Splunk Search Mockup with Fragments

This example is substantially the same as the mocked-splunk-search example with a couple of changes:

- The first pipeline looks up a `search_keyword` in a database and then uses a fragment to encapsulate and launch the Splunk Search.

- The second pipeline uses a fragment to encapsulate the interaction with the Splunk search APIs and then parses the search results and loads one or more Snowflake tables.

The pipelines and fragments can be downloaded from here

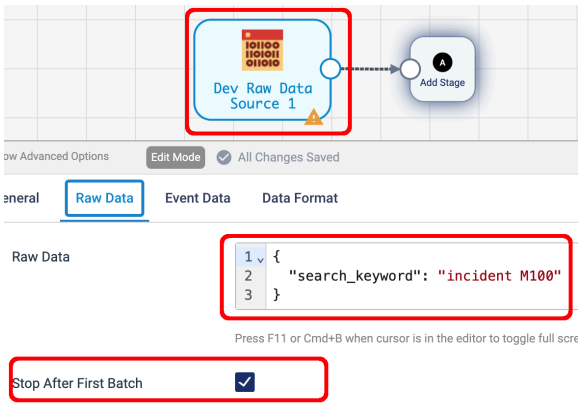
Pipeline 1

Here is the first pipeline, that queries a database and uses a fragment to start the Splunk search:

The origin is a JDBC Query Consumer that executes the query:

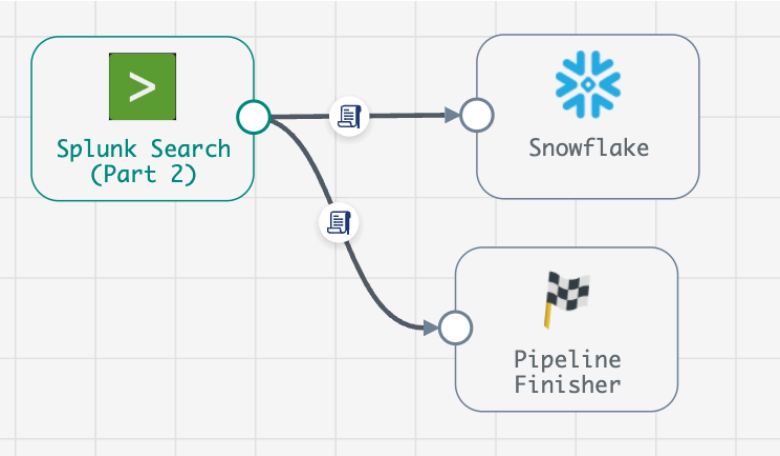
```
SELECT search_keyword
FROM streamsets.splunk_search_keywords
WHERE id = 1 ORDER BY id
```

In order to run this example one would need a database to be queried. In the meantime, you could substitute a Dev Raw Data Source to mock getting the same value, like this:

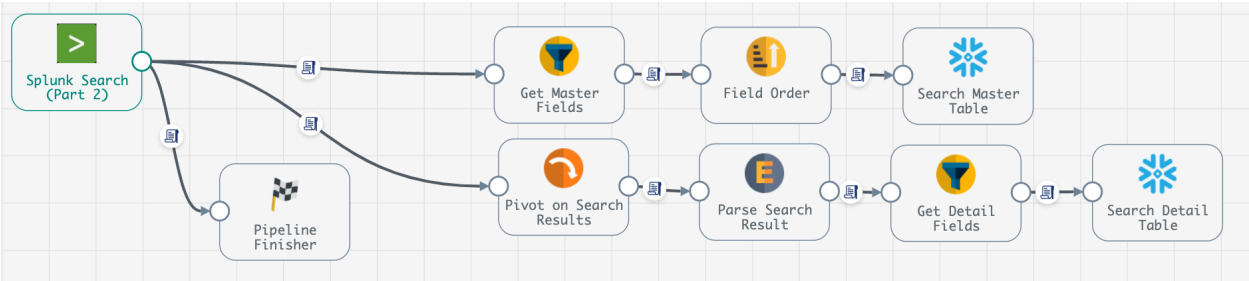


Pipeline 2

Here is a simple version of the second pipeline, that uses a fragment to complete the Splunk Search and inserts the unparsed search results into a single Snowflake table:



