Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004
DOI: https://doi.org/10.24867/13BE27Indjic

MIKROSERVIS ZA PRETRAGU REČI U PROJEKTU REČNIKA SRPSKOG JEZIKA

SEARCH MICROSERVICE FOR SERBIAN LANGUAGE DICTIONARY PROJECT

Tanja Inđić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – SOFTVERSKO INŽENJERSTVO

Kratak sadržaj – *U radu je opisan mikroservis za indeksiranje i pretragu tekstualnog sadržaja, karakteristike ElasticSearch alata, programski jezik Python i okruženje Django. Opisani su razvojni okviri neophodni za implementaciju rešenja. Predstavljeno je implementirano rešenje, opisane su funkcionalnosti koje su osnove rada aplikacije i prikazano je dobijeno rešenje.*

Ključne reči: *mikroservis, indeksiranje, pretraga, ElasticSearch, Python programski jezik, Django razvojni okvir*

Abstract – *The thesis describes microservice for indexing and data retrieval, ElasticSearch characteristics, the programming language Python and Django framework. Frameworks and tools needed to for implementation into the solution are described. The implemented solution is presented, the functionalities that are the carriers of the application work are shown and screenshots of developed application are shown.*

Keywords: *microservice, indexing, search, ElasticSearch, Python programming language, Django framework*

1. UVOD

Ovaj rad se bazira na temi razvoja i korišćenja digitalnog rečnika srpskog jezika, uz osvrt na dosadašnje, klasično korišćenje putem knjiga. Kako je čest je slučaj da se određene reči izgovaraju i pišu pogrešno ili one koje se retko koriste padnu u zaborav, a savremen čovek je navikao da mu je sve dostupno na par klikova na računaru, javila se i potreba za kreiranjem *online* srpskog rečnika - rečnika Matice srpske.

Zadatak ovog rada jeste projektovanje *web* aplikacije za indeksiranje i pretragu reči u cilju kreiranja i pretrage korpusa za srpski jezik. Rešenje je potrebno implementirati u *Python* [1] programskom jeziku uz upotrebu *Django* [2] razvojnog okvira a uz oslonac na *ElasticSearch*[3] alat, koji implementira funkcije potrebne za indeksiranje i pretragu dokumenata.

2. TEORIJSKE OSNOVE

U ovom poglavlju se nalazi opis teorijskih osnova na kojima je zasnovan ovaj rad.

[1] https://www.python.org/
[2] https://www.djangoproject.com/
[3] https://www.elastic.co/

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

2.1 Rečnik

Sa razvojem ljudske civilizacije, napredovala je i komunikacija i načini ostavljanja traga i zapisivanja istorije čovečanstva. Nakon crteža javljaju se prva slova, reči i jezici. Tako nastaju i prvi rečnici i dugo vremena su se koristili u svom izvornom, pisanom obliku. Međutim, sa napretkom tehnologije, sve više podataka biva dostupno a i traženo u digitalnom formatu. Iz tog razloga postoji potreba za razvojem *online* srpskog rečnika koji bi bio dostupniji i lakše pretraživ.

2.1.1 Razvoj računara i povećanje obima podataka

Vremenom računari postaju sve dostupniji, njihova primena je sve raznovrsnija, a tehnološkim napretkom brzina procesiranja podataka i količina sa kojom mogu da barataju raste. Samim tim, rano se javlja potreba za njihovom organizacijom, klasifikacijom i efikasnom pretragom, pojavljuju se brojni algoritmi za smeštanje i brzo pronalaženje podataka [1]. Uz to, kako se spektar krajnjih korisnika širi, sve više se radi na razvoju korisničkih interfejsa koji bi pojednostavili i ubrzali rad na računarima. Uporedo sa razvojem računara, radi se i na razvoju infrastrukture koja bi povezala više računara putem telekomunikacija. Rezultat ovih napora bio je ARPANET [2], prva računarska mreža koja je trebala da obezbedi komunikaciju vojnih laboratorija, vladinih biroa i univerziteta. Originalni ARPANET je vremenom prerastao u Internet, što je dovelo do još većeg rasta količine dostupnih podataka, međutim da bi iko iskoristio bilo koji podatak potrebno je prvo da može da ga pronađe.

