Poznań, 18.01.2020 r.

Tymoteusz Hajder
Jacek Kmiecik
Michał Stancelewski

PROJEKT:

WEB SCRAPER

Podstawy Teleinformatyki

Informatyka, studia niestacjonarne, sem. VII

1. Temat projektu

1.1. Co jest tematem projektu

Tematem niniejszego projektu jest program internetowy działający jako tzw. webscrapper. Jest to narzędzie służące do zbierania danych ze stron internetowych poprzez działanie nazywane “web scrappingiem”.

1.2. Czym jest *web scrapping*

Web scrapping to proces polegający na kolekcjonowaniu danych ze stron WWW. Strona internetowa jest analizowana przez oprogramowanie scrapujące.

Webscrapper pobiera kod HTML strony internetowej. Następnie zaimplementowany w nim parser selekcjonuje pożądane typy danych, a te są zapisywane do przygotowanej dla webscrappera bazy danych. W trakcie tego procesu dane mogą być formatowane i poddawane wstępnej obróbce. Dane znajdujące się w bazie mogą służyć potem innym aplikacjom w dalszej obróbce i analizie.

1.3. Dlaczego ten temat został wybrany

Temat został wybrany ze względu na posiadane przez nasz zespół doświadczenie z technologiami webowymi, takimi jak np. PHP, HTML i Javascript.

Są to technologie, które z jednej strony nadają się do wykorzystania przy tworzeniu aplikacji scrapującej, zaś z drugiej strony - są to również najpopularniejsze języki wykorzystywane w serwisach internetowych, które nasza aplikacja będzie analizować.

2. Podział prac

Tymoteusz Hajder - opracowanie architektury komponentu, wraz z bazową implementacją komend obsługiwanych przez wiersz polecenia.

Jacek Kmiecik - research i implementacja front-endu.

Michał Stancelewski - projekt warstwy backendowej odpowiedzialnej za obsługę kontrolerów i widoków.

3. Funkcjonalność aplikacji

3.1. Pobieranie kodu HTML strony

Aplikacja powinna w pierwszej kolejności uzyskiwać jakiś materiał do analizy. W tym celu program musi łączyć się z siecią WWW poprzez HTTP i pobierać znajdujący się pod wskazanym adresem internetowym dokument HTML.

3.2. Analiza danych z HTML

Aplikacja musi przeprowadzać analizę składniową pobranego kodu HTML. Konieczne jest to, aby rozpoznać fragmenty kodu z drzewa DOM i następnie wyselekcjonować te zawierające wartościowe dane.

3.3. Selekcja istotnych danych

Naszej aplikacji nie są potrzebne wszystkie dane, które zostaną pobrane z plików poprzez HTTP. Taka zawartość jak np. kod CSS czy Javascript nie będą wykorzystywane w dalszym procesie. Aplikacji jest więc potrzebne narzędzie, które wydzieli z całego pobranego kodu tylko te fragmenty, które zamierzamy potem umieścić w naszej bazie danych, a reszta zbędnych informacji zostanie pominięta.

Przykładowe dane z treści plików HTML, które mogą być przydatne z punktu widzenia naszej aplikacji to np.:

- treść nagłówków H1
- div z główną treścią (artykułem) strony
- div z kategoriami artykułu

Odnalezienie poszczególnych elementów można wykonać dzięki analizie danych

(opisanej w punkcie 3.2) i wyszukaniu odpowiednich klas HTML.

3.4. Zapisanie zebranych danych do bazy

Zapisywanie zbieranych informacji odbywa się poprzez uruchomienie polecenia “php bin/cli.php app:pull:all” z poziomu głównego repozytorium. Polecenie to uruchamia skrypt pobierający dane z serwisów, które zdefiniowane są w pliku config/sites.php. Źródła danych dla serwisów muszą być podane jako adresy url do sitemap. Skrypt ten pobiera i parsuje sitemapę, a następnie przekazuje odpowiednie dane do dedykowanego procesora, który przetwarza informację i zapisuje ją docelowo w elasticsearch’u.

