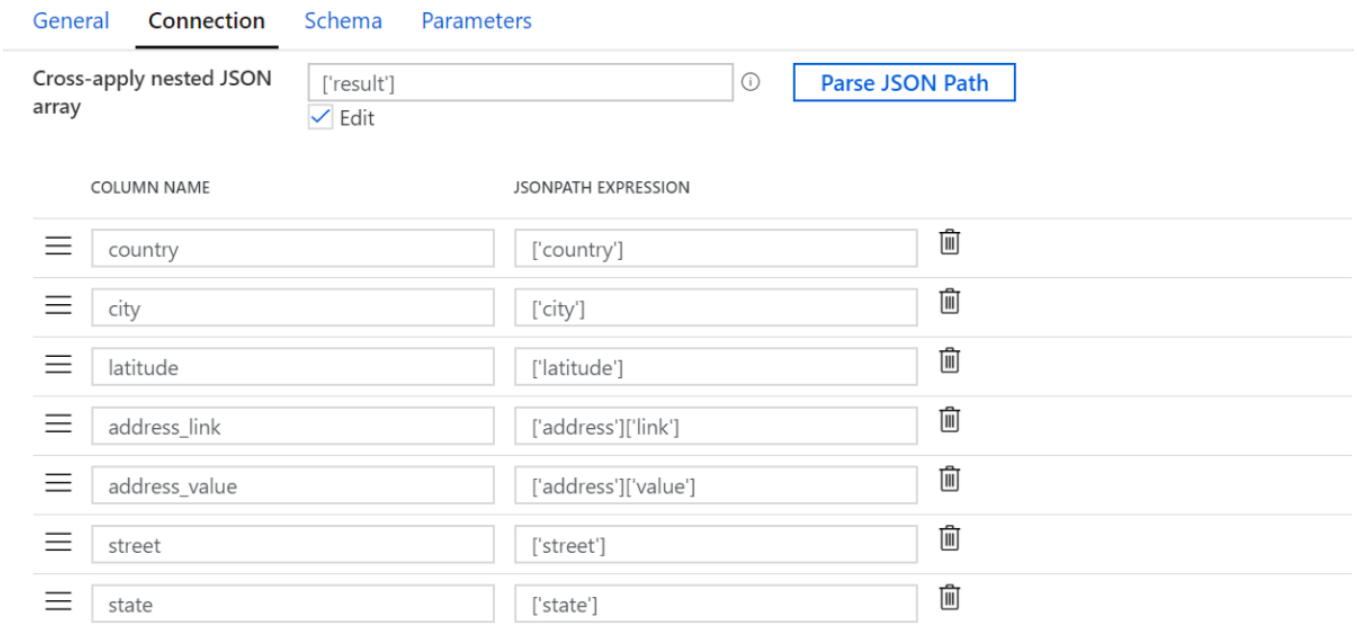
b. Manually update the JSON of the dataset using JSON editor

c. Example of edited JSON



d. Manually updating this ensures nested json is mapped to the right columns.

