Daily Slack updates integrating Flight and Weather APIs with data sources

Kausik Chattapadhyay

Bellevue University

DSC650: Big Data Final Project

Prof. Nasheb Ismaily

02/26/2024

Abstract

In modern application development, integrating data from diverse sources in various formats is a common challenge, especially in the realm of machine learning applications. Apache NiFi presents a powerful solution for seamlessly orchestrating data flow between different systems. This project demonstrates how Apache NiFi can be leveraged to collect data from two distinct APIs - aviationstack.com for real-time flight data and visualcrossing.com for weather data - and subsequently write this data to multiple databases while also sending daily notifications to Slack. The project provides a step-by-step research guide on setting up data ingestion, transformation, and dissemination pipelines using Apache NiFi processors.

Data Ingestion:

- Utilizing the Invoke HTTP processor to retrieve data from aviationstack.com and visualcrossing.com APIs.
- Splitting the JSON data into individual records using the SplitJSON processor.

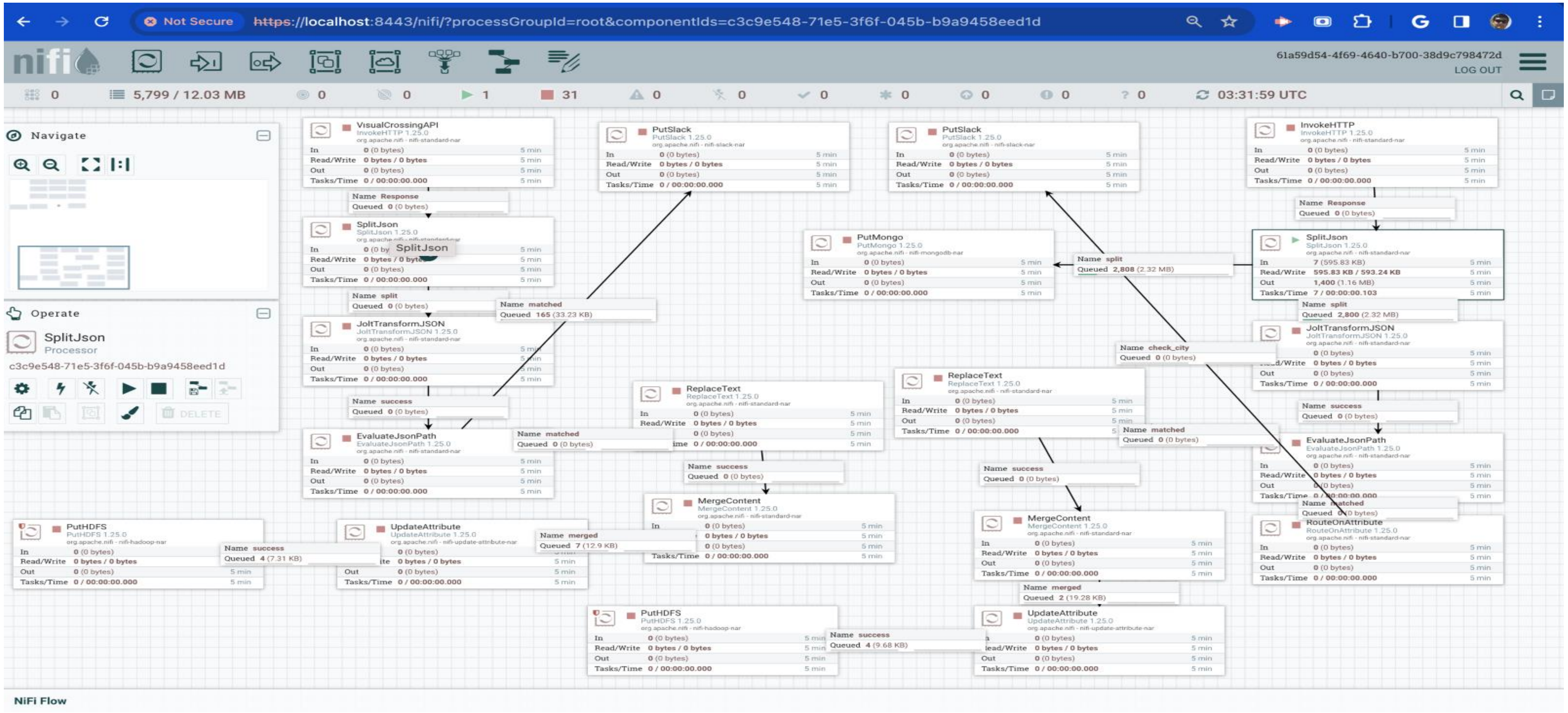
Database Integration:

- MongoDB Integration:
    - Using PutMongo processor to write data to MongoDB.
- HADOOP HDFS Integration:
    - Converting JSON data to AVRO format.
    - Converting AVRO records to CSV format.

Slack Notifications:

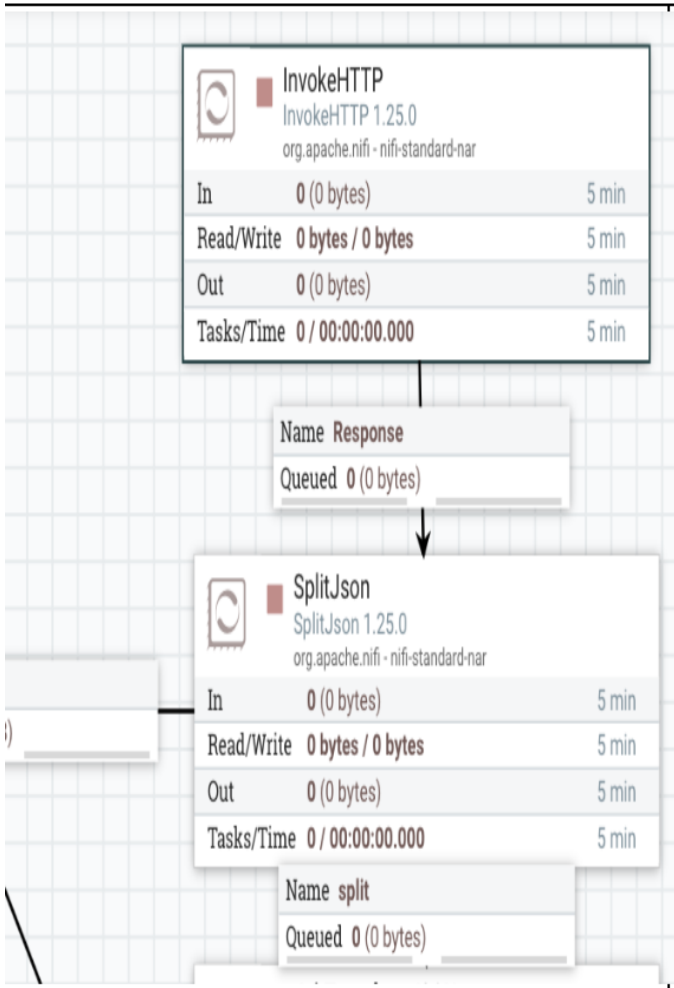
- Selecting specific attributes from the data using EvaluateJsonPath processor.
- Routing data based on attributes using RouteOnAttribute processor.
- Sending notifications to Slack using PutSlack processor.

The first API I will use aviationstack.com to access real-time flight data. The second API I will use is visualcrossing.com to access the weather data of any city. Please find below my complete NIFI Flow.



I will select a city from the data we receive from the first API and fetch the flight information. I will also bring the weather information of the selected city with the data we receive from the other API.

First, let us call the processor from which I will pull data from the API. When I run the Processor, data is returned in such a JSON format.