3.5. Przeszukiwanie zaindeksowanego zbioru z bazy danych

Treści zapisane przez aplikację w bazie danych można przeglądać. W tym celu aplikacja udostępnia narzędzie wyszukiwarki, która przeszukuję bazę danych. Wyszukiwanie może odbywać się po zadanych słowach kluczowych, bądź kategoriach.

4. Wykorzystywane technologie

4.1. PHP

Przy wyborze podstawowego języka do tworzenia logiki aplikacji wybór padł na język PHP. Jest on popularnym językiem przy projektowaniu aplikacji webowych.

PHP to interpretowany skryptowy język programowania, stosowany głównie do tworzenia stron internetowych i aplikacji webowych. Powstał w roku 1995, a obecnie najnowsza opublikowana wersja to 7.4. PHP wykorzystywany jest najczęściej do tworzenia skryptów działających po stronie serwera.

Język PHP posiada wiele przydatnych bibliotek, które można zastosować przy programach internetowych. I to zarówno dostępne w standardzie języka PHP jak i rozwiązania opensource tworzone przez bardzo liczną społeczność skupioną wokół tego języka programowania. Są to np. takie klasy jak:

- PHP Html Parser - klasa służąca do, jak sama nazwa wskazuje, analizy składniowej kodu HTML. Działa na drzewie DOM, podobnie jak popularne jQuery. Można dzięki niej np. wyszukiwać konkretne elementy w strukturze strony, co będzie przydatne w naszym Webscrapperze.

- SimpleXMLElement - podobnie jak wcześniejsza, ta klasa umożliwia

analizę kodu, tym razem jednak jest to kod XML. Klasa dostępna jest w PHP w standardzie od wersji PHP 5.

- *http://vavatech.pl/technologie/jezyki-programowania/php*
- *https://www.w3schools.com/php/php_ref_simplexml.asp*
- *https://pl.wikipedia.org/wiki/PHP*

4.2. Symfony

Aby ułatwić pracę z językami programowania naturalnym narzędziem są frameworki - czyli platformy programistyczne, często opisywane jako “szkielet do budowy aplikacji”. Dostarczają one zestawy przygotowanych komponentów do budowy programu. Jednym z takich frameworków - który został wykorzystany w naszej aplikacji - jest Symfony.

Symfony jest opensource-owym frameworkiem napisanym w języku PHP. Bazuje na wzorcu MVC. Jego głównym celem jest przyspieszenie tworzenia programu i zmniejszenie ilości potrzebnego do napisania kodu.

Dodatkową zaletą Symfony jest wbudowana obsługa dodatkowych narzędzi, jak np. Twig.

- *https://pl.wikipedia.org/wiki/Symfony*
- *https://hackernoon.com/7-good-reasons-to-use-symfony-framework-for-your-project-265f96dcf759*

4.3. Twig

Twig jest opensource-owym systemem szablonów dla języka PHP. Oparty jest na składni Django. Wyposażony jest m.in. w filtry. Zwiększa również bezpieczeństwo strony (*autoescaping*).

System szablonów to rodzaj biblioteki programistycznej. Można dzięki nim tworzyć dynamiczne strony internetowe. Sam szablon jest plikiem tekstowym ze specjalnymi znacznikami, w które aplikacja wstawia odpowiednie - wcześniej przez nią przetworzone - dane. Zaletą stosowania szablonów jest wyraźne odseparowanie od siebie warstwy prezentacyjnej programu od logiki aplikacji.

- *https://en.wikipedia.org/wiki/Twig_(template_engine)*
- *https://pl.wikipedia.org/wiki/System_szablon%C3%B3w*

4.4. Slim

Slim jest microframeworkiem PHP wykorzystywanym w tworzeniu aplikacji webowych oraz interfejsów API. Slim działa jako dyspozytor. Odbiera żądanie HTTP, a następnie wywołuje odpowiednią procedurę zwrotną i zwraca odpowiedź.

