Weather Scraping Project Report

Batuhan Turgay

13/11/2023

Introduction

The project focuses on gathering weather data for various provinces in Turkey from three distinct websites: Met Office, HavadurumuX, and Weather.com. Python serves as the primary programming language, with Selenium for web automation, BeautifulSoup for HTML parsing, and pymongo for MongoDB interactions. This report provides an in-depth overview of the project, emphasizing library choices, coding methodologies, and operational details.

Project Overview

The goal of the project is to create a robust and efficient weather data scraper for multiple Turkish cities. The script fetches high and low temperatures for the next 7 days from different weather websites concurrently. The choice of multiple sources ensures data accuracy and reliability. The collected data is then stored in a MongoDB database for easy access and retrieval.

Library Choices and Methodologies

1. Web Scraping Functions

Met Office (`get_weather_metoffice`):

- **Selenium Library:**
  - *Why:* Met Office employs dynamic content loading through JavaScript, necessitating an interactive approach for data retrieval.
  - *How:* Selenium automates browser interactions, simulating user actions to input search queries and extract temperature data.

HavadurumuX (`get_weather_havadurumux`):

- **Requests and BeautifulSoup Libraries:**
  - *Why:* HavadurumuX relies less on JavaScript, allowing for a simpler HTTP request and HTML parsing approach.
  - *How:* The `requests` library fetches HTML content, and BeautifulSoup extracts temperature information from the parsed HTML.

Weather.com (`get_weather_weather`):

- **Selenium Library and BeautifulSoup:**
  - *Why:* Similar to Met Office, Weather.com requires interactive browsing for data extraction.
  - *How:* Selenium navigates the website, inputs search queries, and BeautifulSoup aids in parsing the HTML for temperature data extraction.

2. MongoDB Connection

- **pymongo Library:**
  - *Why:* MongoDB is chosen for its flexibility and scalability, and pymongo is the official driver for Python.
  - *How:* The `MongoClient` establishes a connection, and MongoDB serves as the storage solution for the scraped weather data.
  - *DB Name* DB name is batuhan_turgay

3. Scraping Algorithm

- **Dictionary Usage (TR_metoffice and TR):**
  - *Why:* Dictionaries facilitate mapping province codes to city names, streamlining the iteration through cities.
  - *How:* The script iterates through the `TR` dictionary, determining which scraping functions to use based on the presence of cities in the `TR_metoffice` dictionary.

4. Multithreading

- **concurrent.futures Library:**
  - *Why:* To enhance efficiency, multithreading is introduced for concurrent scraping of weather data for multiple cities.
  - *How:* The `ThreadPoolExecutor` allows parallel execution of scraping tasks, improving overall performance.

Operational Details

Instructions for Running the Project

1. **Install Dependencies:**
   - Ensure the necessary Python packages (`bs4`, `pymongo`, `requests`, `tqdm`, `selenium`, `concurrent.futures`, `webdriver_manager`) are installed.
2. **Web Driver:**
   - Install Chrome as the web driver since the script is configured for Chrome. Make sure Chrome is available on your machine.
3. **MongoDB:**
   - Configure the MongoDB connection details according to your setup (local or MongoDB Atlas).
4. **Run the Script:**
   - Execute the Python script. The script will fetch weather data for specified cities from multiple sources and store it in the MongoDB database.

Conclusion

In conclusion, the project showcases an effective and versatile weather data scraper for Turkish cities. The selection of libraries and methodologies reflects a balance between the requirements of dynamic content loading, simplicity, and data storage. Users can follow the provided instructions to set up and run the script for their specific use cases.

Execution Time

The script's execution time is monitored, offering insights into its efficiency. During testing, the script demonstrated reliable performance, successfully collecting and storing weather data from various sources.

Execution time depends on the processing speed of the computer and the quality of the internet connection.

For further inquiries or assistance, please refer to the contact information provided within the script (BT).