Utilizing Real-Time Transit Data for Travel Optimization

Tim Spann

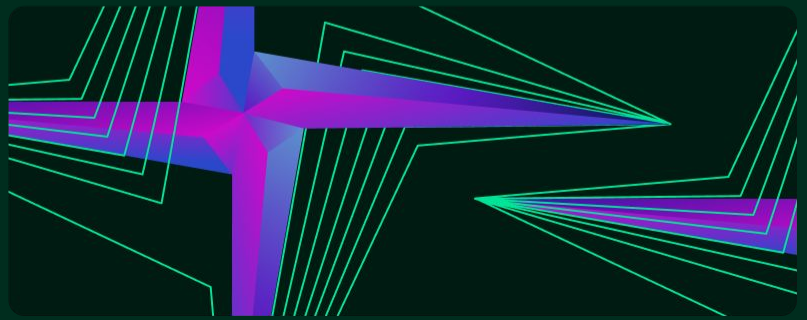
SR SE, Snowflake

DATA SCIENCE SUMMIT

A lecture selected by a Program Council consisting of recognized leaders in the Data and AI field.

Warsaw,
20.11.2025 - 21.11.2025

OFFICIAL LECTURE OF THE DATA SCIENCE SUMMIT



Event organized by: Academic Partners Foundation

Tim Spann

paasdev.bsky.social
@PaasDev // Blog: datainmotion.dev
Senior Solutions Engineer, Snowflake
NY/NJ/Philly - Cloud Data + AI Meetups
ex-Zilliz, ex-Pivotal, ex-Cloudera, ex-HPE, ex-StreamNative, ex-Hortonworks.

https://medium.com/@tspann
https://github.com/tspannhw

snowflake Streamlit

AI + Streaming Weekly by Tim Spann

https://bit.ly/32dAJft

This week in Snowflake, Apache NiFi, Apache Flink, Apache Kafka, ML, AI, Streamlit, Jupyter, Apache Iceberg, Apache Polaris, Python, Java, LLM, GenAI, Vectors and Open Source friends.

The MTA Data Challenge

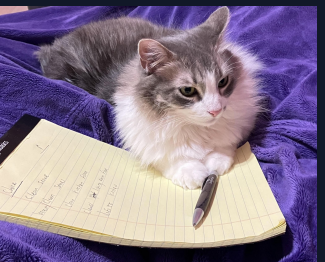
- The largest transit system in the US.
- Generates a 'staggering' amount of **data** every minute.
- Data is siloed: GTFS-RT for subways, SIRI for buses.
- Key data (like service status) can be in outdated, hard-to-parse XML formats.
- Internal tools often run reports for the *next day*, not in real-time.

stockcake.com

Our Objective: From "Next Day" to "Right Now"

1

Single, Real-Time View



The Goal

To build a predictive system that processes thousands of real-time data points, avoids delays, and provides the *truly* optimal route, right now.

We want to move from historical analysis to real-time, actionable decisions.

Fueling the System: Data Sources

MTA GTFS-RT

Real-time subway vehicle positions, trip updates, and service alerts.

MTA SIRI

Real-time bus locations, estimated times of arrival, and service status.

External Data

Real-time weather APIs,NYC traffic cameras, news, search, public safety alerts, and city event schedules.

General (Google) Transit Feed Specification

GTFS - Protocol Buffers (binary format)

Organizations:

- Open Transit Software Foundation
- Mobility Data
- Google
- GTFS.org

Transit Real-Time Data Pipeline

Ingest (Apache NiFi)

Capture, transform, and route all diverse data streams.

Store (Snowflake)

Land raw & prepared data in a scalable cloud platform.

Structure (Iceberg)