InvokeHTTP
InvokeHTTP 1.25.0
org.apache.nifi - nifi-standard-nar

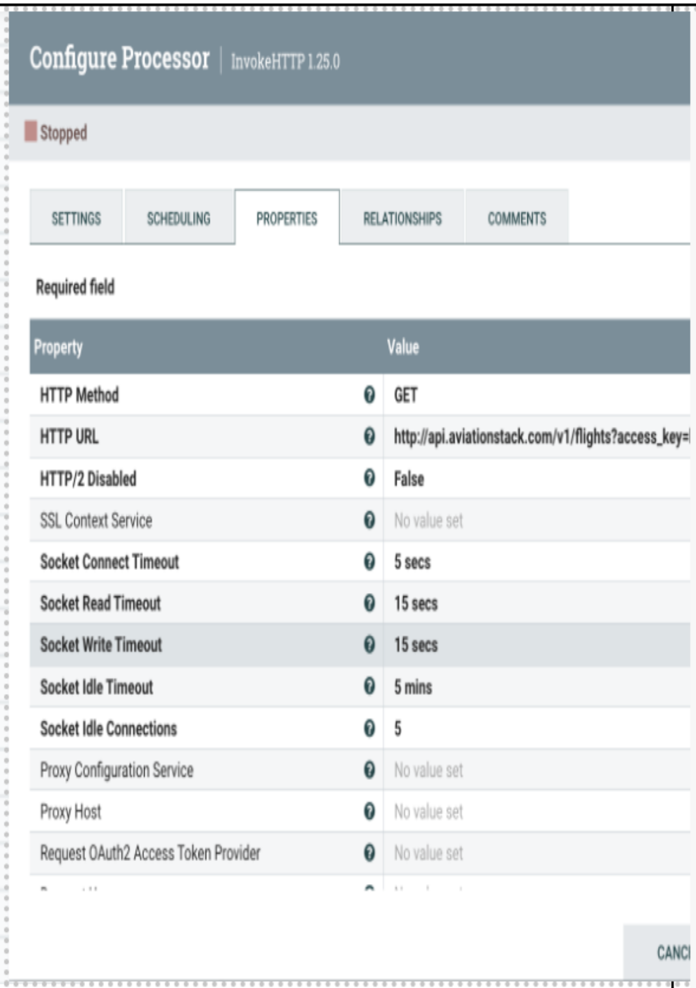
In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name Response
Queued 0 (0 bytes)

SplitJson
SplitJson 1.25.0
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name split
Queued 0 (0 bytes)

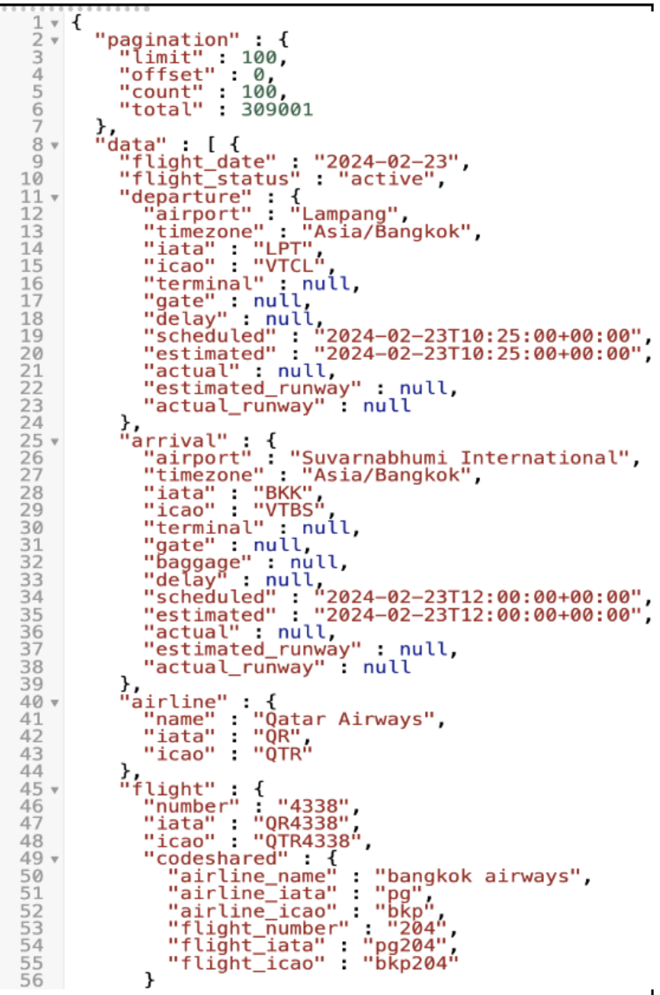


Configure Processor | InvokeHTTP 1.25.0

Stopped

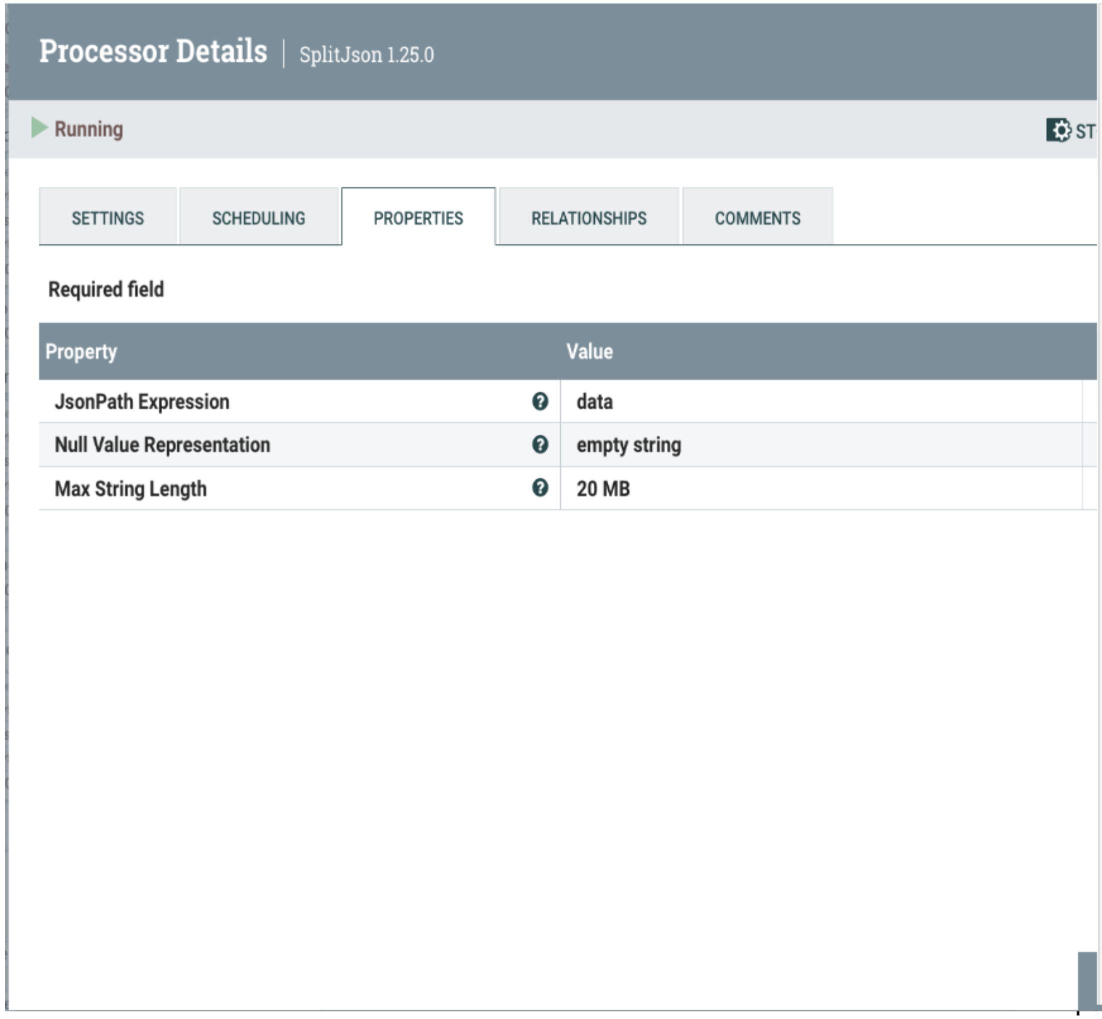
Required field

Property	Value
HTTP Method	GET
HTTP URL	http://api.aviationstack.com/v1/flights?access_key=
HTTP/2 Disabled	False
SSL Context Service	No value set
Socket Connect Timeout	5 secs
Socket Read Timeout	15 secs
Socket Write Timeout	15 secs
Socket Idle Timeout	5 mins
Socket Idle Connections	5
Proxy Configuration Service	No value set
Proxy Host	No value set
Request OAuth2 Access Token Provider	No value set

