

Upravljanje digitalnim dokumentima (UDD)

Pretraga CV dokumenta i priloženih pisama u okviru IT firme

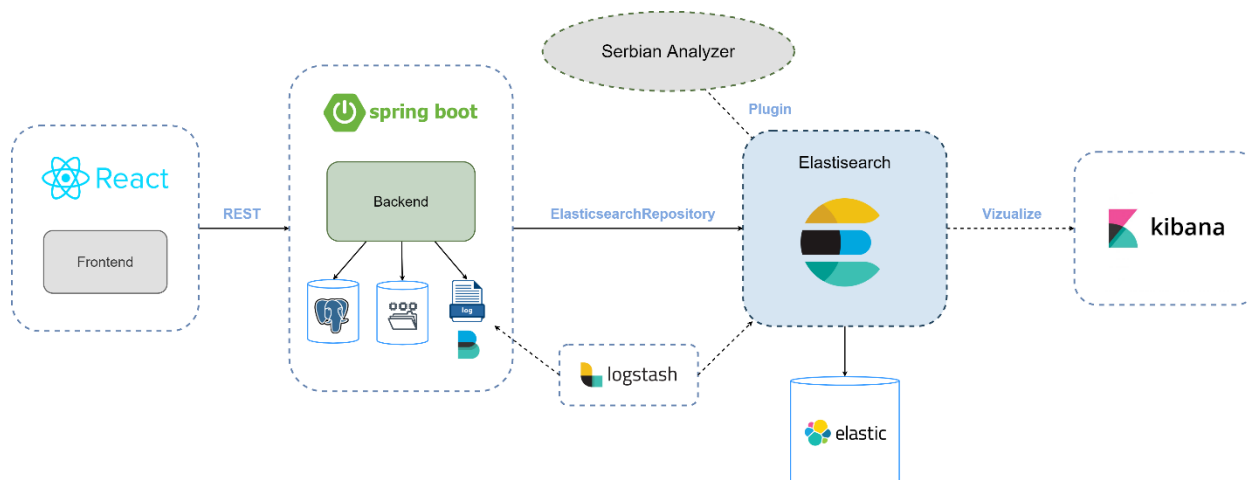
Opis sistema i traženih funkcionalnosti

U okviru sistema IT firme za potrebe predmeta upravljanje digitalnim dokumentima neophodno je obezbediti adekvatnu formu putem koje bi kandidati mogli da se prijave za određeni oglas za posao unošenjem svojih osnovnih podataka i ostavljanjem CV dokumenta (pdf). Takođe je potrebno kreirati formu za zadavanje upita kako bi zaposleni u HR službi na efikasan način mogli da vrše pretragu kolekcije dostavljenih CV dokumenata i priloženih pisama.

Prethodno opisanu pretragu potrebno je implementirati uz oslonac na Elasticsearch platformu, gde je prilikom prijave kandidata neophodno izvršiti indeksiranje odgovarajućih podataka kako bi se omogućila efikasna pretraga po imenu, prezimenu i stepenu stručne spremljanosti, kao i prema sadržaju CV dokumenta. Potrebno je omogućiti i kombinaciju ovih parametara upotrebom AND i OR operatora i obezbediti podršku za zadavanje fraza u upitima. Upite je potrebno preprocesirati pomoću SerbianAnalyzer-a, dok je prilikom prikaza rezultata potrebno kreirati dinamički sažetak (Highlighter). Još jedan od zahteva jeste pretraga po geolokaciji gde je potrebno uneti ime grada i radijus u okviru kojeg će se vršiti pretraga aplikacija. Takođe se zahteva korišćenje ELK Stack-a za dobijanje statistika iz kojih gradova i u koje vreme su aplikanti najviše pristupali formi.

Arhitektura sistema i skladištenje podataka

Arhitektura celokupnog sistema prikazana je na slici 1. Za implementaciju frontend dela aplikacije koji predstavlja korisnički sloj koristiće se React biblioteka zasnovana na JavaScript-u, dok će backend deo koji predstavlja serverski sloj biti realizovan kao višeslojna Java Spring Boot aplikacija. Na slici 1 je takođe prikazano na koji način su Elasticsearch i ostale komponente povezane u okviru ovog sistema kao i način na koji one međusobno komuniciraju, o čemu će biti više reči u nastavku. Podaci koji se odnose na pristigle prijave za posao, putanju do CV dokumenata (pdf) i ostali relevantni entiteti koji su neophodni za funkcionisanje sistema čuvaće se u okviru PostgreSQL relacione baze podataka, dok će se za podatke kao što su ime, prezime, stepen obrazovanja aplikanta i naziv CV dokumenta koji su neophodni za efikasnu pretragu vršiti indeksiranje korišćenjem ElasticsearchRepository-ja koji zahteva definisanje index unita-a i njegovog ključa i ti podaci će se čuvati u okviru Elasticsearch-a o čemu će biti više priče u nastavku. Skladištenje kolekcije CV dokumenti (pdf) će se vršiti u okviru fajl sistema.