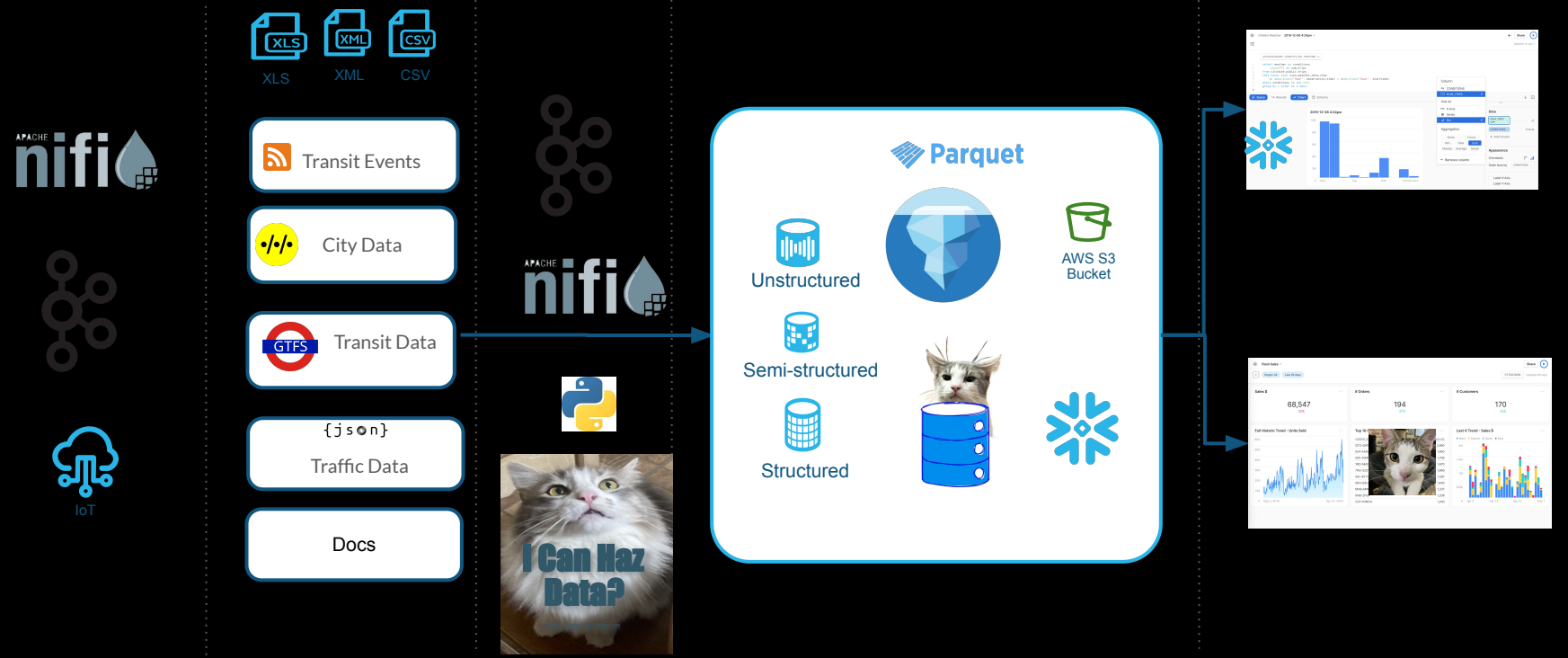
Create an open, high-performance table format.

Analyze (AI/LLM)

Feed structured, real-time data to models.

SNOWFLAKE CORTEX AGENTS

Real-Time AI Open Lakehouse with Open Data



Step 1: Ingest with NiFi

The Data Flow "Swiss Army Knife"

Apache NiFi is the key to taming this chaos.

- Visually connects to any source (REST, Kafka, Protobuf, XML...).
- Perfect for parsing messy, outdated formats like RSS and XML
- Guarantees data delivery & full data provenance.

Step 2: Store & Structure

Snowflake

The cloud **data** platform. It decouples storage & compute, handling massive scale & concurrency for our **AI** queries.

Apache Iceberg

The open table format. **Apache Iceberg** enables high-performance analytics on the data lake and provides time-travel & schema evolution for our models.

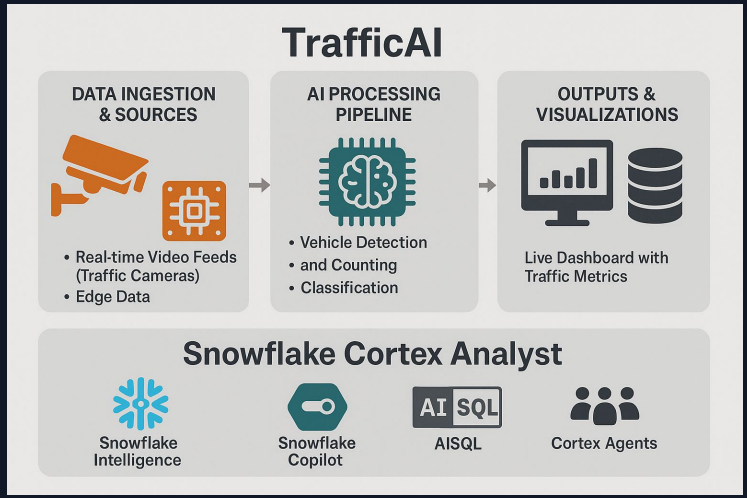
Step 3: The Brain (AI & GenAI)

Predictive AI: Models (like XGBoost) forecast delays based on current crowding, weather, and known signal issues.