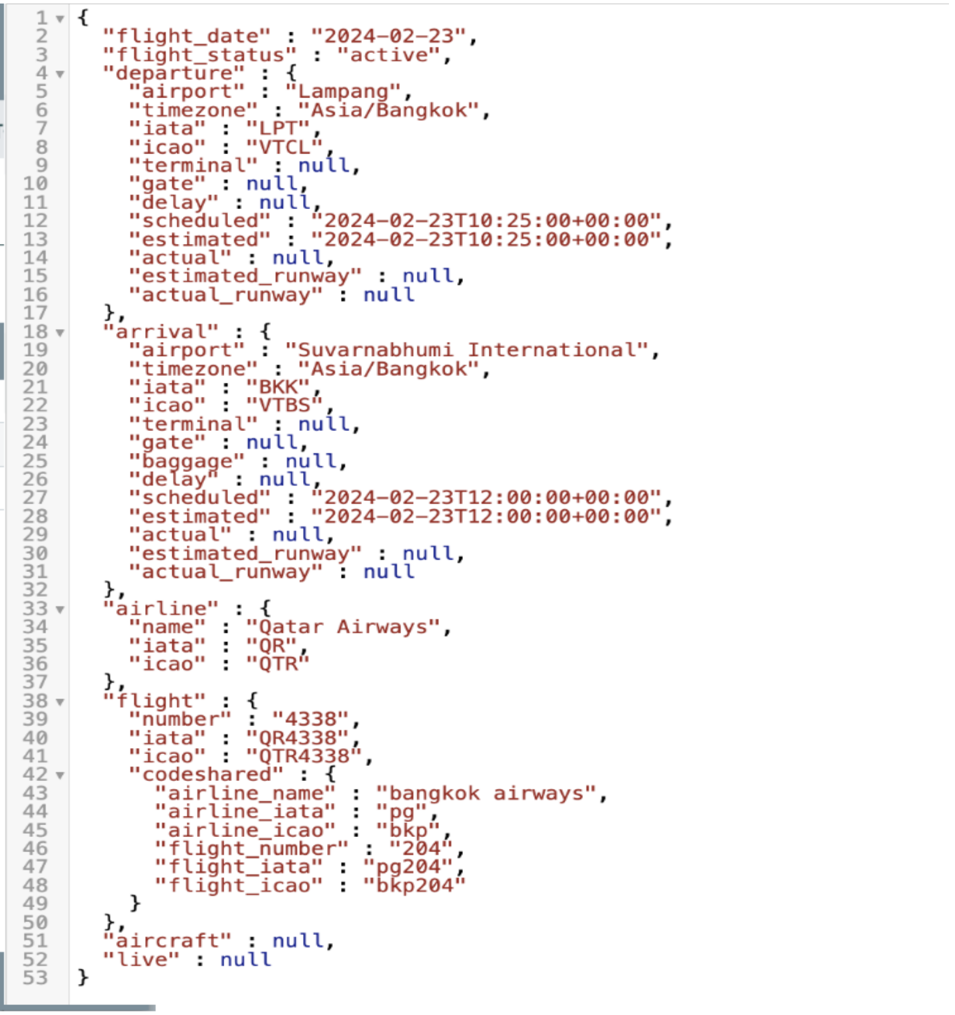


```
{
  "pagination" : {
    "limit" : 100,
    "offset" : 0,
    "count" : 100,
    "total" : 309001
  },
  "data" : [ {
    "flight_date" : "2024-02-23",
    "flight_status" : "active",
    "departure" : {
      "airport" : "Lampang",
      "timezone" : "Asia/Bangkok",
      "iata" : "LPT",
      "icao" : "VTCL",
      "terminal" : null,
      "gate" : null,
      "delay" : null,
      "scheduled" : "2024-02-23T10:25:00+00:00",
      "estimated" : "2024-02-23T10:25:00+00:00",
      "actual" : null,
      "estimated_runway" : null,
      "actual_runway" : null
    },
    "arrival" : {
      "airport" : "Suvarnabhumi International",
      "timezone" : "Asia/Bangkok",
      "iata" : "BKK",
      "icao" : "VTBS",
      "terminal" : null,
      "gate" : null,
      "baggage" : null,
      "delay" : null,
      "scheduled" : "2024-02-23T12:00:00+00:00",
      "estimated" : "2024-02-23T12:00:00+00:00",
      "actual" : null,
      "estimated_runway" : null,
      "actual_runway" : null
    },
    "airline" : {
      "name" : "Qatar Airways",
      "iata" : "QR",
      "icao" : "QTR"
    },
    "flight" : {
      "number" : "4338",
      "iata" : "QR4338",
      "icao" : "QTR4338",
      "codeshared" : {
        "airline_name" : "bangkok airways",
        "airline_iata" : "pg",
        "airline_icao" : "bkp",
        "flight_number" : "204",
        "flight_iata" : "pg204",
        "flight_icao" : "bkp204"
      }
```

Then I split the data into “days” using the SplitJSON processor. In this way, I have access to each record.