2.1.2 Alati za pretragu

Jedan od prvih sistema za pretragu bio je *Archie* [3], koji je služio kao alat za indeksiranje FTP (*File Transfer Protocol*) fajlova. U odnosu na današnje pretraživače, može se reći da je *Archie* bio primitivan alat, budući da nije podržavao proizvoljne korisničke upite, niti je indeksirao sadržaje fajlova, te je pretraga bila ograničena samo na tačne nazive. Indeksiranje sadržaja prvi put je podržao *Gopher* [3] protokol, a ideja iza ovog sistema bila je laka upotreba, hijerarhijski prikaz i kretanje kroz sistem, jednostavna sintaksa i podrška za pretrage. Bio je zamišljen kao globalni fajl sistem prikazan kao hijerarhija hiperlinkova, međutim već od 1993. godine umesto njega počinje da se koristi *World Wide Web* [2] – *web*.

Kako se funkcionalnost pretrage i indeksiranja tražila i u manjim i jednostavnijim okruženjima, dolazi do razvoja *Lucene* [5] biblioteke a zatim i *ElasticSearch* alata koji postaje globalno korišćen kao dodatak u širokom spektru sistema kojima je potrebna ovakva funkcionalnost. Omogućava samostalno korišćenje a do skoro i kao *embedded* deo servisa.

3. ALATI ZA RAZVOJ

ElasticSearch je jedan od najpoznatijih distribuiranih alata za pretragu nastao 2010. godine, a njegov autor je Shay Banon [4]. Zasniva se na *Lucene* biblioteci, napisan je u *Java* programskom jeziku i pruža mogućnost rada sa tekstualnim, numeričkim, geospacijalnim, strukturiranim i nestrukturiranim podacima [5].

Podrazumeva rad sa dokumentima kao osnovnom vrstom organizacije podataka koji se reprezentuju u JSON formatu. Podržava HTTP veb interfejs putem kojeg je moguće vršiti pretragu, dodavanje i manipulaciju nad dokumentima. Pre samog indeksiranja, potrebno je definisati model dokumenta i način na koji se njegova polja čuvaju i indeksiranju.

Neretko se dešava da podaci sa kojima se radi budu izvan trenutnog sistema, te se ne može uticati na njihov format. Zbog toga je postupak transformacije podataka u oblik koji je pogodan za indeksiranje i pretragu neophodan i ovaj korak je ključan za dobar rad sistema a enkapsuliran je u obliku dodatka – *plugin*. Ovakav dodatak se naziva *analyzer* i sastoji se iz filtera karaktera, tokenizera i filtera tokena [5].

Filtriranje karaktera podrazumeva uklanjanje, dodavanje i izmenu pojedinačnih karaktera u dokumentu, koristi se pri parsiranju HTML fajlova za uklanjanje oznaka za elemente, transformaciju ili brisanje određenih karaktera, uklanjanje akcenata, prebacivanje iz jednog pisma u drugo.

Tokenizer preuzima transformisani niz karaktera i služi za razdvajanje podataka u tokene, na primer tekst može razdvojiti na niz reči na osnovu razmaka, znakova interpukcije i slično. Takođe, zadužen je za pamćenje redosleda ili pozicije tokena, kao i početak i kraj originalne reči u nizu karaktera.

Filter tokena služi za krajnju transformaciju nakon razdvajanja teksta na niz tokena. Može prebaciti sve elemente ovog niza u mala slova ili procesirati podatke na osnovu odabranog jezika. Moguće je definisati listu stop reči za određeni jezik, tj. listu često korišćenih reči koje nije potrebno indeksirati i treba izbaciti iz daljeg procesiranja.

3.1 *Python* programski jezik

Python je interpretirani, objektno orijentisani programski jezik visokog nivoa nastao 1991. godine. Podržava imperativni, objektno-orijentisani i funkcionalni stil programiranja. Svaki podatak predstavljen je kao objekat ili kao relacija između objekata. Budući da koristi strukturu podataka visokog nivoa, u kombinaciji sa dinamičkim kucanjem i dinamičkim vezivanjem, *Python* se često koristi za brz razvoj aplikacija. Ovaj programski jezik podržava dodatne module i pakete, omogućava modularnost i ponovnu iskorišćenost koda. Moguće ga je interpretirati i kompajlirati.

