USING OAUTH2 WITH QUICKBASE PIPELINES AND PROCESSING JSON DATA

By Mike Frishman (Discord @mikefrishman)

Contents

Abstract.. 1
1) OBTAINING CLIENT CREDENTIALS 2
2) CREATING PIPELINE TRIGGER 2
3) OBTAINING ACCESS TOKEN 2
4) OBTAINING JSON DATA 3
5) PROCESSING JSON DATA 5
6) UPDATING QUICKBASE 6
7) PROCESSING NESTED/CHILD RECORDS 7
8) PIPELINE LAYOUT 10

Abstract

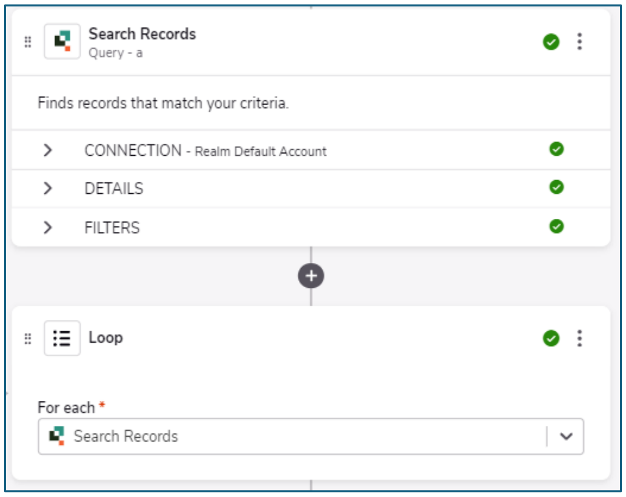
OAuth 2.0 is an authorization protocol that is designed to grant access to a remote API. OAuth 2.0 works by first obtaining static credentials, a “Client ID” and a “Client Secret”, from the remote system. Using these credentials, the system requiring access (Quickbase) requests a dynamic “Access Token” from the remote API, which is then used as authorization for the API calls. This Access Token typically expires after a certain amount of time (1 hour, 1 day, etc.) and must be refreshed regularly using the client ID and secret to call for another token. The following walk-through will show you how to setup OAuth 2.0 using the “client credentials” method.

1) OBTAINING CLIENT CREDENTIALS

You will first need to obtain a “Client ID” and a “Client Secret” from the remote website. Each website has a different way of obtaining this information, so read their API instructions for more information.

2) CREATING PIPELINE TRIGGER

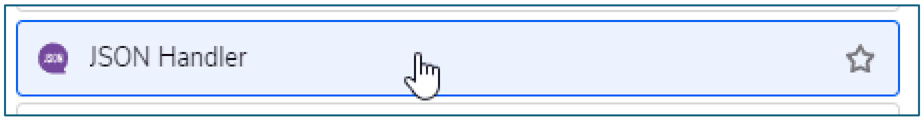
a) Create a trigger that initiates the pipeline. The trigger for the pipeline can either be an action (add record, update record) or a scheduled activity. If it’s a scheduled activity that triggers a Search Records action, then the Fetch JSON action is inside the loop created by the Search Records step.

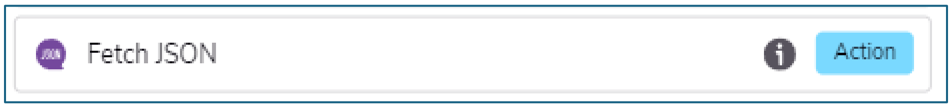

Search Records
Query - a
Finds records that match your criteria.
CONNECTION - Realm Default Account
DETAILS
FILTERS
Loop
For each *
Search Records

3) OBTAINING ACCESS TOKEN

In this step, you will use the client credentials that you received in **OBTAINING CLIENT CREDENTIALS** to obtain the limited-time access token.

a) After the trigger step, add the step **JSON Handler** and action **Fetch JSON**





b) Enter the field values below. While these are typical field values, they may not be the same values required by the remote system. Most systems have instructions available for their APIs that you can obtain online.