Generative AI (LLMs): The **LLM** translates complex model outputs ("Delay_Prob_A=0.85") into simple, natural language advice.

Real-Time Decisions: The entire system runs on a continuous flow of **data**, allowing the **GenAI** to update its recommendations at any moment.

Cortex AI SQL




```
SELECT SNOWFLAKE.CORTEX.COMPLETE("claude-4-5-sonnet",
    "Analyze this traffic image and describe what you see.
Respond in compact JSON format.",
    TO_FILE("@TRAFFIC", "cars.jpg")) INTO :result;
```

Easy, Connected, Trusted

SNOWFLAKE CORTEX AI

AGENTIC BUSINESS INSIGHT

- Agent Apps — **Snowflake Intelligence** PU Soon
- Agent APIs — **Cortex Agents** GA Soon

STATE OF THE ART RETRIEVAL

- Unstructured Data Retrieval — **Cortex Search**
- Structured Data Retrieval — **Cortex Analyst**

SCALABLE AI PROCESSING

- Multimodal AI-powered SQL — **Cortex AISQL** PU
- Document Processing — **Document AI, Parse, Embed**

MODELS

OpenAI
ANTHROP\C
Meta
MISTRAL AI_
deepseek
snowflake

GOVERNANCE

- RBAC
- Guardrails
- Evaluations
- Monitoring
- AI Gateway

DATA — STR, DOC, AUDIO, IMAGE

DATA INGESTION

Fully managed extraction, load, preprocess, and activation of multimodal data

snowflake MARKETPLACE

Putting It All Together

Avoid the 2 train. Based on current signal problems at 72nd St and platform overcrowding at Times Square, I recommend taking the Q train to 34th St. This will save you an estimated 15 minutes.

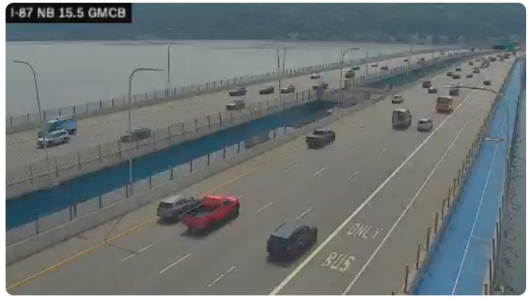
— Your Real-Time Travel Assistant

Streamlit Apps New York City Map

Packages

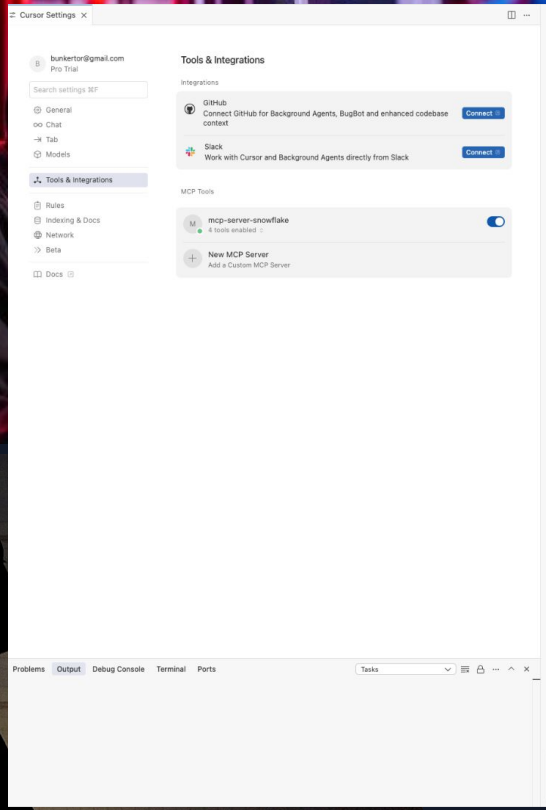
```python
# Import python packages
import streamlit as st
from snowflake.snowpark.context import get_active_session
from snowflake.snowpark.functions import *
from snowflake.snowpark.types import *
import json
import pandas as pd
import numpy as np
import pydeck as pdk
from snowflake.snowpark.files import SnowflakeFile
from snowflake.snowpark.functions import col
from streamlit.components.v1 import html

st.set_page_config(layout="wide")

st.title(f"New York City")

session = get_active_session()

traffic_latlon = session.table('DEMO.DEMO.VWNYCTRAFFIC')

st.markdown('#### Traffic Station Locations')

# Convert to pandas DataFrame
df = traffic_latlon.to_pandas()

# Create PyDeck layer
layer = pdk.Layer(
    "ScatterplotLayer",
    df,
    get_position=["LONGITUDE", "LATITUDE"],
    get_color=[255, 0, 0, 160],
    get_radius=100,
    pickable=True
)

# Create view state centered on NYC
view_state = pdk.ViewState(
    latitude=40.7128,
    longitude=-74.0060,
    zoom=10,
    pitch=0
)

# Create deck with view state
deck = pdk.Deck(
    layers=[layer],
    initial_view_state=view_state,
    map_style="mapbox://styles/mapbox/light-v9",
```