- *http://www.slimframework.com/docs/v4/*

4.5. Composer

Composer jest systemem zarządzania pakietami dla języka PHP. Jes to - inaczej mówiąc - zestaw narzędzi służący do zarządzania pakietami oprogramowania. Composer funkcjonuje jako aplikacja wiersza poleceń i umożliwia m.in. automatyczną instalację i konfigurację dodatkowych komponentów do projektu.

- *https://en.wikipedia.org/wiki/Composer_(software)*

4.6. Elasticsearch

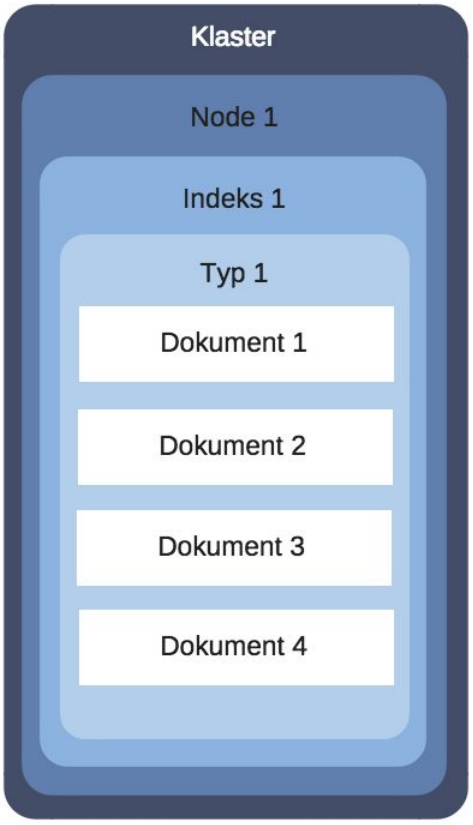
4.6.1. Czym jest Elasticsearch

Elasticsearch to silnik wyszukiwania pełnotekstowego autorstwa firmy Elastic NV. Narzędzie to jest bazą danych wykorzystującą technologię Apache Lucene.

Apache Lucene to opensource-owa biblioteka programistyczna. Oferuje ona funkcje wyszukiwania informacji pozwalające na zbieranie, indeksowanie i wyszukiwanie tekstu. Lucene została napisana w Javie i jej API umożliwia integrację z aplikacjami w tymże języku. Aby rozszerzyć zastosowania Lucene powstają projekty rozbudowujące jej możliwości. Elasticsearch jest właśnie jednym z nich.

Elasticsearch umożliwia kompleksowe przeszukiwanie, filtrowanie i grupowanie danych. Najistotniejszym czynnikiem wpływającym na popularność stosowania Elasticsearch jest jego bogaty zasób bibliotek dla popularnych języków (np. Java, PHP, C#, Ruby, Python), a także dostępne API - REST.

4.6.2. Budowa Elasticsearch



Fundamentalnym elementem Elasticsearch jest **node**, czyli pojedynczy serwer, standardowo operujący na porcie 9200. To w nim dochodzi do przetwarzania danych. Node-y mogą być grupowane w klastry (**cluster**). Wewnątrz node-a znajdują się kolekcje dokumentów o zbliżonej charakterystyce - indeksy (**index**). Na podstawie nazwy indeksu odwołujemy się do konkretnych kolekcji dokumentów. Kolejnym poziomem w budowie Elasticsearch są typy (**type**) - odpowiadają one tabelom z relacyjnych baz danych. Struktura typów jest jednak zależna od tego jakie rekordy są w nich przechowywane. Rekordami zapisywanymi w typach - w formacie JSON - są tzw. dokumenty (**document**).

- *https://en.wikipedia.org/wiki/Elasticsearch*
- *https://geek.justjoin.it/dlaczego-elasticsearch-jest-fajniejszy-od-klasycznych-metod-wyszukiwania-wzorca-w-tekscie/*
- *https://smartbees.pl/blog/elasticsearch-czyli-wszystko-o-wyszukiwaniu-pelnotekstowym*
- *https://czterytygodnie.pl/wprowadzenie-do-elasticsearch/*

4.7. HTML + CSS + JS

Jak wspomniane jest we wcześniejszych podpunktach, *back-end* aplikacji oparty jest na języku PHP. Naturalnym (choć oczywiście nie jedynym) rozwiązaniem dotyczącym *front-endu* jest w tym przypadku zastosowanie standardowych technologii webowych, czyli języków HTML, CSS i Javascript.