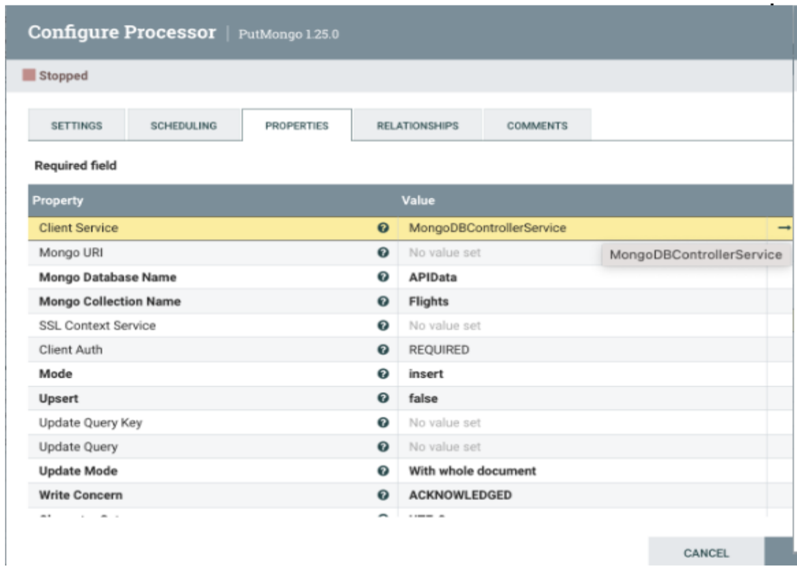


Property	Value
JsonPath Expression	data
Null Value Representation	empty string
Max String Length	20 MB



```
{
  "flight_date" : "2024-02-23",
  "flight_status" : "active",
  "departure" : {
    "airport" : "Lampang",
    "timezone" : "Asia/Bangkok",
    "iata" : "LPT",
    "icao" : "VTCL",
    "terminal" : null,
    "gate" : null,
    "delay" : null,
    "scheduled" : "2024-02-23T10:25:00+00:00",
    "estimated" : "2024-02-23T10:25:00+00:00",
    "actual" : null,
    "estimated_runway" : null,
    "actual_runway" : null
  },
  "arrival" : {
    "airport" : "Suvarnabhumi International",
    "timezone" : "Asia/Bangkok",
    "iata" : "BKK",
    "icao" : "VTBS",
    "terminal" : null,
    "gate" : null,
    "baggage" : null,
    "delay" : null,
    "scheduled" : "2024-02-23T12:00:00+00:00",
    "estimated" : "2024-02-23T12:00:00+00:00",
    "actual" : null,
    "estimated_runway" : null,
    "actual_runway" : null
  },
  "airline" : {
    "name" : "Qatar Airways",
    "iata" : "QR",
    "icao" : "QTR"
  },
  "flight" : {
    "number" : "4338",
    "iata" : "QR4338",
    "icao" : "QTR4338",
    "codeshared" : {
      "airline_name" : "bangkok airways",
      "airline_iata" : "pg",
      "airline_icao" : "bkp",
      "flight_number" : "204",
      "flight_iata" : "pg204",
      "flight_icao" : "bkp204"
    }
  },
  "aircraft" : null,
  "live" : null
}
```

I will write the data from the SplitJSON processor to MongoDB using the PutMONGO processor also in HDFS.



Configure Processor | PutMongo 1.25.0

Stopped

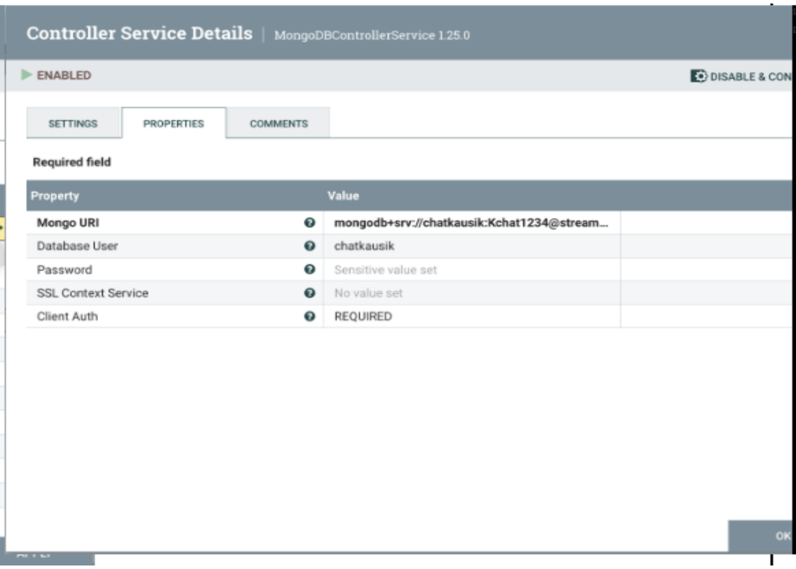
SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

Required field

Property	Value
Client Service	MongoDBControllerService
Mongo URI	No value set
Mongo Database Name	APIData
Mongo Collection Name	Flights
SSL Context Service	No value set
Client Auth	REQUIRED
Mode	insert
Upsert	false
Update Query Key	No value set
Update Query	No value set
Update Mode	With whole document
Write Concern	ACKNOWLEDGED

MongoDBControllerService

CANCEL



Controller Service Details | MongoDBControllerService 1.25.0

ENABLED

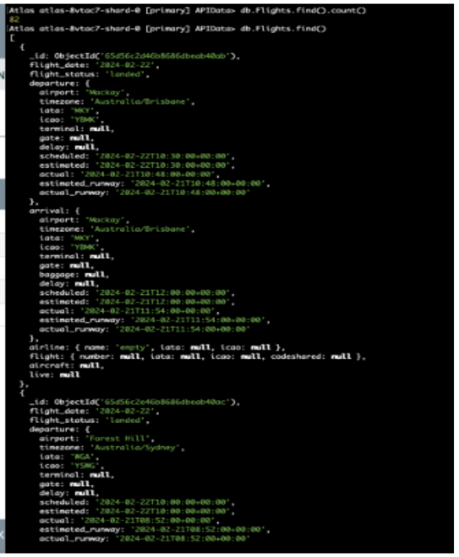
DISABLE & CON

SETTINGS | PROPERTIES | COMMENTS

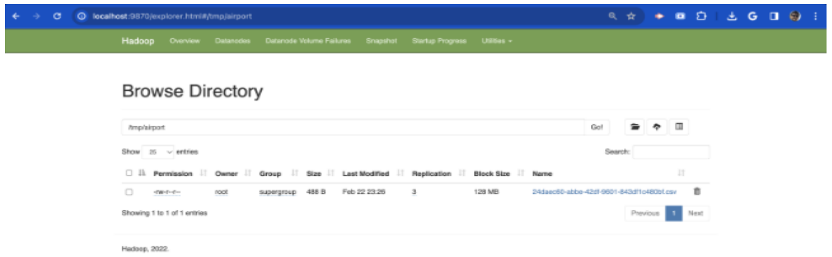
Required field

Property	Value
Mongo URI	mongodb+srv://chatkausik:Kchat1234@stream...
Database User	chatkausik
Password	Sensitive value set
SSL Context Service	No value set
Client Auth	REQUIRED

OK



```
Atlas atlas-8vtoc7-shard-0 [primary] APIData> db.Flights.find().count()
82
Atlas atlas-8vtoc7-shard-0 [primary] APIData> db.Flights.find()
[
  {
    _id: ObjectId("65d56c2d46b8686dbeab40ab"),
    flight_date: '2024-02-22',
    flight_status: 'landed',
    departure: {
      airport: "Mackay",
      timezone: 'Australia/Brisbane',
      iata: "MKY",
      icao: "YBMK",
      terminal: null,
      gate: null,
      delay: null,
      scheduled: '2024-02-22T10:30:00+00:00',
      estimated: '2024-02-22T10:30:00+00:00',
      actual: '2024-02-21T10:48:00+00:00',
      estimated_runway: '2024-02-21T10:48:00+00:00',
      actual_runway: '2024-02-21T10:48:00+00:00'
    },
    arrival: {
      airport: "Mackay",
      timezone: 'Australia/Brisbane',
      iata: "MKY",
      icao: "YBMK",
      terminal: null,
      gate: null,
      baggage: null,
      delay: null,
      scheduled: '2024-02-21T12:00:00+00:00',
      estimated: '2024-02-21T12:00:00+00:00',
      actual: '2024-02-21T11:54:00+00:00',
      estimated_runway: '2024-02-21T11:54:00+00:00',
      actual_runway: '2024-02-21T11:54:00+00:00'
    },
    airline: { name: "empty", iata: null, icao: null },
    flight: { number: null, iata: null, icao: null, codeshared: null },
    aircraft: null,
    live: null
  },
  {
    _id: ObjectId("65d56c2e46b8686dbeab40ac"),
    flight_date: '2024-02-22',
    flight_status: 'landed',
    departure: {
      airport: "Forest Hill",
      timezone: 'Australia/Sydney',
      iata: "WGA",
      icao: "YSWG",
      terminal: null,
      gate: null,
      delay: null,
      scheduled: '2024-02-22T10:00:00+00:00',
      estimated: '2024-02-22T10:00:00+00:00',
      actual: '2024-02-21T08:52:00+00:00',
      estimated_runway: '2024-02-21T08:52:00+00:00',
      actual_runway: '2024-02-21T08:52:00+00:00'
```



localhost:9870/explorer.html#/tmp/airport

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Browse Directory

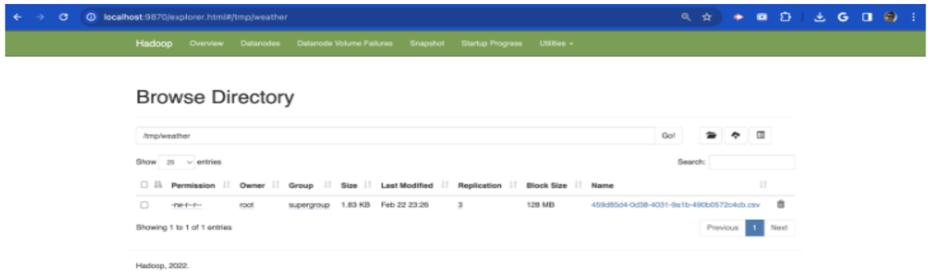
/tmp/airport

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	488 B	Feb 22 23:26	3	128 MB	24daec60-abbe-42df-9601-843df1c480bf.csv