i) **Authentication Schema:** No Authentication
ii) **Disable Ssl Certificate Validation:** No
iii) **JSON URL:** {example} https://apicenter.eagleview.com/oauth2/v1/token

*The remote system will provide you their specific URL for obtaining the token in their API instructions

iv) **Outgoing request's method type:** POST
v) **Headers:** Content-Type:application/x-www-form-urlencoded
vi) **Request Body:** grant_type=client_credentials&client_id=xxxxxxxxxxxxxxx&client_secret=xxxxxxxxxxxxxxx

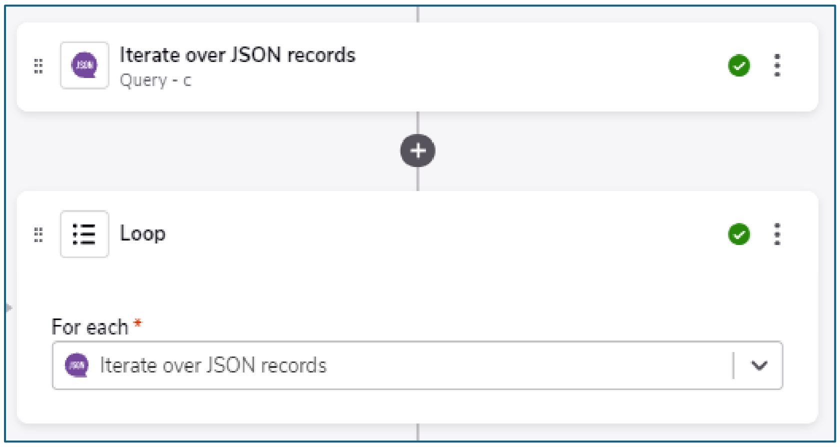
*Replace xxxxxxx with the client id and client secret that you obtained in OBTAINING CLIENT CREDENTIALS. The format above may not exactly match the requirement for your remote system, but it is common.

vii) **Username:** {blank}
viii) **Password:** {blank}
ix) **Token:** {blank}
x) **Proxy Connection Via On Premises Agent:** {blank}
xi) **Force Content Encoding:** {blank}

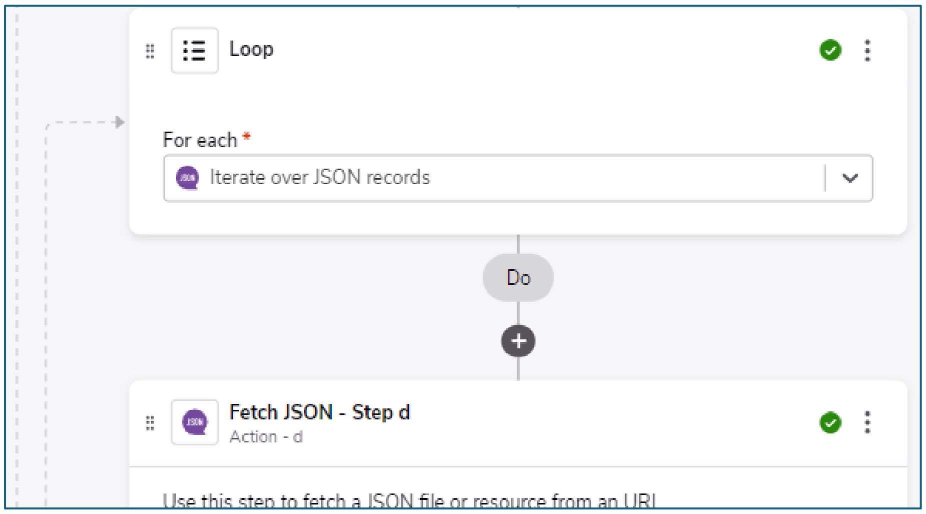
4) OBTAINING JSON DATA

In this step, you will be pulling the actual data from the remote system, using the access token for authentication that you obtained in OBTAINING ACCESS TOKEN above.

a) Add the step **JSON Handler**, then select **Iterate over JSON records.**



b) This will create a Loop. Inside the loop, add the step **JSON Handler**, then select **Fetch JSON**.


