PHP Web Scraper für reguläre Ausdrücke
Zusammenfassung

Monday 6th January, 2020 - 13:44

Flavio De Jesus Matias
Universität Luxemburg
Email: flavio.dejesus.001@student.uni.lu

Giacomo Di Tollo
Universität Luxemburg
Email: giacomo.ditollo@unive.it

***Abstract*—Dieses Dokument ist die Zusammenfassung des Bachelor-Semesterprojekts von Flavio De Jesus Matias unter der Leitung von Giacomo di Tollo.**

1. Einführung & Projektbeschreibung

Web Scraping spielt heutzutage eine große Rolle in der digitalen Geschäftsbranche. Beim Web-Scraping wird ein Crawler verwendet, der als Internet-Bot definiert werden kann, um Informationen von einer externen Website abzurufen. Dieser Bot durchsucht ständig das Internet und sucht anhand eines definierten Themas nach neuen spezifischen Informationen. Mit anderen Worten, beim Web-Scraping werden Daten von einer anderen Website extrahiert, die später für den kommerziellen oder persönlichen Gebrauch analysiert werden.

Ziel des Projekts ist es, ein webbasiertes Scraping-Interface zu erstellen, das mithilfe eines definierten Algorithmus das Auftreten regulärer Ausdrücke auf einer Website abruft, um deren Inhalt zu beschreiben.

2. Voraussetzungen

2.1. Eingabe & Ausgabe

Um richtig zu funktionieren, muss der Algorithmus vom Benutzer eine Eingabe bekommen. Diese Eingabe besteht aus der/den URL(s) und der Bedingung. Ohne diese Eingabe kann der Algorithmus nicht arbeiten, da keine analysierenden Daten vorhanden sind. Zum Schluss muss der Algorithmus die Ergebnisse als Antwort zurück auf das Interface zurückgeben.

2.2. Benutzerfreundliches Interface

Ein wichtiger Teil der Benutzeroberfläche bestand darin, sie so benutzerfreundlich wie möglich zu gestalten. Dies bedeutet, dass der Benutzer sehr wenige bis keine Erklärungen benötigt, um das Konzept und Interface zu verstehen. Die Benutzeroberfläche wurde so einfach und attraktiv wie möglich gehalten, um eine angenehme Benutzererfahrung bei der Nutzung der Website zu ermöglichen.

3. Design & Produktion

3.1. Scientific Deliverable

Der wissenschaftliche Teil des Projekts besteht darin, einen Algorithmus zu entwickeln und zu verwenden, um das Auftreten regulärer Ausdrücke auf einer Website abzurufen. Der Algorithmus wurde in 3 verschiedene Dateien aufgeteilt und organisiert.

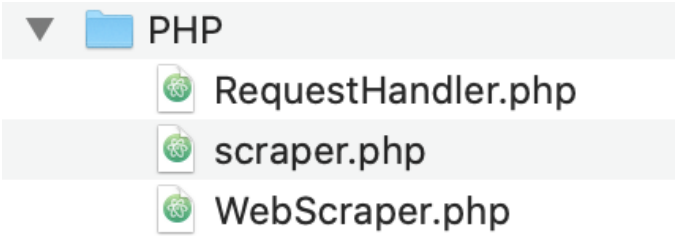

Figure 1: Dateistruktur des Codes mit den 3 Dateien

3.1.1. WebScraper.php. Diese Datei enthält den Algorithmus selbst, der mit der kostenlosen PHP-Bibliothek '*Html2Text*' erstellt wurde. Diese Bibliothek wandelt den Quellcode einer Website in formatierte Daten um, die für die Zwecke des Projekts einfacher zu bearbeiten sind. Mithilfe einer Satzerkennungsfunktion werden die Sätze auf der Website identifiziert und in einer Liste gespeichert. Mit Hilfe einer anderen Funktion wird die vom Benutzer angegebene Bedingung in eine Bedingung geändert, die von PHP ausgewertet werden kann. Diese Bedingung wird dann für jeden der identifizierten Sätze überprüft und die Anzahl der Vorkommen der regulären Ausdrucks werden dadurch ausgerechnet.

3.1.2. RequestHandler.php. Diese Datei bereitet alle Anfragen an die WebScraper-Klasse vor. Es wird die Antwort für das Interface vorbereitet und die erforderlichen Informationen für das nächste Scrape ausgerechnet. Es ist die Middleware zwischen dem Benutzer und dem Scraping-Algorithmus.