Slika 1. Arhitektura sistema

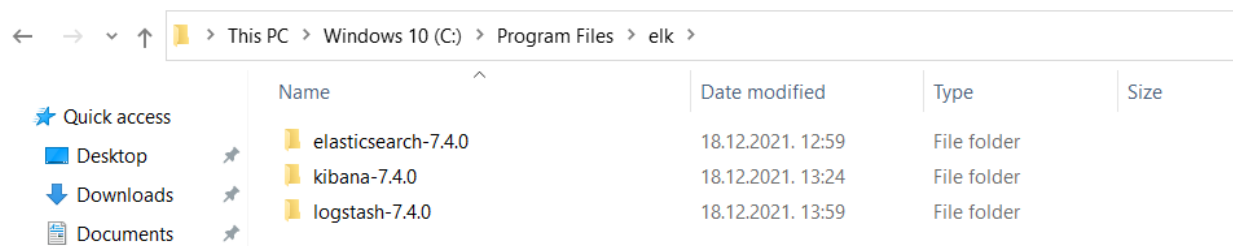
Komunikacija između komponenti sistema

Klijentska aplikacija će komunicirati sa serverom pozivanjem odgovarajućih REST endpoint-a. Obzirom da Elasticsearch takođe nudi REST API-je, mogla bi se uspostaviti REST komunikacija između Elasticsearch-a i backend dela sistema, gde bi se pozivom odgovarajućih API-ja pravili novi dokumenti i vršili upiti nad njima. U okviru ovog sistema, komunikacija sa Elasticsearch-om će se vršiti korišćenjem SpringData Elasticsearch-a čime se olakšava indeksiranje, pisanje upita i CRUD operacija. Za pisanje upita će se koristiti ElasticsearchTemplate kako bi se na efikasniji način mogao definisati dinamički sažetak (Highlighter) i kako bi se na efikasniji način manipulisalo pretragom.

Konfiguracija Elasticsearch-a

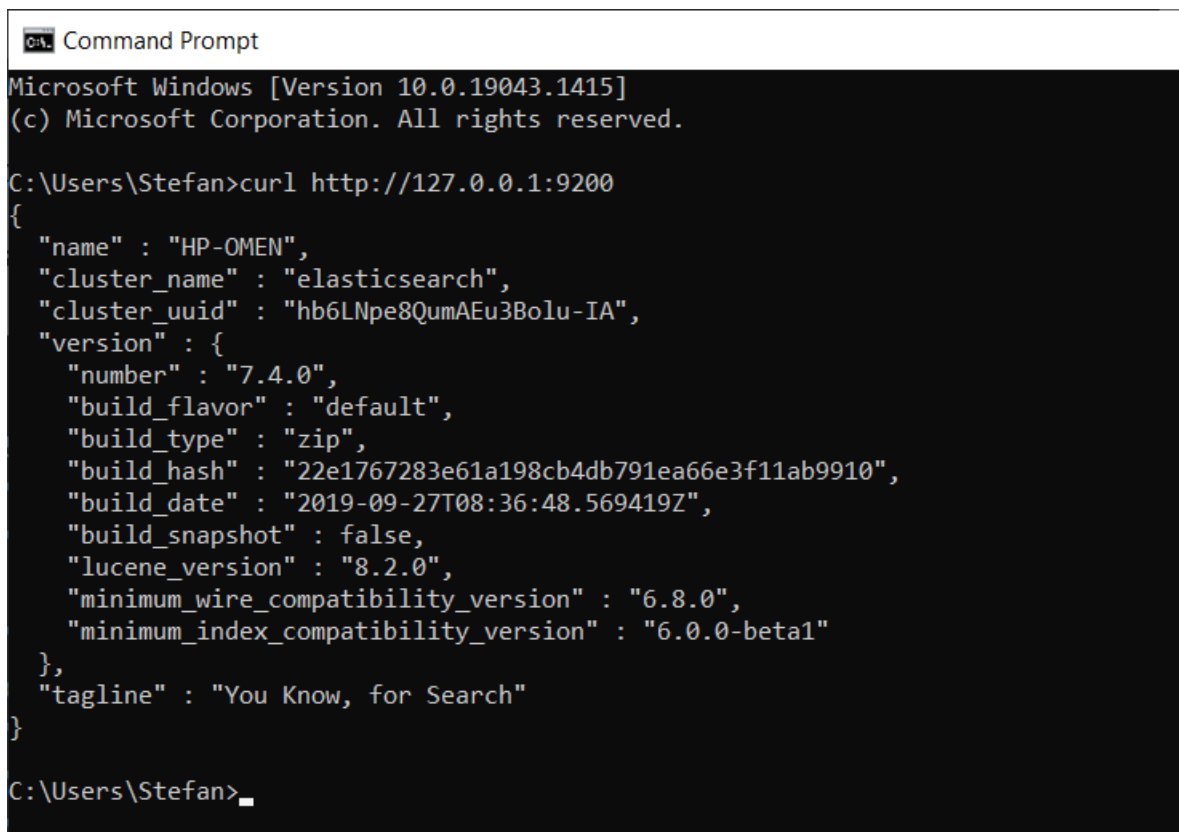
S obzirom da se u zahtevu aplikacije pored upotrebe Elasticsearch-a za indeksiranje i pretragu dokumenata zahteva i upotreba ELK stack-a koji podrazumeva korišćenje Kibane i Logstash-a za praćenje statističkih podataka, u ovom odeljku će biti opisani i njihova konfiguracija.