Showing 1 to 1 of 1 entries

Hadoop, 2022.



localhost:9870/explorer.html#/tmp/weather

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Browse Directory

/tmp/weather

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	1.83 KB	Feb 22 23:26	3	128 MB	459d85d4-0d38-4031-9a1b-490b0572c4cb.csv

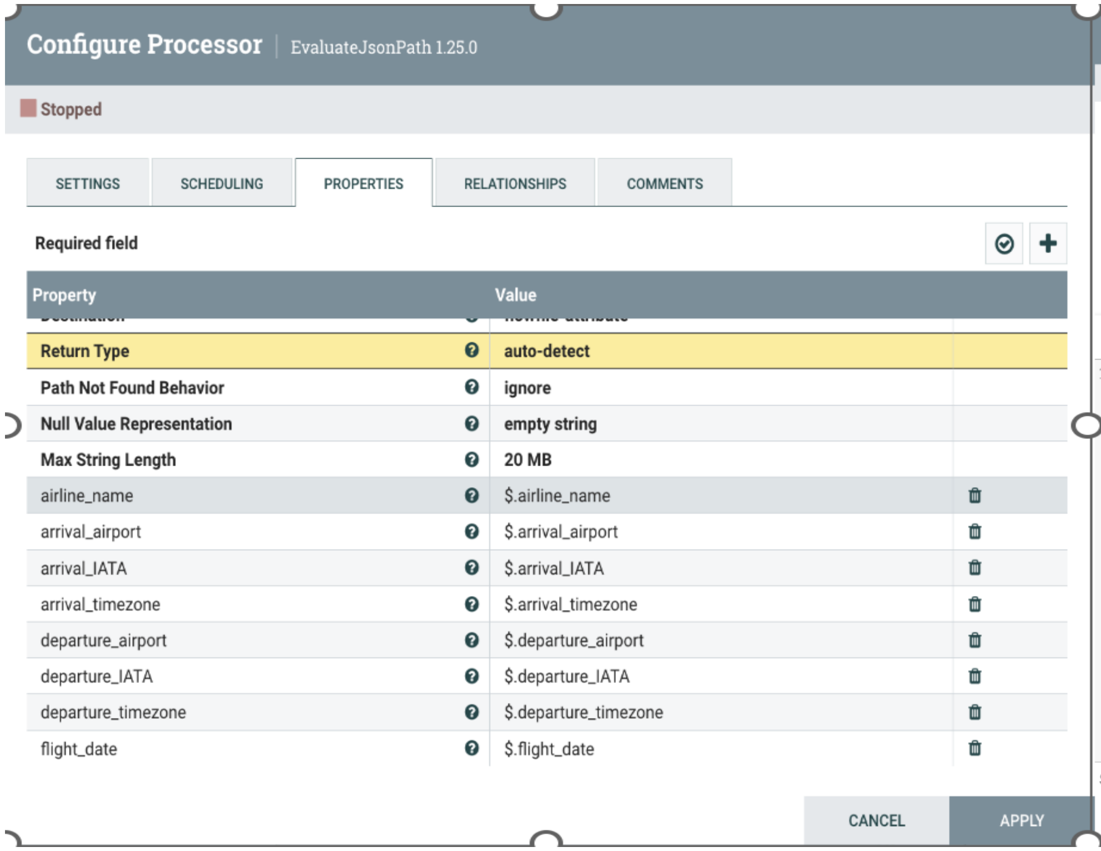
Showing 1 to 1 of 1 entries

Hadoop, 2022.

Now, let us get notifications to the Slack app using the incoming data. As I mentioned before, I would choose a city and bring its flight information. For this I need to make the data attribute.

Slack Notifications

I moved on to adding Slack notifications.



Configure Processor | EvaluateJsonPath 1.25.0

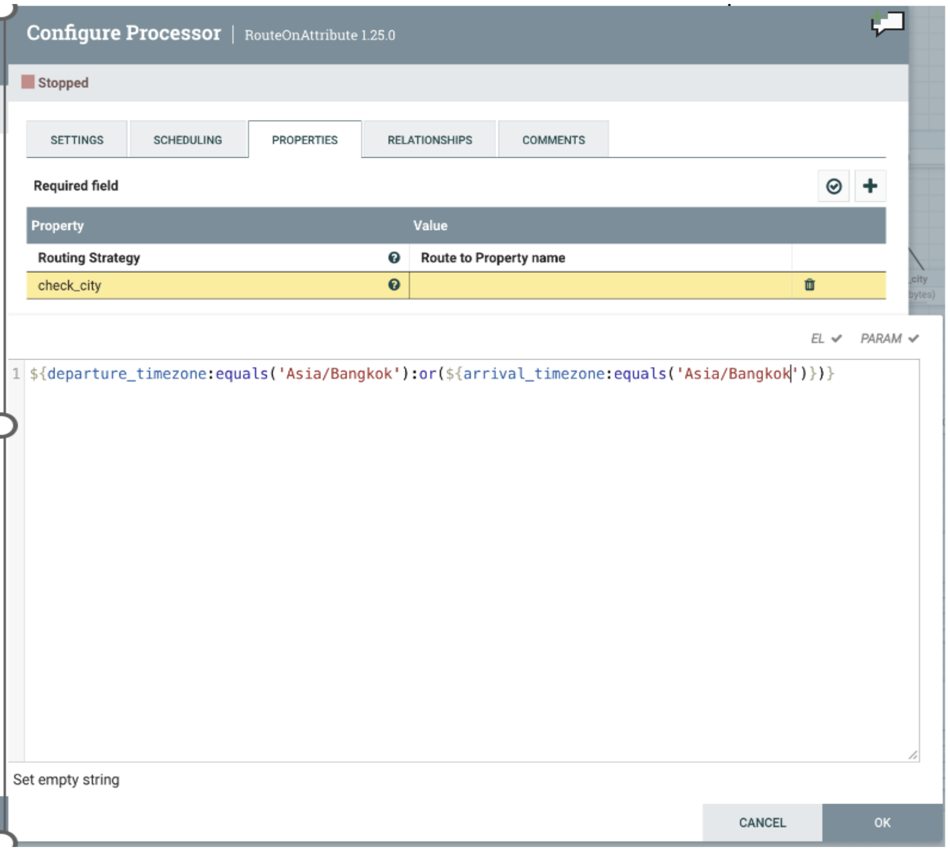
Stopped

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

Required field

Property	Value
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
Max String Length	20 MB
airline_name	$.airline_name
arrival_airport	$.arrival_airport
arrival_IATA	$.arrival_IATA
arrival_timezone	$.arrival_timezone
departure_airport	$.departure_airport
departure_IATA	$.departure_IATA
departure_timezone	$.departure_timezone
flight_date	$.flight_date