Loop

For each *

Iterate over JSON records

Do

Fetch JSON - Step d

Action - d

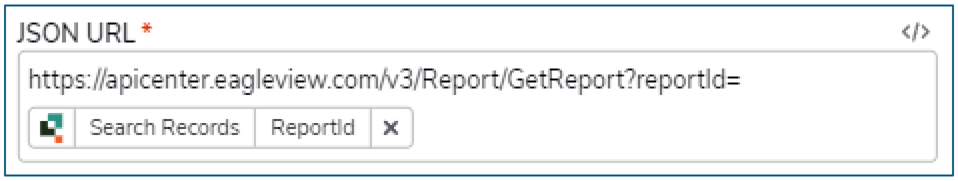
Use this step to fetch a JSON file or resource from an URL

c) Enter the field values below. While these are typical field values, they may not be the same values required by the remote system. Most systems have instructions available for their APIs that you can obtain online.

i) **Authentication Schema:** No Authentication

ii) **Disable Ssl Certificate Validation:** No

iii) **JSON URL:** {example}

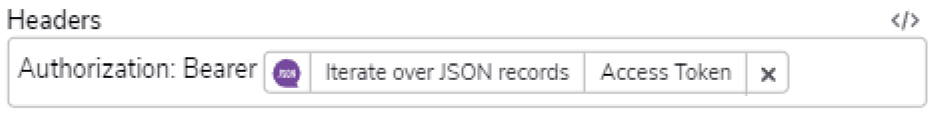

JSON URL *

https://apicenter.eagleview.com/v3/Report/GetReport?reportId= Search Records ReportId

*The remote system will provide you with a list of URLs for their API, which is dependent on what you are trying to accomplish. In the example above, I am requesting report data with the ReportId (obtained separately and available as a record in Quickbase) as the identifier.

iv) **Outgoing request's method type:** GET (or POST or PUT, depending on the API you are requesting)

v) **Headers:** Authorization: Bearer {{Access Token}}


Headers

Authorization: Bearer Iterate over JSON records Access Token

*This step uses the access token you received in **OBTAINING ACCESS TOKEN** as the authorization to obtain the actual data from the remote system.

vi) **Request Body:** {blank}

vii) **Username:** {blank}

viii) **Password:** {blank}

ix) **Token:** {blank}

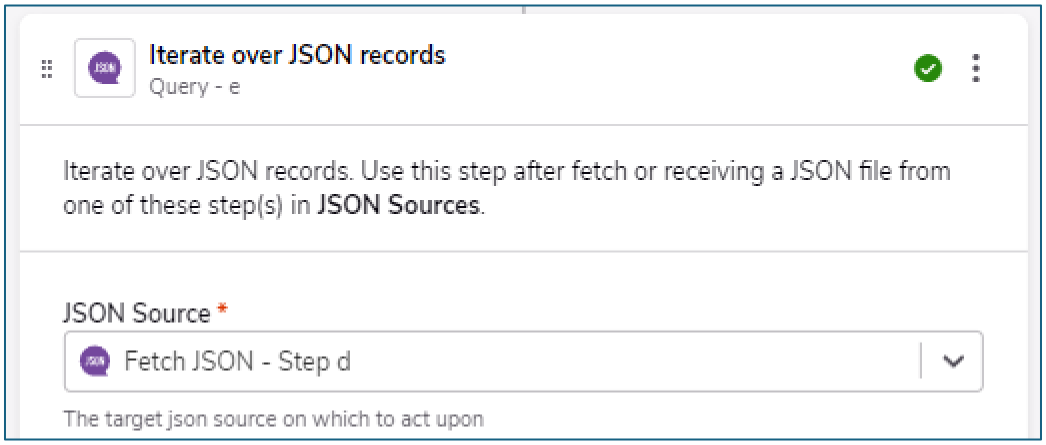
x) **Proxy Connection Via On Premises Agent:** {blank}

xi) **Force Content Encoding:** {blank}

5) PROCESSING JSON DATA

In this step you will be processing the data you received from the remote system in the previous step. This data might be in the form of an array, so you must tell the pipeline how to handle each iteration.

a) Add the step **JSON Handler**, then select **Iterate over JSON records**.