Za potrebe ovog projekta sa sajta <https://www.elastic.co/downloads> su preuzeti Elasticsearch, Kibana i Logstash verzije 7.4.0 zbog kasnije integracije sa SerbianAnalyzer plugin-om. Nakon što su preuzeti zip fajlovi, raspakovani su u okviru C fajl sistema na računaru (Slika 2).



Slika 2. Raspakovani folderi Elasticsearch, Kibana, Logstash

Kako bi se pokrenuo Elasticsearch na lokalnoj mašini, neophodno je otvoriti command prompt kao administrator, pozicionirati se u korenski direktorijum raspakovanog fajla, nakon čega je potrebno izvršiti komandu `bin\elasticsearch.bat` čime će se izvršiti njegovo pokretanje. Nakon uspešnog pokretanja Elasticsearch će biti pokrenut na lokalnoj mašini na portu 9200. Na slici 3 prikazan je rezultat uspešnog pokretanja.



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

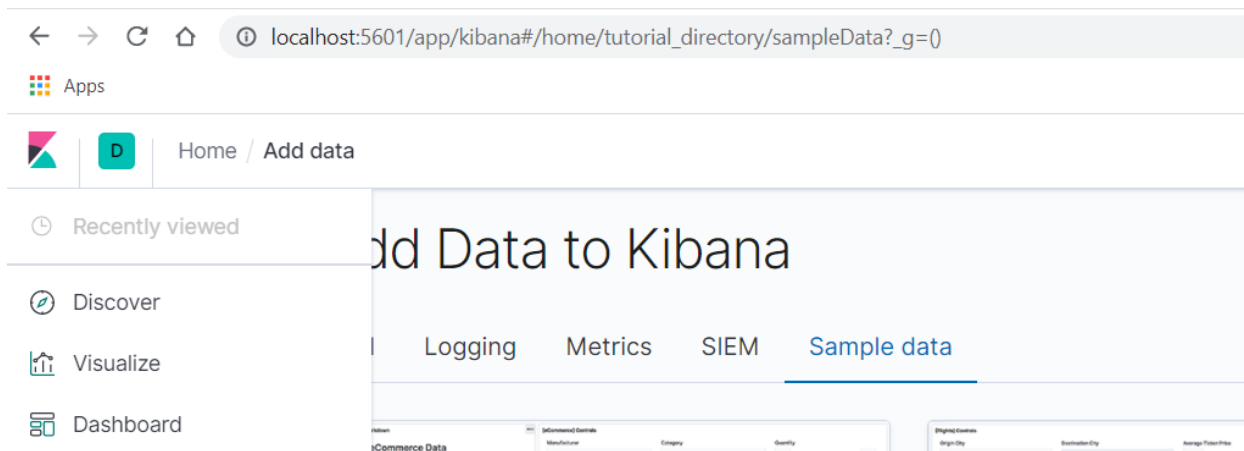
C:\Users\Stefan>curl http://127.0.0.1:9200
{
  "name" : "HP-OMEN",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "hb6LNpe8QumAEu3Bolu-IA",
  "version" : {
    "number" : "7.4.0",
    "build_flavor" : "default",
    "build_type" : "zip",
    "build_hash" : "22e1767283e61a198cb4db791ea66e3f11ab9910",
    "build_date" : "2019-09-27T08:36:48.569419Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}