3.2 *Django* razvojni okvir

Django je razvojni okvir koji omogućava lako razvijanje naprednih i dinamičnih web aplikacija uz korišćenje *Python*-a. Ono što izdvaja ovu platformu je podrška za autorizaciju i autentifikaciju korisnika koja se lako može prilagoditi svakom tipu sistema. Veoma je fleksibilan, omogućava jednostavan i intuitivan rad sa bazama podataka.

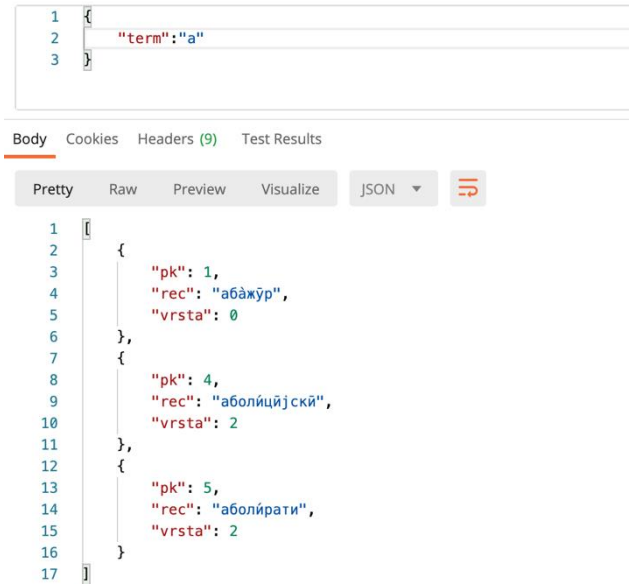
Podržava MTV[4] arhitekturu (*model-template-views*) koja omogućava deljenje implementacije u više slojeva. Ukoliko posmatramo kako funkcioniše srž ovog alata, može se reći da zapravo podržava standardnu MVC[5] arhitekturu (*model-view-controller*): koristi svoj maper za translaciju između definisanih modela podataka i relacionih modela, zatim ima svoj sistem koji podržava korišćenje veb šablona za kreiranje prikaza i takođe sadrži deo za komunikaciju sa klijentima koji je baziran na putanjama – URL (*Uniform resource locator*).

4. IMPLEMENTACIJA REŠENJA

Projekat za Rečnik srpskog jezika je osmišljen kao podrška za razvoj *online* rečnika Matice srpske. Ovaj projekat se razvija na Katedri za informatiku Fakulteta tehničkih nauka u Novom Sadu. Poput drugih rečnika, ima namenu da omogući krajnjim korisnicima pretragu reči i na ćirilici i latinici, ali je namenjen i za leksikografe koji kreiraju sadržaj samog rečnika. Projektovan je po ugledu na mikroservisnu arhitekturu i pored ovog servisa postoje još i servis za digitalizaciju dokumenata, servis za ekstrakciju sadržaja, klijentska aplikacija za unos i pretragu rečničkih odrednica i servis za pripremu rečnika za štampu.

Programski jezik korišćen za implementaciju aplikacije za indeksiranje i pretragu je *Python*, a razvojno okruženje je *PyCharm*. Rešenje je implementirano korišćenjem *Django* razvojnog okvira uz oslonac na *ElasticSearch* kao alat za indeksiranje i pretragu. Komunikacija sa u okviru sistema se odigrava preko REST servisa, a *Django* razvojni okvir se koristi za mapiranje JSON opisa na odgovarajuće klase.

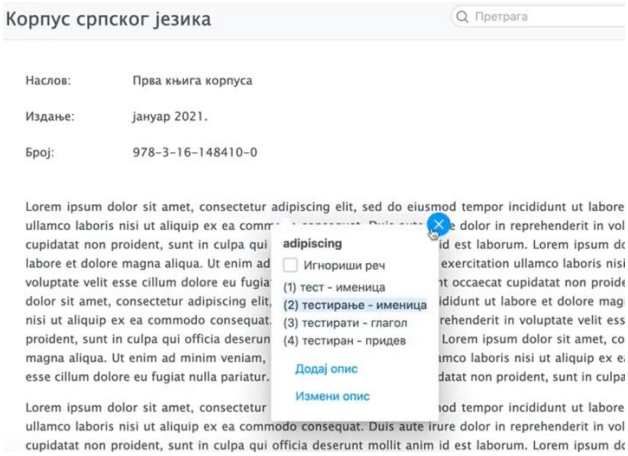
Za pretragu rečničkih odrednica bilo je potrebno implementirati *auto-complete* funkcionalnost i to uz podršku za oba pisma. Od podataka koji se unose pri kreiranju odrednice izdvojeni su primarni ključ, reč i vrsta reči, na osnovu kojih se mogu iz baze izvući dodatne informacije.