HTML - to hipertekstowy język znaczników, wykorzystywany do tworzenia dokumentów tekstowych. Pozwala on opisać strukturę informacji obecnych na stronie internetowej, czyli rozmieszczenie treści.

CSS - czyli kaskadowe arkusze stylów, to język służący do opisywania formy wyświetlania elementów strony internetowej. Pozwala on na dostosowanie wyglądu strony internetowej oraz precyzyjne umiejscowienie poszczególnych elementów z dokumentu HTML.

Javascript - jest, w przeciwieństwie do dwóch poprzednich technologii, językiem programowania. Pisane w nim skrypty wykorzystywane są m.in. do reagowania na różne wydarzenia dziejące się na stronie. Javascript - inaczej niż PHP - jest wykonywany na urządzeniu użytkownika, więc umożliwia dzięki temu wygodniejszą dla użytkującego interakcję ze stroną. Można dzięki temu dokonywać takich działań jak np. walidacja danych, lub komunikację z serwerem (AJAX) bez przeładowywania strony.

- *https://pl.wikipedia.org/wiki/HTML*
- *https://pl.wikipedia.org/wiki/Kaskadowe_arkusze_styl%C3%B3w*
- *https://pl.wikipedia.org/wiki/JavaScript*

4.8. Bootstrap

Bootstrap jest biblioteką CSS. Zawiera od gotowe narzędzia ułatwiające projektowanie układu oraz stylizacji stron internetowych. Bootstrap zbudowany jest z HTML, CSS oraz Javascript.

Głównymi zaletami Bootstrapa są gotowe komponenty do stylizacji elementów HTML, w tym różnych animacji (np. rozwijane menu) oraz gotowe klasy HTML zapewniające responsywność struktury strony.

- *https://pl.wikipedia.org/wiki/Bootstrap_(framework)*

5. Architektura

Aplikacja zbudowana jest z dwóch komponentów:

- **Linii poleceń** - z której można uruchomić pobieranie nowych treści do bazy danych.

- **Serwisu web** - umożliwiającego łatwe przeszukiwanie informacji z silnika baz danych poprzez prosty interfejs.

6. Instrukcja użytkowania aplikacji

Aby skorzystać z wyszukiwarki w platformie należy zacząć od rozpoczęcia pobierania danych do silnika baz danych. Pobieranie danych rozpoczynamy wchodząc w główny katalog repozytorium, uruchamiając komendę “php bin/cli.php

app:pull:all”. W czasie indeksowania danych aplikacja na żywo będzie informować o postępie prac - pokazując kolejno odwiedzane strony.

Po ukończeniu pobierania danych należy wejść na uprzednio skonfigurowany adres w przeglądarce (dedykowany hostname). W naszym testowym, deweloperskim przypadku jest to adres http://webscraper.test/. Po wejściu na ten adres ukaże się strona główna, zawierająca kategorie dostępne do przeszukiwania, wraz z ilością dokumentów powiązanych z danym zagadnieniem. Aby wyszukać dokumenty powiązane z daną kategorią należy wybrać dowolną z nich i kliknąć w przycisk - lupę. Po kliknięciu zostaniemy przekierowani na stronę wyników wyszukiwania - z wybraną kategorią. Wyszukiwanie jest możliwe nie tylko po kategorii, ale także po kategorii i słowie kluczowym. Aby to zrobić należy przejść na stronę główną, wybrać kategorię i wpisać słowo kluczowe.

Dostępne jest również wyszukiwanie informacji w oparciu o samo słowo kluczowe.

Aby okresowo pobierać informacje z serwisów można skorzystać z systemowego **CRONa** - w przypadku systemów z rodziny Linux.

Wszystkie technologie wykorzystane w projekcie są dostępne na platformę Raspberry PI.