Upravljanje digitalnim dokumentima

Nevena Atić E2 115-2022

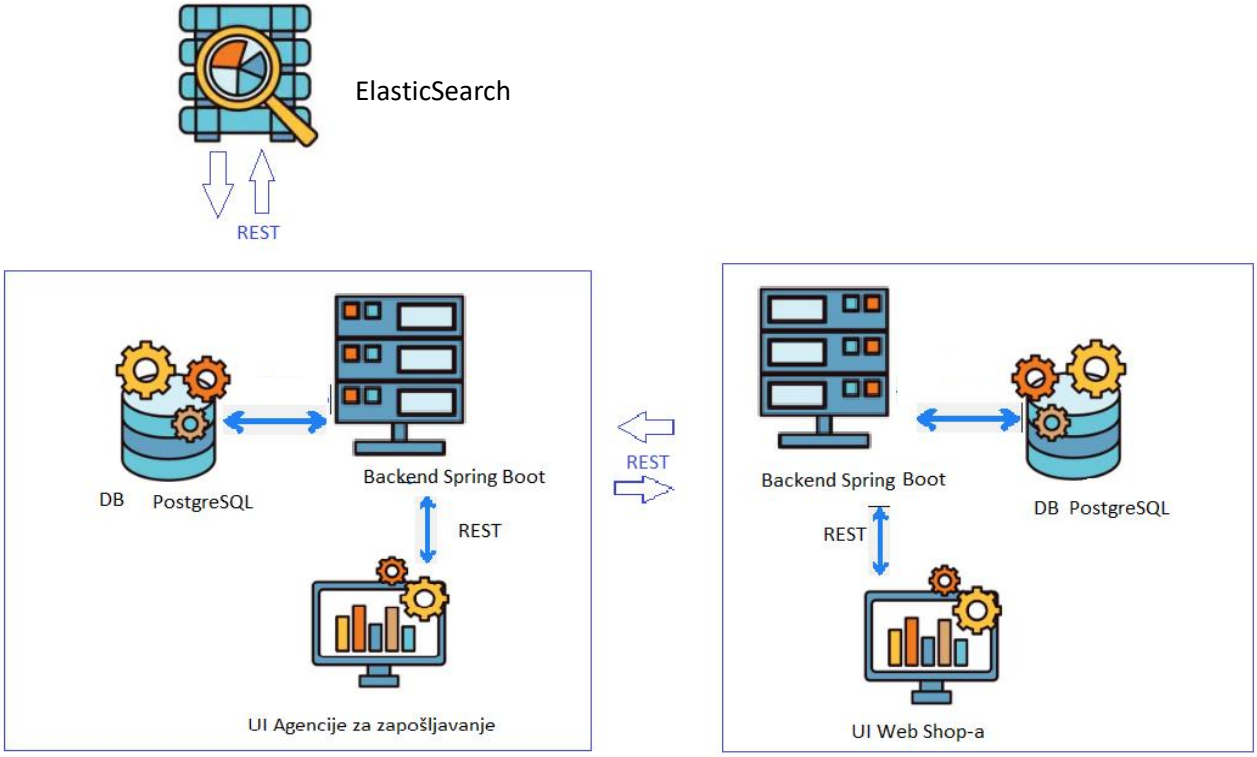
Prva kontrolna tačka

1. Uvod

Agencija za zapošljavanje predstavlja aplikaciju koja omogućava jednostavnu registraciju klijenata na sistem u potrazi za poslom, kao i mogućnosti pretplate i različite vrste plaćanja članarine u Web Shop prodavnici (PayPal, Bitcoin, karticom i QR kod). Kreiranje zahteva za zapošljavanje svakog korisnika započinje se popunjavanjem forme u kojoj se, pored osnovnih ličnih podataka, dostavlja CV i propratno pismo u PDF formatu, pa samim tim ova aplikacija skladišti veliki broj fajlova koje treba sačuvati, obraditi i aplikantu predložiti adekvatna potencijalna mesta za zaposlenje na osnovu njihove pretrage. U tu svrhu koristi se ElasticSearch, koji omogućava pretraživanje kolekcije fajlova na osnovu odgovarajućih upita.

U daljem teksu biće opisana arhitektura aplikacije i konfiguracija ElasticSearch-a, pretprocesiranje tekstova na srpskom jeziku (SerbianAnalyzer), indexing unit i geoprostorna pretraga.

2. Arhitektura aplikacije



Slika 1 – Prikaz arhitekture sistema

Dve glavne celine sistema jesu Agencija za zapošljavanje i Web Shop prodavnica, preko koje se vrši plaćanje. Obe su slojevite arhitekture, a slojevi su:

- Model sloj
- DTO (DataTransferObject za dostavljanje adekvatnih informacija u komunikaciji između Spring Boot i Angular aplikacija),
- Repository sloj za pristup informacijama iz baze podataka,
- Service sloj gde je implementirana logika aplikacije
- Controller sloj u kome se nalaze API-ji.