Iterate over JSON records
Query - e

Iterate over JSON records. Use this step after fetch or receiving a JSON file from one of these step(s) in **JSON Sources**.

JSON Source *
Fetch JSON - Step d
The target json source on which to act upon

*__IMPORTANT:__ Make sure your **JSON Source** is the JSON you receive in Step d, and *not* the JSON you receive in Step b.

b) Enter the field values below. While these are typical field values, they may not be the same values required by the remote system. Most systems have instructions available for their APIs that you can obtain online.
   i) **Authentication Schema:** No Authentication
   ii) **JSON Schema Sample:** {See below}

   *The "Schema" you paste here is an example of what Quickbase will receive when the API request is processed, and how it will be formatted. This tells Quickbase what data it should expect, and how to handle that data when it comes.

   You may obtain this schema sample from the remote system's API instructions, or you may use Postman to test your API. The response you receive in Postman can be pasted into this step as a sample of what Quickbase can expect to receive.

   iii) **JSON Records Path:** {blank}
   iv) **Limit:** {blank}
   v) **Filters:** {blank}

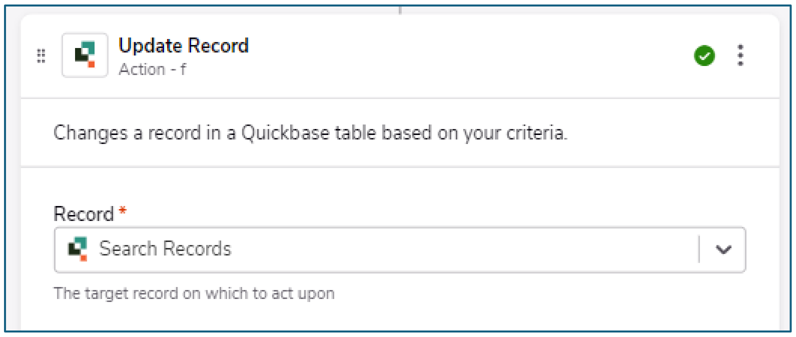
6) UPDATING QUICKBASE

a) By adding the **Iterate over JSON records** function in **PROCESSING JSON DATA**, a loop will be created in Pipelines. In Step f you must tell Quickbase what you want done with each JSON record.

b) You may Create Record, Update Record, or any other Action. In our example, we will be updating existing records.

   i) **Record:** {see below}

   *Select the step that includes the records you want to update, in this example, we are updating the record that we searched for in Step a.



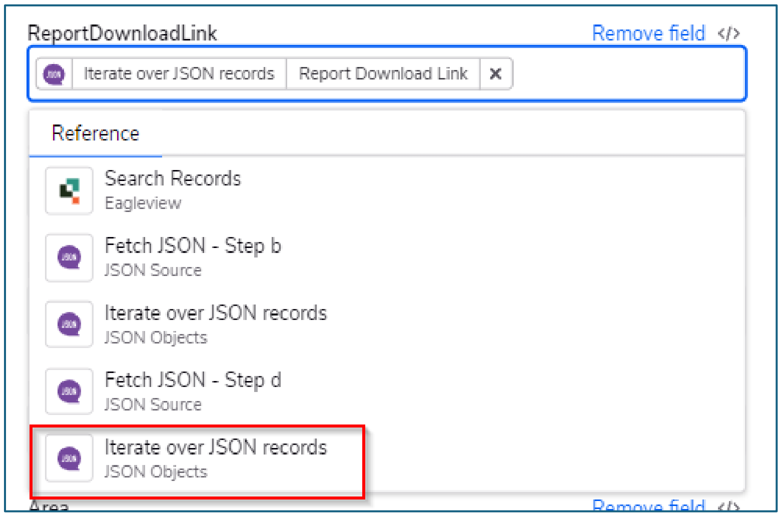
Update Record
Action - f

Changes a record in a Quickbase table based on your criteria.