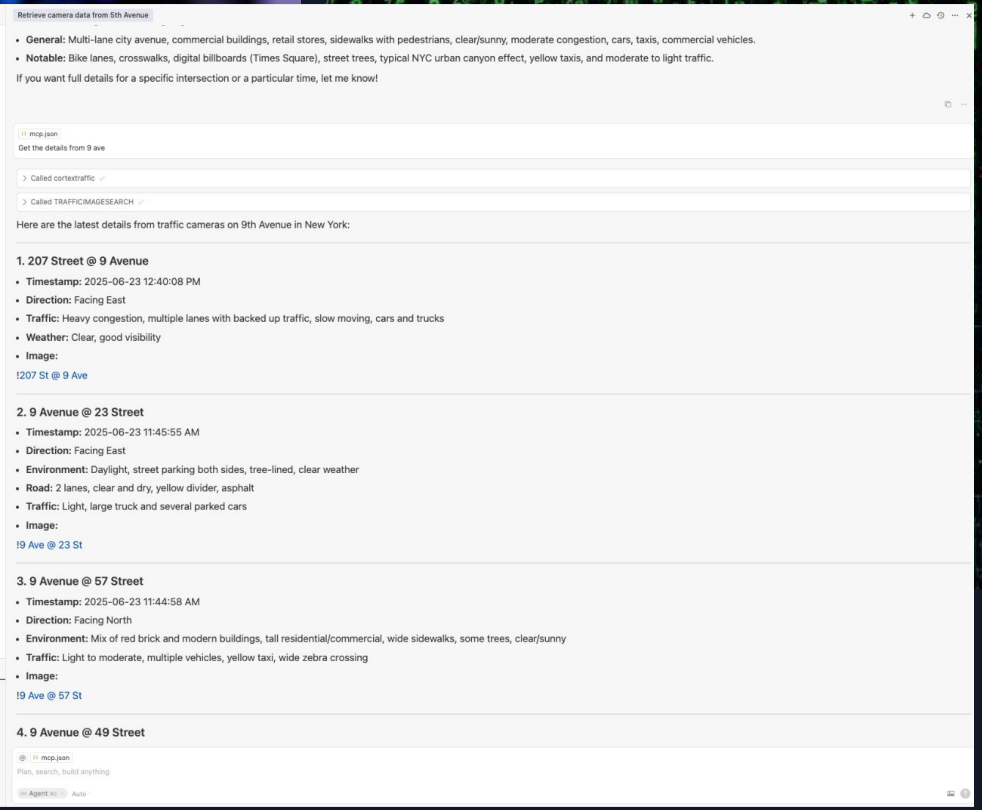

```
{
  "confidence_level" : "high"
  "total_count" : 12
  "vehicle_types" : [
    0 : "cars"
    1 : "SUVs"
    2 : "pickup trucks"
  ]
}
```


```
{
  "confidence_level" : "high"
  "total_count" : 8
  "vehicle_types" : [
    0 : "passenger cars"
    1 : "SUV"
    2 : "pickup truck"
  ]
}
```


Model Context Protocol





Key Takeaways

Apache NiFi is Key

Apache NiFi is essential for taming diverse, messy, and real-time sources.

Open Foundations

Snowflake & **Apache Iceberg** provide a scalable, open foundation for **AI**.