CANCEL APPLY



Configure Processor | RouteOnAttribute 1.25.0

Stopped

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

Required field

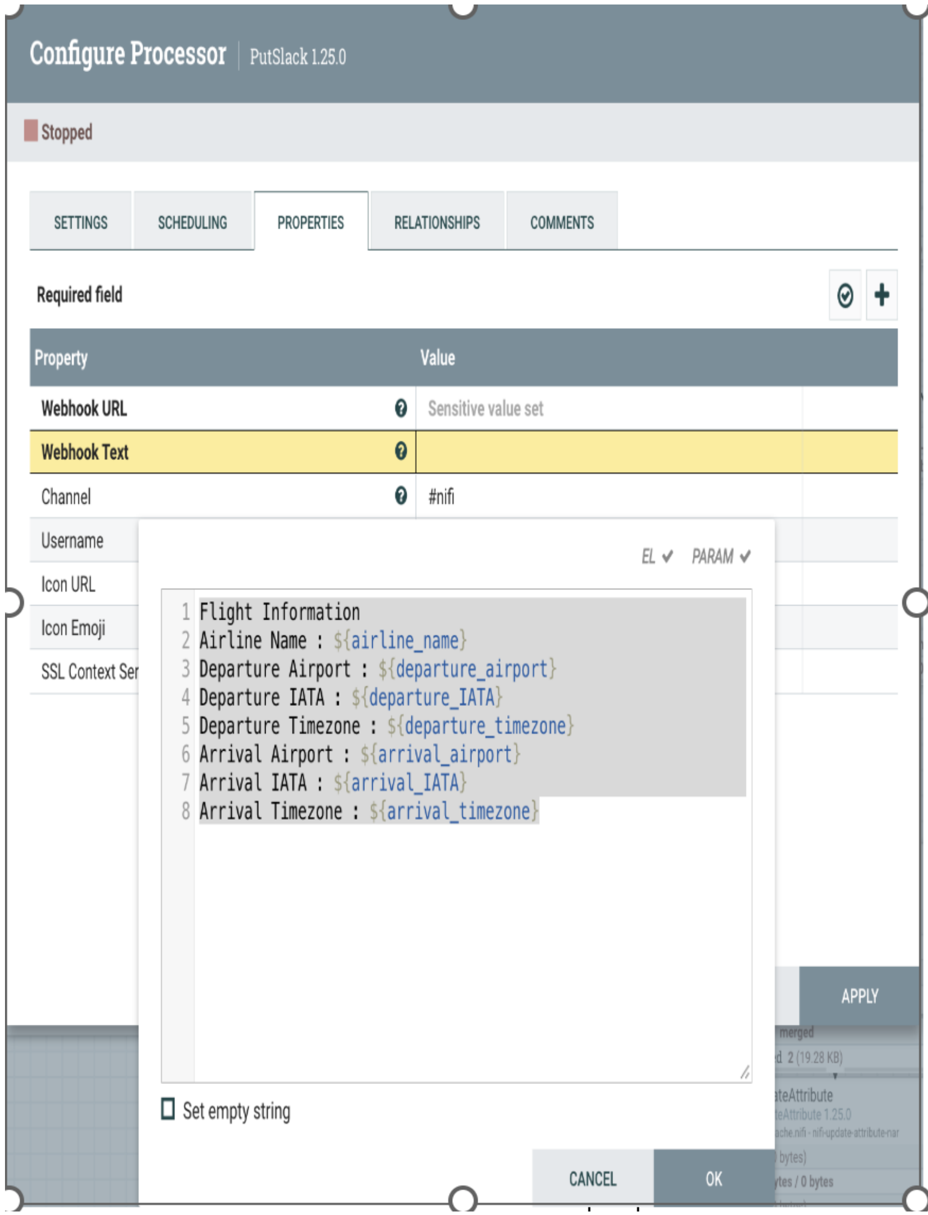
Property	Value
Routing Strategy	Route to Property name
check_city	

EL ✓ PARAM ✓

```
${departure_timezone:equals('Asia/Bangkok'):or(${arrival_timezone:equals('Asia/Bangkok')})}
```

Set empty string

CANCEL OK



Configure Processor | PutSlack 1.25.0

Stopped

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

Required field

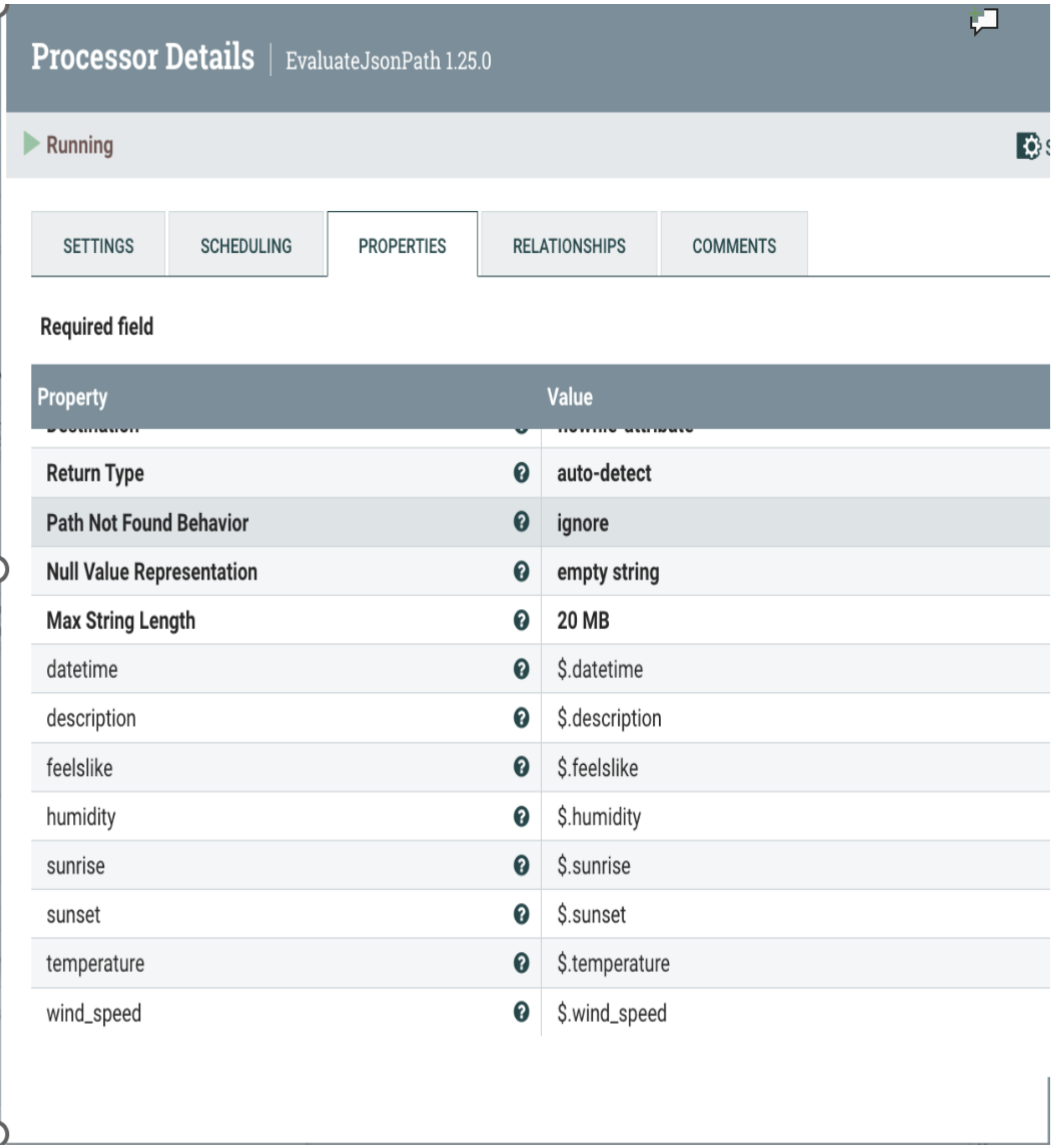
Property	Value
Webhook URL	Sensitive value set
Webhook Text	
Channel	#nifi
Username	
Icon URL	
Icon Emoji	
SSL Context Ser	

