# **Documentation: Coffee Supply Chain Data Gathering**

## **Overview**

This script collects data on:

1. **Coffee-related News**: Fetches articles about coffee production, trade, prices, and climate-related issues from the News API.
2. **Weather Data**: Retrieves real-time weather information for specific coffee-producing regions using the OpenWeatherMap API.

The collected data is saved in CSV files for further analysis, aiding in supply chain disruption prediction and optimization.

## **Dependencies**

Ensure the following Python packages are installed:

* requests: For sending HTTP requests to APIs.
* pandas: For handling and processing the news data.
* csv: For writing weather data to a CSV file.

Install the required packages using:

pip install requests pandas

## **Part 1: Coffee-Related News**

### ****Function:**** fetch\_coffee\_news()

This function fetches news articles related to coffee from the News API.

#### **Parameters**

* None directly. The API key is stored as a global variable (NEWS\_API\_KEY).

#### **Key Query Parameters**

* q: Keywords for searching coffee-related news.
* from: Date range for news articles (last 30 days).
* sortBy: Sorts articles by relevance.
* language: Restricts articles to English.
* pageSize: Fetches up to 100 articles.

#### **Returns**

* A pandas DataFrame containing the following fields:
  + Source (news outlet)
  + Title
  + Description
  + Content
  + Published Date
  + URL

## **Part 2: Weather Data for Coffee-Producing Regions**

### ****Global Variables****

* API\_KEY: OpenWeatherMap API key for authentication.
* url: Base URL for fetching current weather data.
* coffee\_regions: List of specific coffee-producing regions and cities.

### ****Function:**** fetch\_weather(region\_name, api\_key)

Fetches current weather data for a specific region.

#### **Parameters**

* region\_name (str): Name of the region or city (e.g., "Manaus, Brazil").
* api\_key (str): OpenWeatherMap API key.

#### **Returns**

* A JSON object containing weather data if the request is successful.
* None if the request fails.

#### **Weather Data Fields**

* Temperature (°C)
* Humidity (%)
* Weather Description (e.g., "clear sky")
* Wind Speed (m/s)

## **Main Logic**

### ****1. Fetch News Data****

* Calls fetch\_coffee\_news() to gather news articles.
* Saves the data in coffee\_related\_news\_data.csv.

### ****2. Fetch Weather Data****

* Iterates through coffee\_regions and calls fetch\_weather() for each region.
* Saves the results in coffee\_producing\_regions\_weather.csv.

#### **CSV Output Format**

* Region
* Temperature (°C)
* Humidity (%)
* Weather
* Wind Speed (m/s)

#### **Saving Weather Data**

The weather data is written to a CSV file using the csv.DictWriter class. The CSV file includes a header row with specific field names.

## **Execution Workflow**

1. **Initialize APIs**: Set up the News API and OpenWeatherMap API keys.
2. **Fetch Coffee News**: Use the fetch\_coffee\_news() function to retrieve relevant news articles and save them to a CSV file.
3. **Fetch Weather Data**: Loop through the coffee-producing regions, calling fetch\_weather() for each, and save the weather data to another CSV file.
4. **Error Handling**: The script handles errors gracefully, displaying appropriate error messages if an API request fails.

## **File Outputs**

1. **News Data**: coffee\_related\_news\_data.csv
   * Contains articles about coffee-related topics.
2. **Weather Data**: coffee\_producing\_regions\_weather.csv
   * Contains weather information for coffee-producing regions.