**Test the new api function in prototyping to confirm csv is saving**

**Project: Data Pipeline for Real-Time News Sentiment Analysis**

**Overview**

Build an Apache Airflow pipeline that fetches news articles from an API, processes the text to extract sentiment insights, and stores the results in a database. Additionally, the pipeline will generate visualizations of sentiment trends over time.

**Key Features**

* **Data Collection (Incremental Processing & Scheduling)**
  + Use an API (e.g., NewsAPI) to fetch the latest news articles.
  + Schedule the pipeline to run at regular intervals.
  + Implement backfilling to process historical news data.
* **Data Processing (Atomicity & Idempotency)**
  + Extract relevant fields (e.g., headline, description, publication date): Used execution date to filter on the url. Publication date was then added as a second column.
  + Use NLP (e.g., Vader, TextBlob, or Hugging Face models) for sentiment analysis.
  + Ensure atomicity by making each task self-contained and idempotent.
* **Task Dependencies & Conditional Execution**
  + Branch logic: Only analyze sentiment if the article is in English.
  + Conditional tasks: Store results in different tables based on sentiment score.
* **Data Storage & XComs for Data Passing**
  + Store raw articles in a PostgreSQL or SingleStore database.
  + Store processed sentiment scores in another table.
  + Use XComs to pass data between tasks.
* **Templating & Visualization**
  + Use Jinja templating to dynamically insert timestamps.
  + Generate and store sentiment trend visualizations (e.g., bar charts, time series).
* **DAG Monitoring & Failure Handling**
  + Implement retry logic and failure alerts.
  + Log errors and handle API rate limits.

**Tech Stack**

* Apache Airflow
* Python (for sentiment analysis & API integration)
* PostgreSQL/SingleStore (for data storage)
* Pandas & Matplotlib/Seaborn (for visualization)
* Grafana (optional, for real-time dashboards)