NiFi to PostgreSQL ETL Pipeline

This document describes the configuration of a NiFi pipeline that fetches data from an external API (uselessfacts.jsph.pl) and stores it into a PostgreSQL table (etl).

Controller Service: PostgreSQL
- JDBC URL: jdbc:postgresql://postgres:5432/airflow
- Driver Class: org.postgresql.Driver
- Driver Path: /opt/nifi/nifi-current/lib/custom/postgresql-42.7.5.jar
- Database User: airflow
- Password: airflow

ExecuteSQL
Query:

```
CREATE TABLE IF NOT EXISTS etl (
    id VARCHAR(255) PRIMARY KEY,
    text TEXT NOT NULL,
    source VARCHAR(255),
    source_url TEXT,
    language VARCHAR(10),
    permalink TEXT
);
```

- Database Connection Pooling Service: DBCP connection pool
- Relationship: terminate (except for success)

API Endpoint

- URL: https://uselessfacts.jsph.pl/api/v2/facts/random

InvokeHTTP

- Relationship: terminate (all except for response)

EvaluateJsonPath

Properties:

- id -> $.id
- language -> $.language
- permalink -> $.permalink
- source -> $.source
- source_url -> $.source_url
- text -> $.text

- Relationship: terminate (all except for matched)

PutDatabaseRecord

- Record Reader: JsonTreeReader
- Database Type: PostgreSQL
- Statement Type: upsert
- Database Connection Pool: DBCP connection pool
- Update Keys: id
- Relationship: terminate (all)

Data Flow Summary

1. ExecuteSQL -> Ensures etl table exists in PostgreSQL.
2. InvokeHTTP -> Fetches random fact JSON from API.
3. EvaluateJsonPath -> Extracts relevant fields from JSON response.
4. PutDatabaseRecord -> Inserts/updates records in PostgreSQL.