AI Makes Data Actionable

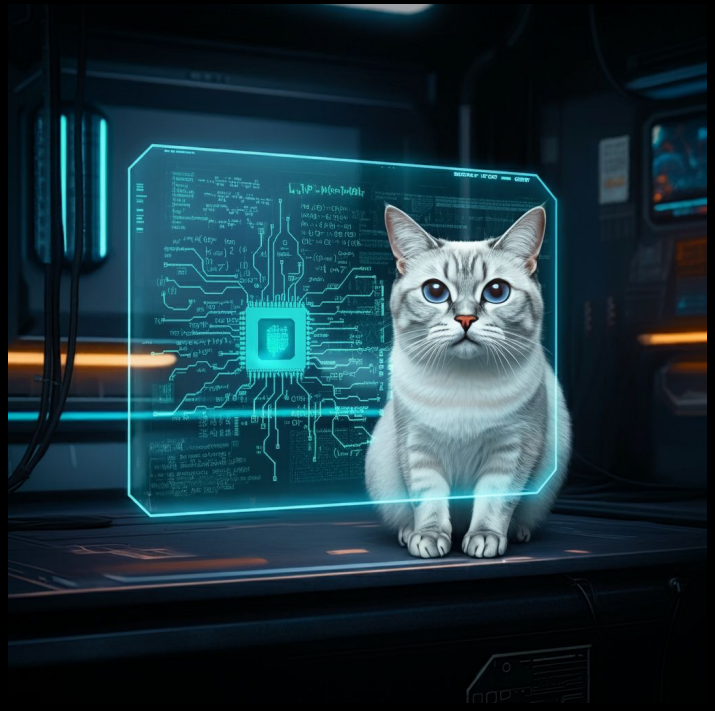
AI/LLMs turn raw numbers into simple, human-readable advice.

Questions?

https://github.com/tspannhw/TrafficAI

DEMO

RESOURCES



https://511ny.org/developers/help

https://github.com/tspannhw/hackathon2025

http://web.mta.info/developers/data/nyct/subway/StationComplexes.csv

https://data.ny.gov/Transportation/MTA-Daily-Ridership-and-Traffic-Beginning-2020/sayj-mze2/about_data

https://data.ny.gov/Transportation/MTA-Bus-Service-Delivered-2015-2019/tw28-zvtk/about_data

https://data.ny.gov/Transportation/MTA-Congestion-Relief-Zone-Vehicle-Entries-Beginni/t6yz-b64h/about_data

Open New York Data Links

Developer Signup
https://511ny.org/developers/resources

Cameras
https://511ny.org/api/getcameras?key=SignUpForFreeKey&format=json

Alerts
https://511ny.org/api/getalerts?key=SignUpForFreeKey&format=json

Events
https://511ny.org/api/getevents?key=SignUpForFreeKey&format=json

Message Signs
https://511ny.org/api/getmessagesigns?key=SignUpForFreeKey&format=json

MTA Subway Hourly Ridership
https://data.ny.gov/Transportation/MTA-Subway-Hourly-Ridership-Beginning-2025/5wq4-mkjj/about_data

Open New York Data Links

MTA Stations
https://data.ny.gov/resource/i9wp-a4ja.json

MTA Bus
https://bustime-beta.mta.info/developers/siri/#
https://bustime-beta.mta.info/developers/siri/monitored-vehicle-journey

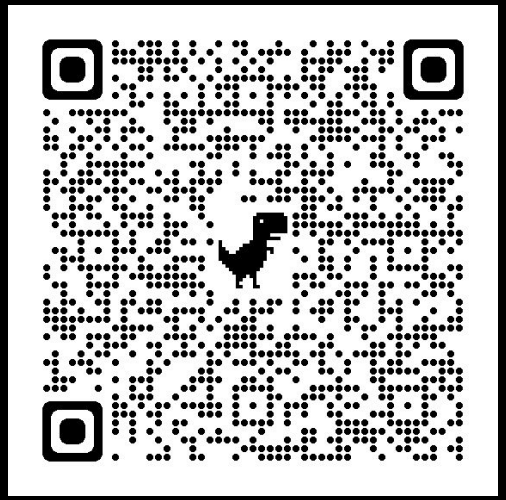
DOT Speed
https://data.cityofnewyork.us/Transportation/DOT-Traffic-Speeds-NBE/i4gi-tjb9/about_data
https://dev.socrata.com/foundry/data.cityofnewyork.us/i4gi-tjb9
https://medium.com/cloudera-inc/nyc-traffic-are-you-kidding-me-6d3fa853903b