e. Sample connection



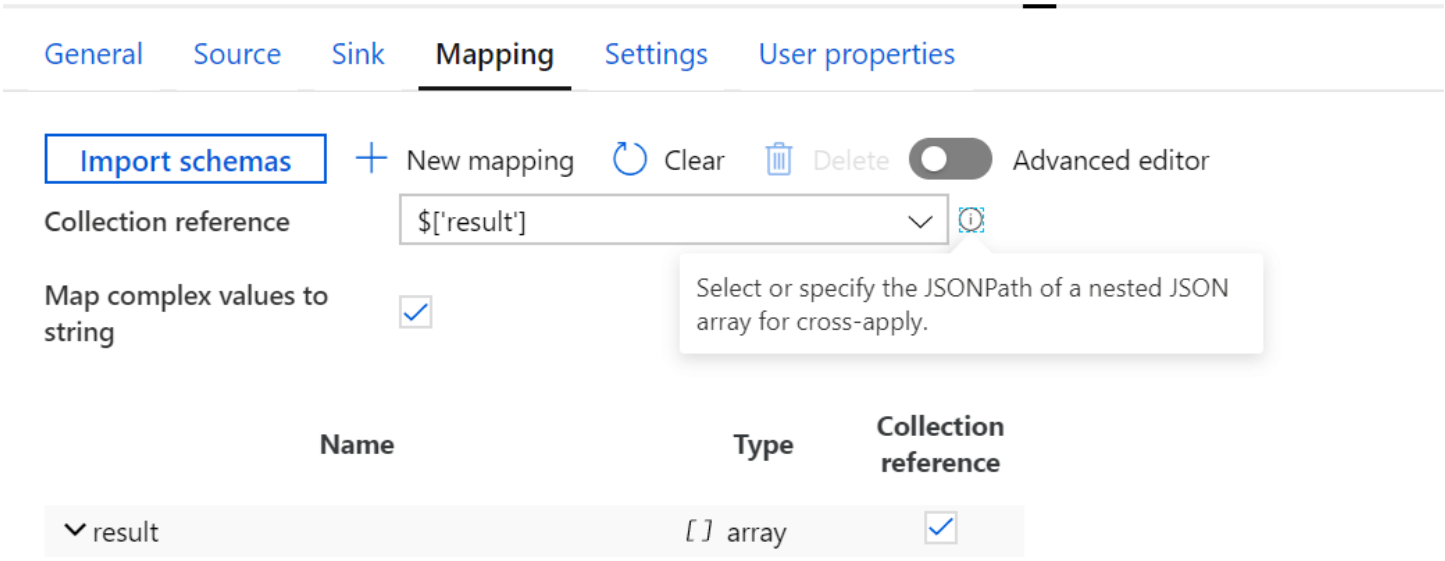
3. Create a **Azure SQL Database** dataset "**DS_Sink_Location**" that points to the destination table. The destination table must be 1-1 to the source, ensure all columns match between blob file and the destination table.
4. In a new Pipeline, create a **Copy data** task to load Blob file to Azure SQL Server

    a) Connect "DS_Source_Location" dataset to the Source tab

    b) Connect "DS_Sink_Location" dataset to the Sink tab

    c) Review Mapping tab, ensure each column is mapped between Blob file and SQL table

    d) Specify the JSONPath of the nested JSON array for cross-apply



e) Execute pipeline

The process outlined above loads the data from a JSON object using an API connection to the Azure SQL Database. Please contact us if you would like a full end-to-end solution to copy data from an API using a delta load and pagination.

Please see our Azure Practice page or contact us if you would like to know more about our Azure offering. Below are links to more of our Azure Blogs:

Should we invest more time getting data into a Data Warehouse (DW) or analysing the data to create insights for the business?

Using SSIS within Azure Data Factory V2

Azure Data Factory Templates with Data Lake

Comments

Michael Shparber

August 18, 2020 at 3:30 pm

Thank you.
How can I get the delta and pagination sample you've mentioned?

Kunal

September 16, 2020 at 7:03 pm

My scenario has requirement to pull data from 1000 API's atleast, so how can I achieve this solution as I don't want to create separate pipelines for each API calls and map each and every time, Please let me know your inputs.
Thanks.

Altis Consulting

September 22, 2020 at 11:18 am

Hi Kunal,
Our Azure framework can cater for multiple API's and different data sources using a single orchestration process, this means you can add/remove/disable APIs without having to create new pipelines. Please get in touch with our team if you would like to discuss how we can help your organisation.

Contact us

If you'd like to find out more about how we can help, simply connect with the regional office nearest to you:

NSW | ACT | VIC | NZ | UK

About Altis

We are an experienced Data and Analytics consultancy combining technical skill, commercial expertise, communication and listening. Our highly-skilled team use information management strategies and tools to help you make intelligent use of your data and make the right commercial decisions.

Social media

Be a part of our wider network and our news, thoughts and opinions across our social network.

Connect with us

If you'd like to be kept in the loop on courses, events and other related topics, simply complete your details and we'll add you to our list.