Linkedin Connections Scrapper Script Documentation

Overview

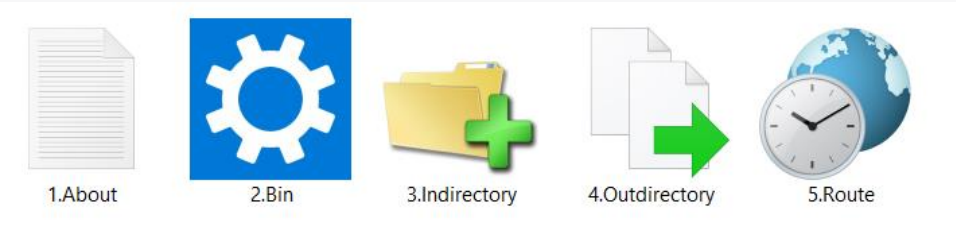
This documentation outlines the functionality, structure, and key components of the web scraping script designed to process CSV files, extract data using Google Custom Search API, and store the results in designated folders. The script is responsible for comparing and processing data present in the input directory and storing processed and unprocessed data in the output directories.

Script Components

1. Introduction

This script is designed to process CSV files containing contact information, utilizing the Google Custom Search API to extract additional data. It performs data comparison to avoid duplication and redundancy while processing new data.

2. Folders Structure



The script operates within a set of folders:

- **About :** Contains documentation of the application.
- **Bin** : Contains the generated code.
- **Indirectory** : Contains CSV files with input data.
- **Outdirectory** : Contains subfolders '1.Extract' and '2.Error'.
    - '1.Extract' stores processed data.
    - '2.Error' stores unprocessed data.
- **Route** : Contains CSV files ('APIkey.csv' and 'CSEid.csv') with API keys and CSE IDs.

3. API Keys and CSE IDs

The script accesses API keys and CSE IDs from the 'Route' folder to authenticate with the Google Custom Search API. The API keys and CSE IDs are stored in 'APIkey.csv' and 'CSEid.csv' files, respectively.

4. Processing Logic

Sorting and Comparison

- The script loads existing processed data from the 'Extract' folder to avoid reprocessing.
- For each input CSV file in the 'Indirectory' folder, it reads rows and compares with existing processed data.
- If data matches with processed data, it's skipped; otherwise, data is processed.

Data Extraction and Storage

- The script constructs a search query using columns from the input data.
- It uses the Google Custom Search API to fetch data related to the search query.
- Extracted data is stored in a DataFrame with additional information like link, description, and snippet.
- Processed data is saved in CSV format in the 'Extract' folder with a filename in the format "Extract-date-time.csv".

Error Handling

- If data extraction fails for a row, the script retries up to a specified number of times.
- If still unsuccessful, the row is added to the error list.

Processing Indirectory CSV Files

- For each CSV file in the 'Indirectory' folder, the script processes rows using the described logic.

5. Multiple API Keys and CSE IDs

- The script uses multiple API keys and CSE IDs for processing to avoid API limitations.
- It sequentially selects API keys and CSE IDs from the respective CSV files for each query.
- After exhausting the API key's daily limit, the script switches to the next available API key.

6. Displaying Progress

- During processing, the script prints information to the terminal.
- It shows the row being processed, the API key in use, and any errors encountered.