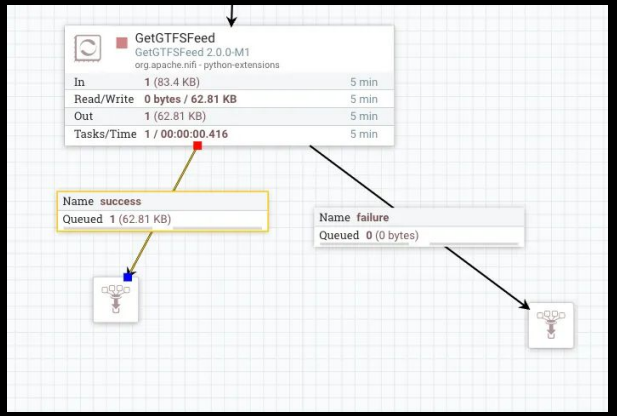
Deeper Resources

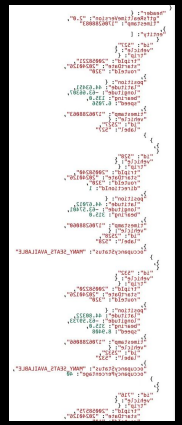
https://github.com/tspannhw/TrafficAI
https://medium.com/@tspann/automating-transit-gtfs-flows-7a5c4d3dafe8
https://medium.com/cloudera-inc/iteration-1-building-a-system-to-consume-all-the-real-time-transit-data-in-the-world-at-once-4322b160df9d
https://medium.com/cloudera-inc/iteration-2-building-a-system-to-consume-all-the-unsecured-real-time-transit-data-in-the-world-5f365c0cf240
https://medium.com/cloudera-inc/subways-and-transit-updates-in-real-time-30c104c359ef
https://medium.com/data-engineering-with-dremio/introducing-apache-iceberg-1-9-0-native-geospatial-support-enhanced-row-lineage-and-more-dead8950d391
https://carto.com/blog/iceberg-geoparquet-carto-reshaping-geospatial
https://forrest.nyc/airflow-ai-iceberg-v3-the-new-stack-for-scalable-geospatial-data/
https://medium.com/@tspann/automating-transit-gtfs-flows-7a5c4d3dafe8
https://medium.com/@tspann/real-time-irish-transit-analytics-ea76164c9595
https://medium.com/cloudera-inc/transit-in-sao-paulo-brasil-flank-style-eaec6753cc63
https://medium.com/cloudera-inc/flank-for-halifax-canada-transit-2d4dc5b8ad64

Get GTFS Data

- Python 3.10+
- GTFS from Transit URL
- Alerts, Trip Updates or Vehicle Positions
- Returns JSON
- google.transit and google.protobuf

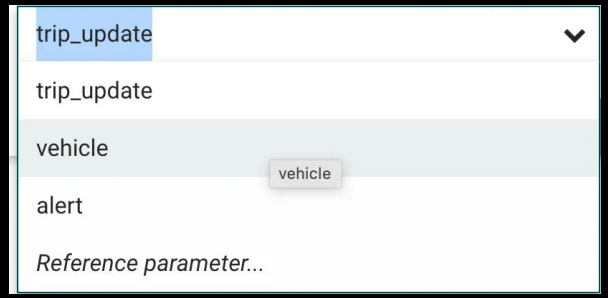


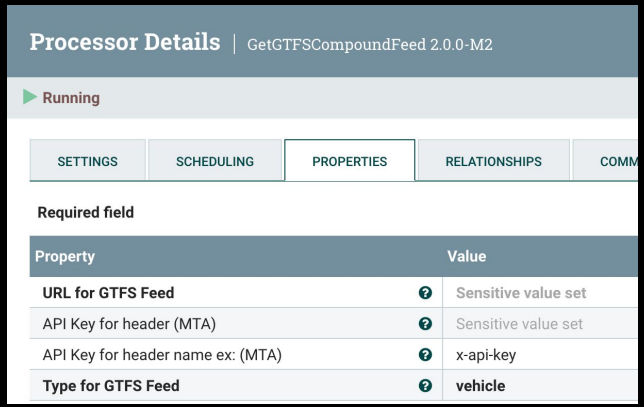




Get Compound GTFS Data

- Python 3.10+
- GTFS to JSON





https://github.com/tspannhw/FLaNK-python-processors/blob/main/GetGTFSCompoundFeed.py

Address To Lat/Long

- Python 3.10+
- geopy Library
- Nominatim
- OpenStreetMaps (OSM)
- openstreetmap.org/copyright
- Returns as attributes and JSON file
- Works with partial addresses
- Categorizes location
- Bounding Box

https://github.com/tspannhw/FLaNKAI-Boston

Thank you for watching!

Remember to leave your questions
and rate the presentation
in the section below.