Record *
Search Records
The target record on which to act upon

   ii) **Add field:** {See below}

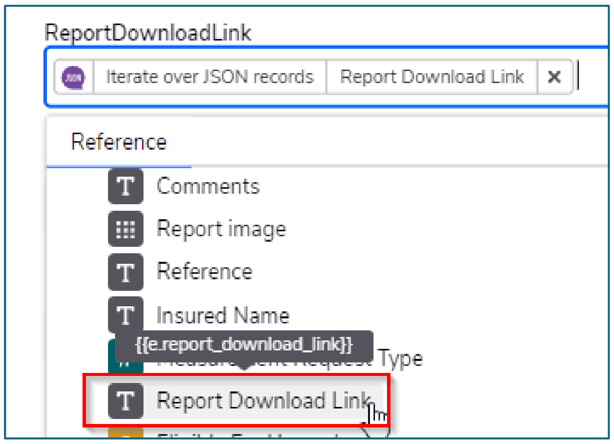
   (1) Add the Quickbase fields you want to receive data, (i.e. ReportDownloadLink). Then select the data that you want to be input into those fields. Most likely you will want to select the recent **Iterate over JSON records** step.



ReportDownloadLink
Iterate over JSON records | Report Download Link

Reference
Search Records — Eagleview
Fetch JSON - Step b — JSON Source
Iterate over JSON records — JSON Objects
Fetch JSON - Step d — JSON Source
Iterate over JSON records — JSON Objects

   (2) Then select the field(s) from the JSON record that should be input into each Quickbase field. The choices available are based on the **Schema Sample** you pasted into the Pipeline in a previous step.



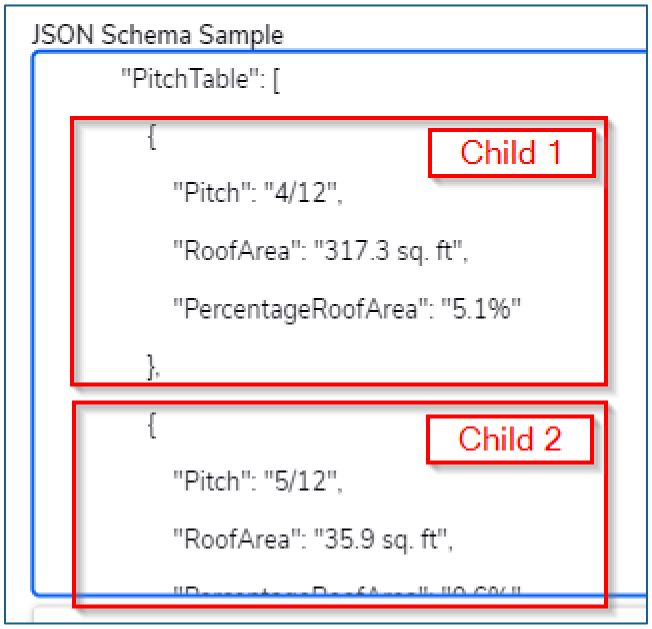
(3) Repeat with additional fields

iii) **Link (advanced):** {blank}

c) At this point, your pipeline should be functional. However, the instructions above only work for the top-level/parent data in each iteration of the JSON. If there are nested/child arrays in your JSON, you will need to add another step to process the child records.

