Scraping mit Python

1. Zweck

Der Scrapper führt eine Google-Suche mit den vom Benutzer angegebenen Parametern durch und speichert die Ergebnisse.

2. Funktionsweise

Das Skript besteht aus mehreren Funktionen:

- **google_search_api(search_term, num_results=10, country=None, language=None, dateRestrict=None, fileType=None, **kwargs)**:
  - Führt die Google-Suche mit den angegebenen Parametern durch.
  - Verwendet die Google Custom Search API.
  - **Parameter:**
    - search_term: Der Suchbegriff.
    - num_results: Die Anzahl der zurückzugebenden Ergebnisse. Standardwert ist 10.
    - country: Beschränkt die Suche auf ein bestimmtes Land (z. B. us, uk, de).
    - language: Beschränkt die Suche auf eine bestimmte Sprache (z. B. "lang_en", "lang_de").
    - dateRestrict: Beschränkt die Suche auf einen bestimmten Zeitraum (z. B. "d7" für die letzten 7 Tage).
    - fileType: Beschränkt die Suche auf einen bestimmten Dateityp (z. B. pdf, doc).
    - **kwargs: Zusätzliche Parameter, die von der Google Custom Search API unterstützt werden.
  - Gibt eine Liste von Dictionaries zurück, wobei jedes Dictionary ein Suchergebnis mit den Schlüsseln "title", "link" und "snippet" darstellt. Gibt None zurück, wenn ein Fehler auftritt.

- **save_results(results, filename="google_results", output_format="json", db_name="search_results.db")**:
  - Speichert die Suchergebnisse in dem angegebenen Format.
  - **Parameter:**
    - results: Die von google_search_api zurückgegebenen Suchergebnisse.
    - filename: Der Name der Ausgabedatei oder der Name der Tabelle in der Datenbank. Standardwert ist "google_results".
    - output_format: Das Ausgabeformat. Mögliche Werte sind "json", "csv" und "db". Standardwert ist "json".

        - db_name: Der Name der SQLite-Datenbank, wenn output_format "db" ist. Standardwert ist "search_results.db".
- **get_user_input(prompt, allowed_values=None)**:
    - Ruft die Benutzereingabe ab und validiert sie.
    - **Parameter:**
        - prompt: Die dem Benutzer anzuzeigende Eingabeaufforderung.
        - allowed_values: Eine Liste von zulässigen Eingabewerten. Wenn angegeben, wird die Eingabe des Benutzers gegen diese Werte validiert.
    - Gibt die Benutzereingabe als Kleinbuchstaben-String zurück.

3. Anforderungen und Parametrisierung

Das Skript erfüllt die folgenden Anforderungen:

- **Speicherung der Ergebnisse in geeigneter Form:** Die Ergebnisse werden in einer der folgenden Formen gespeichert:
    - JSON-Datei: Leicht lesbares Format für die Datenübertragung.
    - CSV-Datei: Geeignet für die Tabellenkalkulation und Datenanalyse.
    - SQLite-Datenbank: Ermöglicht die strukturierte Speicherung und Abfrage der Daten.
- **Parametrisierung der Anforderungen:** Das Skript ist hochgradig parametrisiert, sodass Suchanfragen ohne Code-Änderungen angepasst werden können. Der Benutzer wird zur Eingabe der folgenden Parameter aufgefordert:
    - Suchbegriff
    - Anzahl der Ergebnisse
    - Land
    - Sprache
    - Zeitraum
    - Dateityp
    - Ausgabeformat
- **Dokumentation:** Dieses Dokument dient der Dokumentation des Skripts.

4. Verwendung

1. **Voraussetzungen:**
    - Python 3 ist installiert.
    - Die erforderlichen Python-Bibliotheken sind installiert. Führen Sie folgenden Befehl aus:
      pip install --upgrade google-api-python-client

2. **Konfiguration:**
    - Ersetzen Sie im Skript die Platzhalter API_KEY und SEARCH_ENGINE_ID durch

Ihre tatsächlichen Werte.
   - Der API-Schlüssel wird von der Google Cloud Console bezogen.
   - Die Search Engine ID wird im Control Panel der Programmable Search Engine von Google abgerufen.
3. **Ausführung:**
   - Führen Sie das Skript von der Befehlszeile aus:
     python skriptname.py

   - Das Skript führt Sie durch den Prozess der Eingabe der Suchparameter.
   - Die Ergebnisse werden im angegebenen Format gespeichert.

5. Beispiele

Beispiel 1: Suche nach den neuesten Nachrichten zum Klimawandel und Speichern im JSON-Format:

```
What would you like to search for? Klimawandel
How many results would you like (1-10)? 5
Enter country: us, uk, ch, de: de
Enter language: lang_en, lang_de: lang_de
Enter dateRestriction: d[number], w[number], m[number], y[number]: d7
Enter fileType: pdf, doc, docx, xls, xlsx, ppt, pptx:
Enter output format json, csv, db: json
Results saved to 'google_results.json'
```

Beispiel 2: Suche nach Dokumenten zum Thema "Machine Learning" und Speichern in einer Datenbank:

```
What would you like to search for? Machine Learning
How many results would you like (1-10)? 10
Enter country: us, uk, ch, de:
Enter language: lang_en, lang_de:
Enter dateRestriction: d[number], w[number], m[number], y[number]:
Enter fileType: pdf, doc, docx, xls, xlsx, ppt, pptx:
Enter output format json, csv, db: db
Results saved to table 'google_results' in 'search_results.db'
```

6. Fehlerbehandlung

Das Skript enthält eine grundlegende Fehlerbehandlung für den Fall, dass bei der API-Anfrage ein Fehler auftritt. In diesem Fall wird eine Fehlermeldung ausgegeben und None zurückgegeben.