C:\Users\Stefan>
```

Slika 3. Rezultat uspešnog pokretanja Elasticsearch-a na lokalnoj mašini

Za upotrebu Elasticsearch-a u okviru Spring Boot aplikacije neophodno je dodati dependency `spring-boot-starter-data-elasticsearch` u okviru kojeg postoje podrazumevana podešavanja za uspostavljanje komunikacije sa Elasticsearch-om. Ova podešavanja se mogu menjati putem konfiguracione klase, kao i izmenom samo `elasticsearch.yml` fajla gde je moguće menjati podatke o klasterima, čvorovima, putanjama gde se čuvaju podaci i logovi, kao i port na kom će Elasticsearch biti pokrenut. Za potrebe ovog projekta neće se vršiti dodatna podešavanja.

Nakon uspešnog pokretanja Elasticsearch-a, prateći iste korake moguće je izvršiti pokretanje Kibane nakon što se izvrši komanda `bin\kibana.bat` u okviru korenskog direktorijuma. Nakon uspešnog pokretanja Kibana će biti startovana na portu 5601 na lokalnom računaru. Rezultat pokretanja Kibane i korisničkog interfejsa prikazan je na slici 4.



Slika 4. Korisnički interfejs Kibane nakon pokretanja

Za pokretanje Logstash-a najpre je potrebno kreirati logstash.conf fajl u okviru korenskog direktorijuma. U ovom fajlu je neophodno izvršiti određena podešavanja kako bi se Logstash povezo sa Elasticsearch-om i fajlom koji sadrži logove koji su izgenerisani od strane Spring Boot aplikacije. Izvršavanjem komande `bin\logstash -f logstash.conf` izvršiće se njegovo pokretanje, nakon čega će statističkim podacima o logovima moći da se pristupi kroz korisnički interfejs Kibane.

Konfiguracija SerbianAnalyzer-a

Kako bi se izvršilo preprocesiranje upita upotrebom SerbianAnalyzer-a, potrebno je ispratiti nekoliko koraka za njegovu konfiguraciju i spajanje sa Elasticsearch-om. Plugin se nalazi na sledećem linku <https://github.com/chenejac/udd06> i prilagođen je za Elasticsearch 7.4.0. Najpre je potrebno pomoću Gradle-a buildovati zip fajl koji predstavlja plugin pod nazivom *serbian-analyzer-1.0-SNAPSHOT.zip* koji se potom ubacuje u Elasticsearch. Pozicioniranjem u bin foldera u sam Elasticsearch potrebno je izvršiti sledeću komandu: `elasticsearch-plugin install file:<putanja do zipovanog fajla>`. Izmenom `elasticsearch.yml` fajla moguće je podesiti da SerbianAnalyzer bude default-ni, međutim u okviru ovog projekta će se za svako polje u okviru indeksne strukture postaviti da se koristi SerbianAnalyzer (Slika 5), takođe će se tokom postavljanja upita setovati korišćenje SerbianAnalyzera prilikom korišćenja QueryBuildera.