EL ✓ PARAM ✓

```
Flight Information
Airline Name : ${airline_name}
Departure Airport : ${departure_airport}
Departure IATA : ${departure_IATA}
Departure Timezone : ${departure_timezone}
Arrival Airport : ${arrival_airport}
Arrival IATA : ${arrival_IATA}
Arrival Timezone : ${arrival_timezone}
```

☐ Set empty string

CANCEL OK APPLY



Processor Details | EvaluateJsonPath 1.25.0

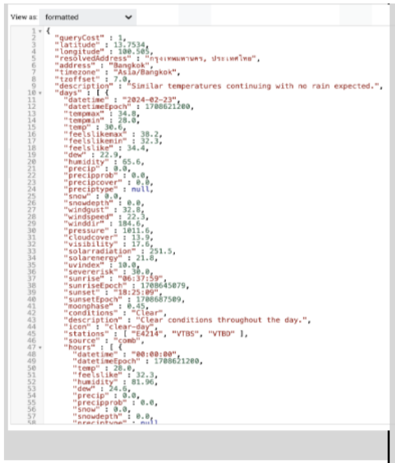
Running

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

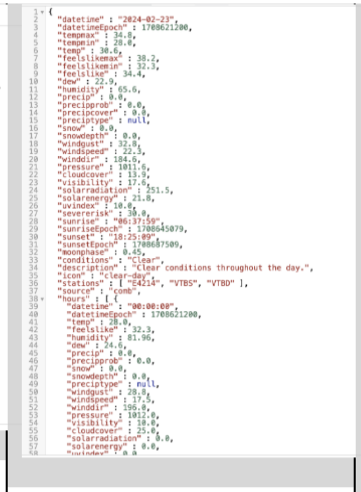
Required field

Property	Value
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
Max String Length	20 MB
datetime	$.datetime
description	$.description
feelslike	$.feelslike
humidity	$.humidity
sunrise	$.sunrise
sunset	$.sunset
temperature	$.temperature
wind_speed	$.wind_speed

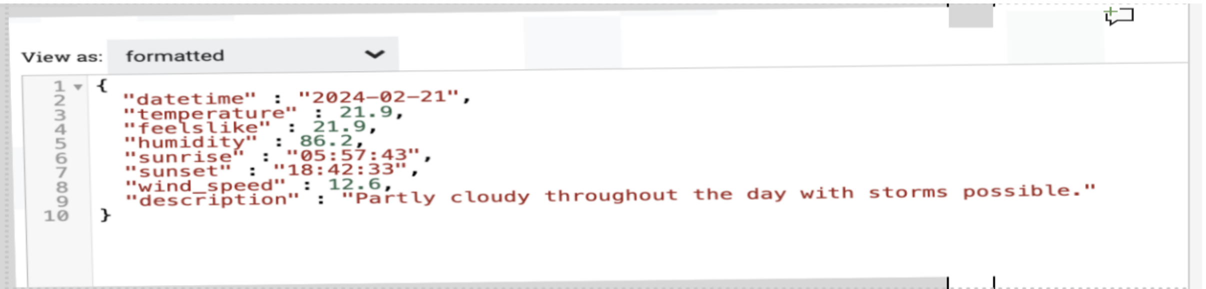
Now I am doing the operations in the first API to the data coming from the other weather API.


View as: formatted

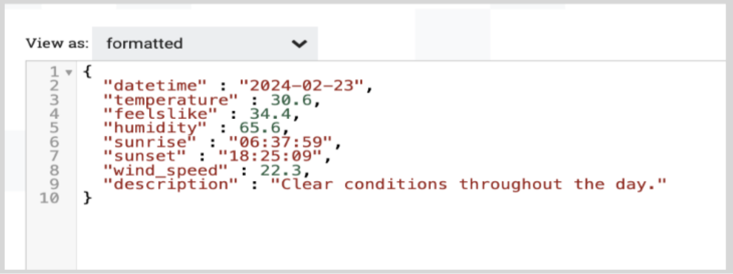
```
{
  "queryCost" : 1,
  "latitude" : 13.7534,
  "longitude" : 100.505,
  "resolvedAddress" : "กรุงเทพมหานคร, ประเทศไทย",
  "address" : "Bangkok",
  "timezone" : "Asia/Bangkok",
  "tzoffset" : 7.0,
  "description" : "Similar temperatures continuing with no rain expected.",
  "days" : [ {
    "datetime" : "2024-02-23",
    "datetimeEpoch" : 1708621200,
    "tempmax" : 34.8,
    "tempmin" : 28.0,
    "temp" : 30.6,
    "feelslikemax" : 38.2,
    "feelslikemin" : 32.3,
    "feelslike" : 34.4,
    "dew" : 22.9,
    "humidity" : 65.6,
    "precip" : 0.0,
    "precipprob" : 0.0,
    "precipcover" : 0.0,
    "preciptype" : null,
    "snow" : 0.0,
    "snowdepth" : 0.0,
    "windgust" : 32.8,
    "windspeed" : 22.3,
    "winddir" : 184.6,
    "pressure" : 1011.6,
    "cloudcover" : 13.9,
    "visibility" : 17.6,
    "solarradiation" : 251.5,
    "solarenergy" : 21.8,
    "uvindex" : 10.0,
    "severerisk" : 30.0,
    "sunrise" : "06:37:59",
    "sunriseEpoch" : 1708645079,
    "sunset" : "18:25:09",
    "sunsetEpoch" : 1708687509,
    "moonphase" : 0.45,
    "conditions" : "Clear",
    "description" : "Clear conditions throughout the day.",
    "icon" : "clear-day",
    "stations" : [ "E4214", "VTBS", "VTBD" ],
    "source" : "comb",
    "hours" : [ {
      "datetime" : "00:00:00",
      "datetimeEpoch" : 1708621200,
      "temp" : 28.0,
      "feelslike" : 32.3,
      "humidity" : 81.96,
      "dew" : 24.6,
      "precip" : 0.0,
      "precipprob" : 0.0,
      "snow" : 0.0,
      "snowdepth" : 0.0,
```