Slika 1. *Prikaz rezultata pretrage odrednica*

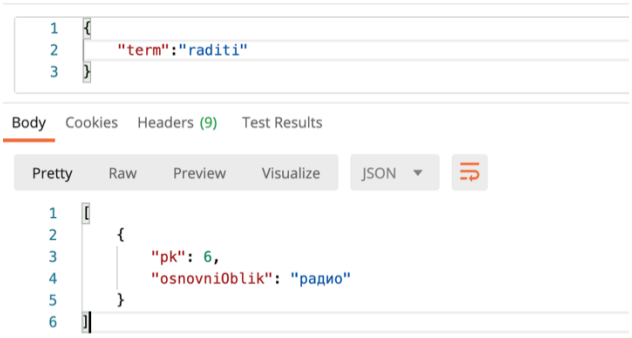
[4] https://djangobook.com/mdj2-django-structure/
[5] https://en.wikipedia.org/wiki/Model-view-controller/

Druga funkcionalnost predstavlja pretragu već postojećih anotiranih reči. Naime, leksikografi u procesu anotiranja dokumenta prolaze kroz njegov sadržaj i potrebno je omogućiti opciju da se svaka reč iz sadržaja definiše kao nova anotirana reč ili da se poveže sa postojećom. Na taj način dobija se anotirani korpus srpskog jezika.



Slika 2. *Prikaz forme za anotaciju reči*

Za svaku anotiranu reč bilo je potrebno indeksirati sve njene oblike i takođe podržati pretragu i na ćirilici i na latinici.



Slika 3. *Prikaz rezultata pretrage za anotiranu reč*

5. ZAKLJUČAK

U radu je opisan servis za indeksiranje i pretragu dokumenata, opisan je programski jezik *Python* i njegov razvojni okvir *Django*. Predstavljen je *ElasticSearch* kao alat na koji se ovaj servis oslanja. Prezentovano je rešenje, objašnjene najznačajnije funkcionalnosti i dat je primer korišćenja aplikacije. Kreirano rešenje može biti korisno za Rečnik srpskog jezika, ali i za druge primene, uz određene modifikacije. Takođe, korišćenje *Django* razvojnog okvira, donelo je velike prednosti i olakšanja u implementaciji sistema.

Rad u ovoj oblasti ukazao je na veliki broj mesta i prilika za proširenje. Kreiranje *web* servisa, autentifikacija i autorizacija korisnika mogu biti odrađeni na više načina, ne samo upotrebom *Django*-a. Takođe, indeksiranje i pretraga ne moraju nužno biti izvršeni korišćenjem *ElasticSearch* alata, ali same transformacije podataka i upita pomoću posebno napravljenog dodatka bi morale biti drugačije implementirane kako bi se podržalo i latinično i ćirilično pismo.

6. LITERATURA

[1] Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable and Maintainable Systems, 2017, ISBN 9781491903070
[2] Internet, ARPANET. Preuzeto sa https://en.wikipedia.org/wiki/Internet
[3] Search Engine. Preuzeto sa https://en.wikipedia.org/wiki/Search_engine
[4] ElasticSearch. Preuzeto sa https://en.wikipedia.org/wiki/Elasticsearch
[5] ElasticSearch dokumentacija. Preuzeto sa https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

Kratka biografija:

Tanja Inđić rođena je 1995. godine u Aranđelovcu. Završila je gimnaziju „Miloš Savković“ u Aranđelovcu. 2014. godine, upisuje se na studije na Fakultetu tehničkih nauka, odsek Računarstvo i automatika. Za dalje usmerenje u okviru pomenutog odseka, odlučuje se za smer Primenjene računarske nauke i informatika. Master studije je slušala na istom odseku i smeru, a za podsmer je odabrala Elektronsko poslovanje.