```

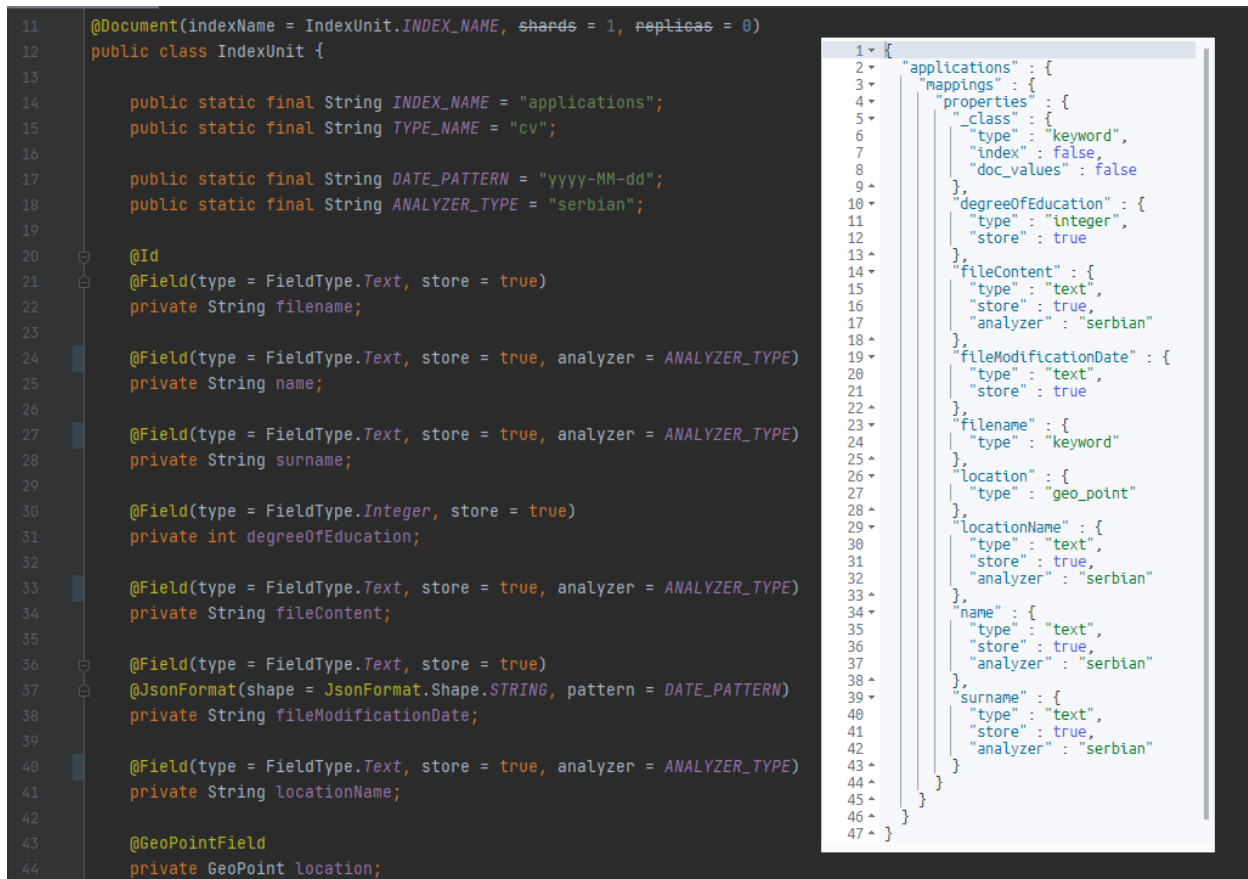
18     public static final String ANALYZER_TYPE = "serbian";
19
20     @Id
21     @Field(type = FieldType.Text, index = false, store = true)
22     private String filename;
23
24     @Field(type = FieldType.Text, store = true, analyzer = ANALYZER_TYPE)
25     private String name;

```

Slika 5. Podešavanje SerbianAnalyzer-a za polja

Indexing Unit (JSON)

Kako bi se vršila pretraga aplikacija i dostavljenih CV dokumenata, najpre ih je neophodno indeksirati u okviru Elasticsearch-a. Spring Boot aplikacija nudi anotaciju *Document* kojom se može opisati određeni dokument za koji se vrši indeksiranje, dok se atributi mogu predstaviti anotacijom *Field*. Na slici 6 je prikazan izgled *IndexUnit* klase u okviru ovog projekta, gde sam naziv fajla predstavlja id dokumenta, dok se parametri kao što su ime, prezime, stepen obrazovanja i sadržaj pdf dokumenta predstavljeni kao polja koja će se koristiti za pretragu. Takođe se nalaze dodatna polja vezana za geoprostornu pretragu. Za sve parametre osim stepena obrazovanja definisan je SerbianAnalyzer, s obzirom da stepen obrazovanja predstavlja celobrojnu vrednost. Desni deo slike predstavlja JSON objekat koji predstavlja strukturu definisanog indexa u okviru Elasticsearch-a, tačnije pozivom GET zahteva u okviru Kibane može se dobiti struktura indeksa definisanog u okviru SpringBoot aplikacije.



Slika 6. IndexUnit klasa i struktura definisanog indeksa

Geoprostorna pretraga

Za potrebe geoprostorne pretrage mogu se vršiti *Geo-distance* upiti koji su dostupni u okviru Elasticsearch-a <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-geo-distance-query.html>. Jedan od načina za pozivanje ovih upita kroz Spring Boot aplikaciju jeste korišćenje *QueryBuilders* klase nad kojom se može pozvati metoda *geoDistanceQuery*, čime se dobija objekat klase *GeoDistanceQueryBuilder*. Nad ovim objektom se pozivom metode *point* može definisati tačka u odnosu na koju se gleda udaljenost, što se može predstaviti objektom tipa *GeoPoint* kome se prosleđuju geografska širina (lat) i geografska dužina (lon). Pozivom metode *distance* nad istim objektom se definiše geografska udaljenost za koju se pretraga vrši. Nakon definisanog grada i željenog radijusa, dobavljanjem indeksnih jedinica, pronaći će se aplikanti koji zadovoljavaju unete parametre.