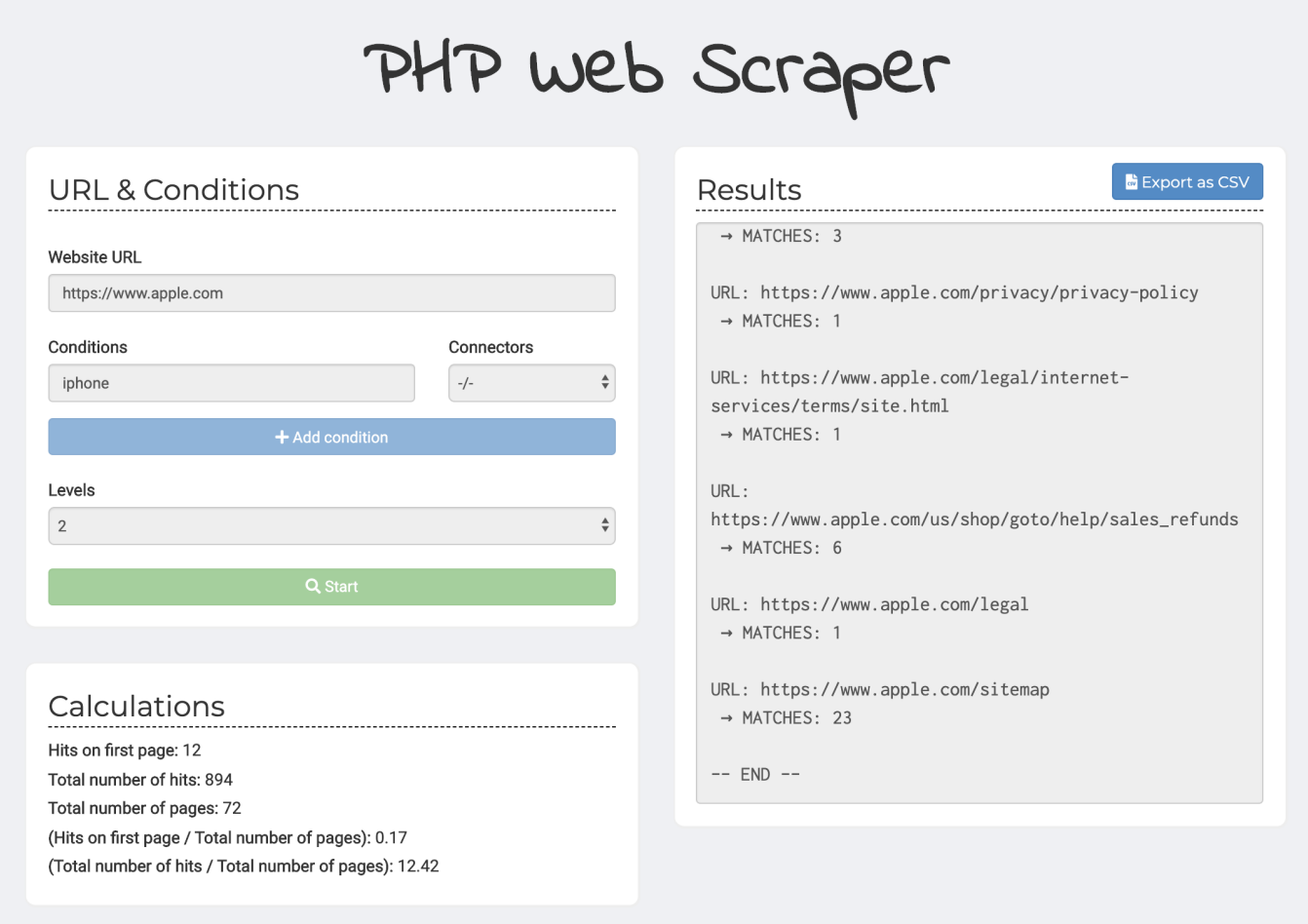

Figure 2: Screenshot des Interfaces nach einem scraping Beispiel

3.1.3. Scraper.php. Diese Datei akzeptiert alle Anfragen vom Interface und sendet die wichtigen Informationen an die RequestHandler-Klasse. Schließlich wird die bereitgestellte Antwort an den Interface zurückgesendet.

3.2. Technical Deliverable

Das technische Teil des Projekts besteht darin, ein Webinterface zu erstellen. Dieses Interface wurde minimal und attraktiv gehalten, um die bestmögliche Benutzererfahrung zu bieten. Die Seite wurde mit Bootstrap 4 erstellt und anschließend in 3 verschiedene Teile unterteilt.

3.2.1. URL & Conditions. Dieses Kasten enthält alle Einstellungen, die der Benutzer ändern kann. Zu Beginn muss der Benutzer zwischen zwei verschiedenen Scraping-Typen wählen: ”URL” und ”Datei”. Je nach gewähltem Typ werden die restlichen Einstellungen angezeigt und der Benutzer kann die gewünschte URL eingeben oder die CSV-Datei hochladen. Der Benutzer kann dann die Levels einstellen und mit dem Scrapen beginnen.

3.2.2. Results. Dieses Kasten ist die ’Konsole’ der Seite. Jede Antwort die der Interface erhält wird in dieses Textfeld geschrieben.

3.2.3. Calculations. Wenn der ausgewählte Typ ”URL” ist, wird dieses Kasten ganz am Ende des Scraping-Vorgangs angezeigt. Es gibt dem Benutzer mehrere Berechnungen über den gesamten Scrape. Wenn der ausgewählte Typ ”Datei” ist, werden diese Daten in die CSV-Exportdatei aufgenommen.

4. Bewertung & Abschluss

Die zu Beginn gestellten Anforderungen wurden erfolgreich erfüllt, da der Benutzer dem Algorithmus die gewünschte Eingabe geben kann sowohl auch die Antwort vom Server zurückerhalten kann. Das Interface wurde ebenfalls mit Erfolg angefertigt und ist sehr benutzerfreundlich, minimal und attraktiv.

In Zukunft könnte das Projekt durch ein maschinelles Lernverfahren verbessert werden, bei dem der Algorithmus seinen Satzerkennungsmechanismus langsam verbessert, um die Scrapingqualität des Endprodukts zu verbessern.