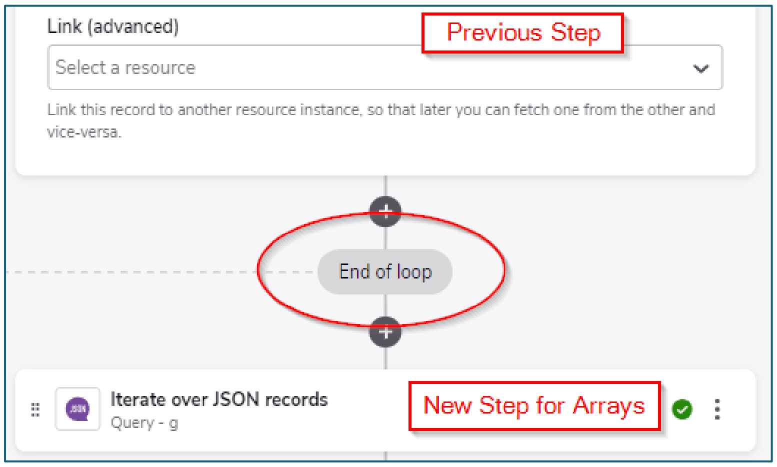
7) PROCESSING NESTED/CHILD RECORDS

If there are child arrays in your JSON, such as the example below, you will need to add another step to process the child records.



```
"PitchTable": [
    {
        "Pitch": "4/12",
        "RoofArea": "317.3 sq. ft",
        "PercentageRoofArea": "5.1%"
    },
    {
        "Pitch": "5/12",
        "RoofArea": "35.9 sq. ft",
```

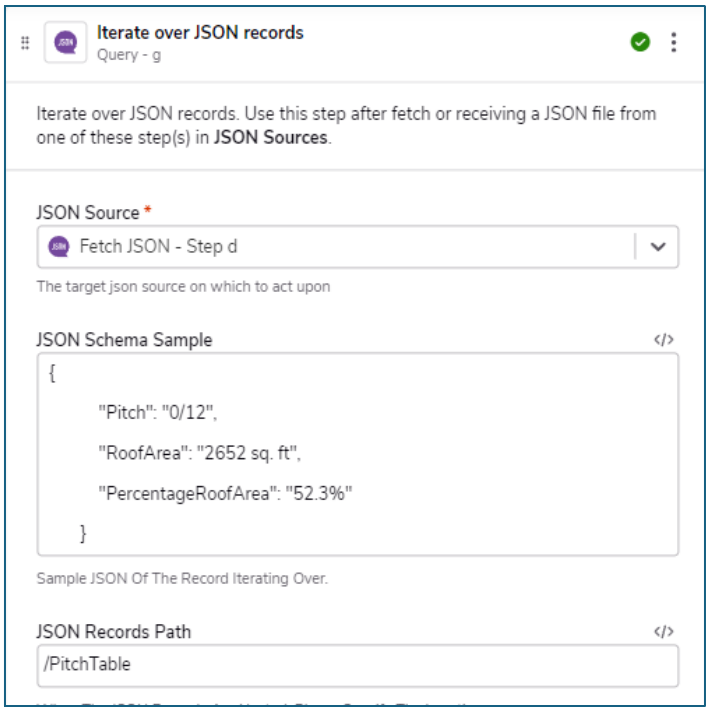
a) Add a new **Iterate over JSON records** to your Pipeline *OUTSIDE THE PREVIOUS LOOP*.


Link (advanced)

Select a resource

Link this record to another resource instance, so that later you can fetch one from the other and vice-versa.

b) Enter the field values below:
   i) **JSON Source:** {same Fetch JSON step used in PROCESSING JSON DATA}
   ii) **JSON Schema Sample:** {a single record from the child array, minus the comma}
   iii) **JSON Records Path:** {forward slash, then the name of the parent in the Schema Sample}


Iterate over JSON records
Query - g

Iterate over JSON records. Use this step after fetch or receiving a JSON file from one of these step(s) in **JSON Sources**.