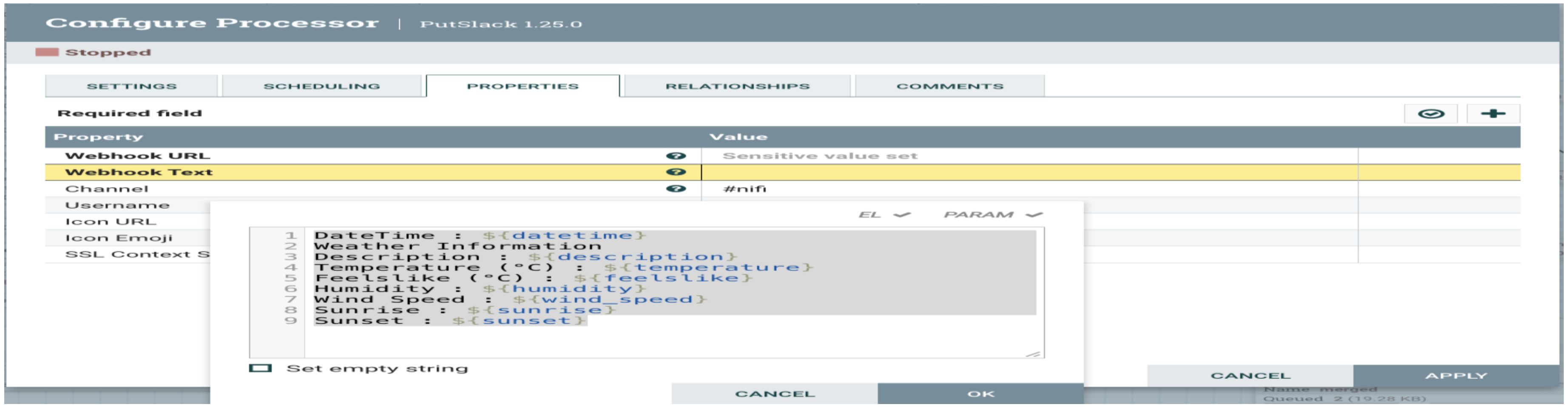
```
{
  "datetime" : "2024-02-23",
  "datetimeEpoch" : 1708621200,
  "tempmax" : 34.8,
  "tempmin" : 28.0,
  "temp" : 30.6,
  "feelslikemax" : 38.2,
  "feelslikemin" : 32.3,
  "feelslike" : 34.4,
  "dew" : 22.9,
  "humidity" : 65.6,
  "precip" : 0.0,
  "precipprob" : 0.0,
  "precipcover" : 0.0,
  "preciptype" : null,
  "snow" : 0.0,
  "snowdepth" : 0.0,
  "windgust" : 32.8,
  "windspeed" : 22.3,
  "winddir" : 184.6,
  "pressure" : 1011.6,
  "cloudcover" : 13.9,
  "visibility" : 17.6,
  "solarradiation" : 251.5,
  "solarenergy" : 21.8,
  "uvindex" : 10.0,
  "severerisk" : 30.0,
  "sunrise" : "06:37:59",
  "sunriseEpoch" : 1708645079,
  "sunset" : "18:25:09",
  "sunsetEpoch" : 1708687509,
  "moonphase" : 0.45,
  "conditions" : "Clear",
  "description" : "Clear conditions throughout the day.",
  "icon" : "clear-day",
  "stations" : [ "E4214", "VTBS", "VTBD" ],
  "source" : "comb",
  "hours" : [ {
    "datetime" : "00:00:00",
    "datetimeEpoch" : 1708621200,
    "temp" : 28.0,
    "feelslike" : 32.3,
    "humidity" : 81.96,
    "dew" : 24.6,
    "precip" : 0.0,
    "precipprob" : 0.0,
    "snow" : 0.0,
    "snowdepth" : 0.0,
    "preciptype" : null,
    "windgust" : 28.8,
    "windspeed" : 17.5,
    "winddir" : 196.0,
    "pressure" : 1012.0,
    "visibility" : 10.0,
    "cloudcover" : 25.0,
    "solarradiation" : 0.0,
    "solarenergy" : 0.0,
```


View as: formatted

```
{
  "datetime" : "2024-02-21",
  "temperature" : 21.9,
  "feelslike" : 21.9,
  "humidity" : 86.2,
  "sunrise" : "05:57:43",
  "sunset" : "18:42:33",
  "wind_speed" : 12.6,
  "description" : "Partly cloudy throughout the day with storms possible."
}
```


View as: formatted

```
{
  "datetime" : "2024-02-23",
  "temperature" : 30.6,
  "feelslike" : 34.4,
  "humidity" : 65.6,
  "sunrise" : "06:37:59",
  "sunset" : "18:25:09",
  "wind_speed" : 22.3,
  "description" : "Clear conditions throughout the day."
}
```



Required field

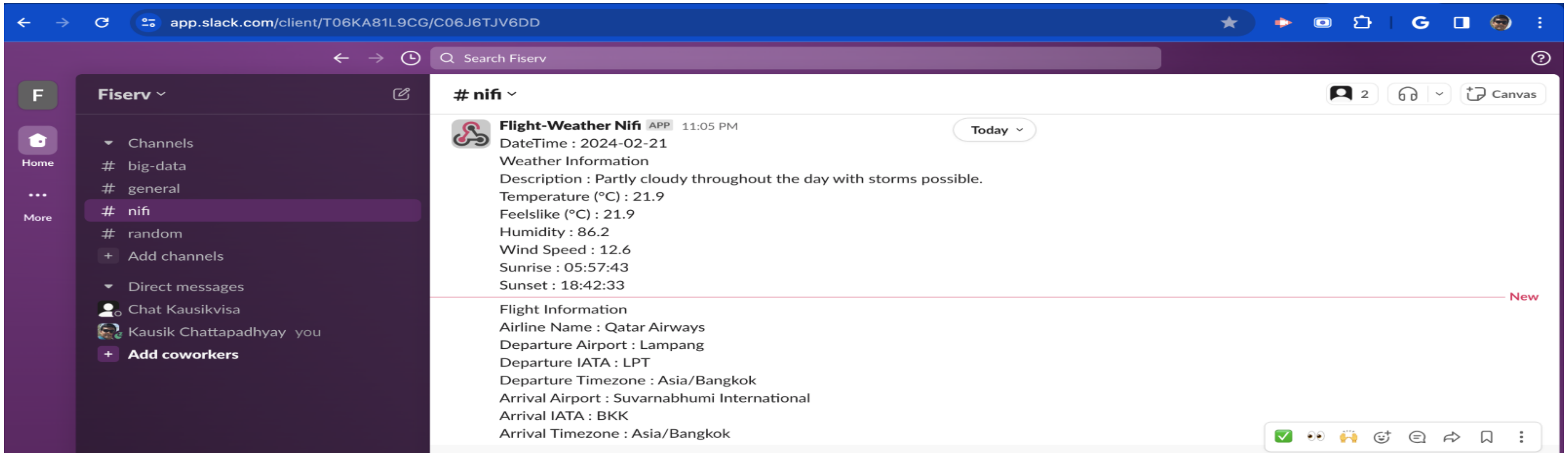
Property	Value
Webhook URL	Sensitive value set
Webhook Text	
Channel	#nifi
Username	
Icon URL	
Icon Emoji	
SSL Context S	

```
DateTime : ${datetime}
Weather Information
Description : ${description}
Temperature (°C) : ${temperature}
Feelslike (°C) : ${feelslike}
Humidity : ${humidity}
Wind Speed : ${wind_speed}
Sunrise : ${sunrise}
Sunset : ${sunset}
```

☐ Set empty string

Name merged
Queued 2 (19.28 KB)

Finally, notifications are coming to slack.



Flight-Weather Nifi APP 11:05 PM
DateTime : 2024-02-21
Weather Information
Description : Partly cloudy throughout the day with storms possible.
Temperature (°C) : 21.9
Feelslike (°C) : 21.9
Humidity : 86.2
Wind Speed : 12.6
Sunrise : 05:57:43
Sunset : 18:42:33
Flight Information
Airline Name : Qatar Airways
Departure Airport : Lampang
Departure IATA : LPT
Departure Timezone : Asia/Bangkok
Arrival Airport : Suvarnabhumi International
Arrival IATA : BKK
Arrival Timezone : Asia/Bangkok

Conclusion:

This project illustrates the versatility of Apache NiFi in orchestrating complex data flows across heterogeneous systems. By leveraging Apache NiFi processors and services, developers can efficiently integrate data from disparate sources, transform it as needed, and distribute it to multiple destinations, all within a unified and visually intuitive interface. Furthermore, the real-time notification capabilities provided by Apache NiFi enable proactive monitoring and alerting, enhancing overall system observability and responsiveness.

Keywords: Apache NiFi, Dataflow, APIs, MongoDB, HDFS, Slack Notifications.