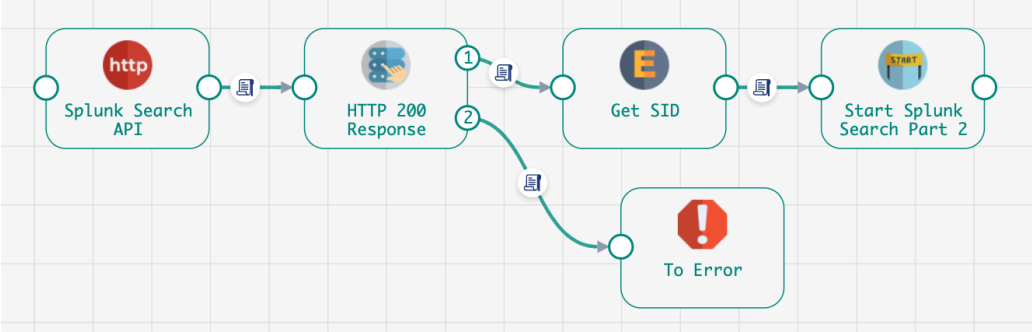
Here is an alternate, more complex implementation of the second pipeline that parses the response into master and detail Snowflake tables, to make clear that the reusable logic is in the fragment, and consumers of the Splunk search results might choose to do different things with the data.



Both versions of the second pipeline use the same fragment.

Fragment 1

The Splunk Search (Part 1) stage in the first pipeline is a fragment that looks like this:

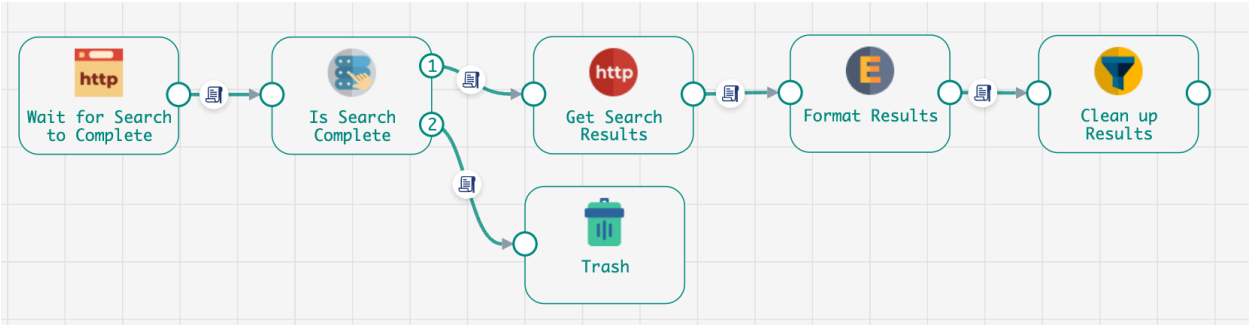


The fragment's Splunk Search API stage is an HTTP Client Processor that calls the `/splunk_submit_search` API.

The fragment's `Start Splunk Search Part 2` stage is a Start Jobs Processor that launches the second pipeline as a Job Template Instance.

Fragment 2

The Splunk Search (Part 2) stage in the second pipeline is a fragment that looks like this:



Create a Job Template for Pipeline 2

Create a Job Template for Pipeline 2 and grab its Job Template ID.

Note that the Job Template will have these Parameters:

You only need to set the `SPLUNK_BASE_URL` as the other two parameters will be set by pipeline 1 when the template is called.

Create a Job for Pipeline 1

Create a Job Pipeline 1. Note that it has these parameters:

Set the `SPLUNK_BASE_URL`, and set the `SPLUNK_PART_2_JOB_ID` to the value of the Job Template for Pipeline 2

Run the Example

Run the example by starting the Job Instance `DB Query to Splunk Search (Part 1),`

That Job will run just long enough to query the database, then launch part 2, and then quit.

You should then see a Job Instance named `Query to Splunk Search (Part 2)-incident M100` run for about 30 seconds while it polls for the search to finish, and then writes the results to Snowflake., before it quits as well.