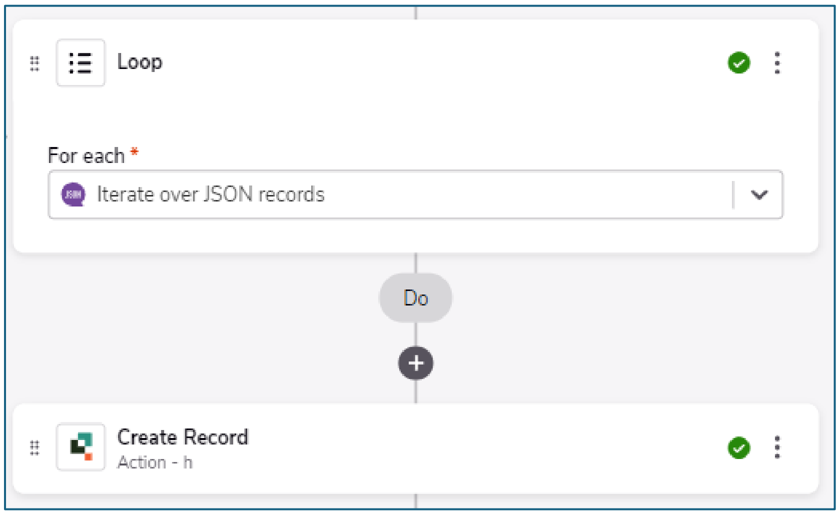
JSON Source *
Fetch JSON - Step d
The target json source on which to act upon

JSON Schema Sample
```
{
        "Pitch": "0/12",
        "RoofArea": "2652 sq. ft",
        "PercentageRoofArea": "52.3%"
    }
```
Sample JSON Of The Record Iterating Over.

JSON Records Path
/PitchTable

   iv) **Limit:** {blank}
   v) **Filter:** {blank}

c) Adding a new **Iterate over JSON records** step will also create a loop. Inside this loop, you will need to tell Quickbase what to do with each nested/child record. Typically, you would not have existing child records for each possible nested value, so you will need to create a record in a child table, linked to the parent.



d) Follow the instructions in UPDATING QUICKBASE to choose the Quickbase fields and to add the mappings from the JSON data.

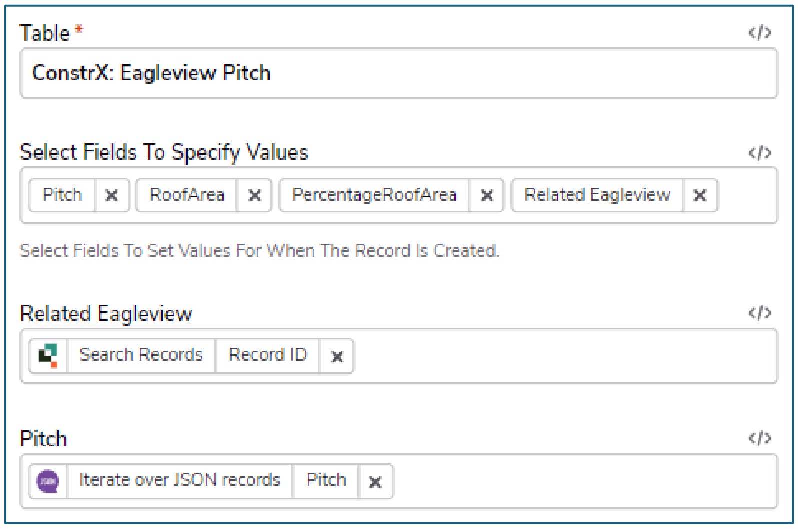


Table *
ConstrX: Eagleview Pitch

Select Fields To Specify Values
Pitch | RoofArea | PercentageRoofArea | Related Eagleview

Select Fields To Set Values For When The Record Is Created.

Related Eagleview
Search Records | Record ID

Pitch
Iterate over JSON records | Pitch

e) You should now have 3 “End of loop” records at the end of the pipeline.

8) PIPELINE LAYOUT

STEP	STEP NAME	PURPOSE	COMMENTS
a	Search Records	Trigger step	Can be Record Created, Record Update, etc.
	Loop		For each {step a}
b	Fetch JSON	Obtain access token	
c	Iterate over JSON records	Identify access token	JSON Source = {step b}
	Loop		For each {step c}
d	Fetch JSON	Obtain data	
e	Iterate over JSON records	Process parent JSON records	JSON Source = {step d}
	Loop		For each {step e}
f	Update Record	Action step	Parent data. Can be Create Record, etc.
	End Loop		
g	Iterate over JSON records	Process nested JSON records	JSON Source = {step d}
	Loop		
h	Create Record	Action step	Nested data
	End Loop		
	End Loop		
	End Loop		