LEVEL2_Weather_Data_Ingestion

BTTF Data Engineer Assignment

Objective

Design and implement a weather data ingestion pipeline that fetches hourly weather forecasts for each city involved in shipments, enriches them with relevant metadata, and stores the results in the raw data zone.

This serves as the external data augmentation source for downstream analysis and KPIs (e.g., fuel consumption vs. weather conditions).

API: Open-Meteo

- **URL**: https://open-meteo.com
- **Why Chosen**:
  - Free & no auth token needed
  - Allows historical + forecasted hourly data
  - Returns clean, structured JSON

Parameters Used:

Parameter	Reason
`temperature_2m`	Affects fuel usage
`windspeed_10m`	Affects driving speed & efficiency
`precipitation`	Rain increases fuel usage
`weathercode`	High-level weather summary
`time`	Needed for temporal joins
`lat` , `lon`	Required per city for API call

Data Source: `shipments.cities`

Pulled from PostgreSQL using the following query:

```
SELECT name, latitude, longitude
FROM shipments.cities
WHERE latitude IS NOT NULL AND longitude IS NOT NULL;
```

Script: weather_fetch.py

- **Language**: Python 3.12
- **Location**: scripts/ingestion/weather_fetch.py

Functional Breakdown

Step	Description
Connect to DB	Pulls all cities with valid coordinates
API Calls	Loops through each city, calls Open-Meteo hourly API
Data Transformation	Flattens JSON to structured tabular format
Logging	Tracks each success/failure, retries, and outputs
Save to File	CSV is written to data/raw/weather/
Retry Logic	Up to 3 attempts per city for resilience

Step	Description
Progress Bar	Added using tqdm for real-time CLI feedback
.env Config	DB credentials secured via dotenv

Retry Logic

- Retries failed API calls up to 3 times
- Includes exponential backoff (delay) via time.sleep()
- All attempts logged to logs/weather.log

Output Format

- **Location**: data/raw/weather/weather_data_YYYY_MM_DD.csv
- **Columns**:
  - city, latitude, longitude, timestamp
  - temperature_2m, windspeed_10m, precipitation, weathercode

---

Logs

- **Location**: logs/weather.log
- **Contents**:
  - DB connection success/failure
  - API request status per city
  - Retry attempts + final failures
  - CSV write confirmation

Test Checklist

Component	Status
PG Connection	✅
API Calls	✅
Retry Logic	✅
CSV Output	✅
Logging	✅
tqdm Progress Bar	✅

Next Step

→ Proceed to LEVEL3_Data_Modeling - a unified data model with schema definitions that integrates shipment and weather data to support analytical use cases like fuel efficiency, city-wise performance, and weather-based metrics.