Backend i jedne i druge aplikacije takođe je realizovan kao Spring Boot aplikacija, a frontend kao Angular. Za skladištenje podataka koristi se PosgreSQL relaciona baza podataka. Komunikacija izmedju frontend i backend aplikacija ostvaruje se slanjem REST zahteva, dok se bazama podataka pristupa isključivo putem backend aplikacije, pozivanjem odgovarajućih servisa.

Međusobna komunikacija između aplikacija takođe se odvija putem REST-a, a na isti način se Agencija za zapošljavanje obraća ElasticSearch-u. Da bi komunikacija između Agencije i ElasticSearch-a bila moguća, potrebno je ubaciti odgovarajuće dependency-je unutar Spring Boot aplikacije za Agenciju, kao i napraviti klijenta u konfiguracionom fajlu koji omogućava slanje REST zahteva na portu 9200 na kom sluša ElasticSearch ukoliko mu se na taj način pristupa.

3. Skladištenje podataka

Podaci koji se čuvaju u okviru Agencije za zapošljavanje su:

- Korisnici sistema koje čine kandidati, kompanije, zaposleni i administrator sistema
- Informacije o zahtevima za zaposlenje
- Transakcije koje se tiču plaćanja članarine korišćenja sistema
- Pretplate

CV-jevi I propratna pisma skladištiće se u okviru ElasticSearch-a, kako bi pretraga podataka bila omogućena. Kako bi se ostvarila I geoprostorna pretraga, potrebno je na ElasticSearch platformi čuvati informacije o geografskoj širini I dužini kompanija.

4. ELK Stack

ELK Stack sastoji se od tri opensource celine kojoj je kasnije dodata i četvrta:

- **ElasticSearch** – pretraga i analiza otvorenog koda, zasnovan na Apache Lucene pretraživaču
- **Logstash** – prikuplja logove iz različitih izvora, izvršava različite transformacije i poboljšanja, pa šalje podatke na različita odredišta
- **Kibana** – sloj vizuelizacije koji radi na vrhu ElasticSearch-a, pružajući korisnicima mogućnost analize i vizuelizacije podataka
- Beats - lagani agenti koji su instalirani na ivičnim hostovima da prikupljaju različite tipove podataka za prosleđivanje u stek

ELK Stack pomaže tako što korisnicima pruža moćnu platformu koja prikuplja i obrađuje podatke iz više izvora podataka, skladišti te podatke u jednom centralizovanom skladištu podataka koje može da se povećava kako podaci rastu i koja obezbeđuje skup alata za analizu podataka. Izvor svih informacija prikupljen na linku.

5. ElasticSearch

ElasticSearch je open source server napisan u Java programskom jeziku, koji služi za pretragu i analizu podataka u realnom vremenu. U okviru ovog projekta će se koristiti ElasticSearch verzija 7.4.0 i potrebno ga je instalirati na računaru. Kako bi Agencija za zapošljavanje (SpringBoot aplikacija) komunicirala sa ElasticSearch-om putem REST-a, potrebno je u dodati dependency prikazan na Slici 2., kao i port na kom će se aplikacija pokretati u aplication.properties fajlu ili nekom konfiguracionom fajlu, poput elasticsearch.yml, naziv baze, ElasticSearch klastera, host i port ElasticSearch servisa na kom se nalazi.

```
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-elasticsearch</artifactId>
    <version>3.0.8.RELEASE</version>
</dependency>
```

Slika 2 – Dependancy pomoću kog Spring Boot aplikacija komunicira sa ElasticSearch-om

Spring Data ElasticSearch koristi ElasticSearchRepository interfejs kojim su omogućene i CRUD operacije, slično kao sa JPA repozitorijumima. Pored ovog interfejsa, koristiće se i ElasticSearchTemplate, koji isto omogućava indeksiranje i pretraživanje. Mapiranje dokumenata na java klase vrši se upotrebom *@Document* anotacije, *@Id* predstavlja filed_id naseg dokumenta i jedinstvene je vrednosti, a *@Filed* predstavlja razlicit tip polja.

6. Logstash

Logstash prikuplja logove unutar ElasticSearch-a, a koristi se i za otpremanje i skladištenje logova. Pokreće se onog trenutka kada se događaj upiše u fajl. Sastoji se od tri komponente – ulaz, filteri i izlaz. Ulaz predstavljaju prosleđeni logovi namenjeni za obradu. Filtriranje se odvija prema setu prethodno definisanih pravila, dakle to je skup uslova za obavljanje određene akcije ili događaja. Izlaz jeste završna faza u Logstash-u, on može da se transformiše u razne formate, ali u našem slučaju to će biti ElasticSearch.

Kako bi se koristio, potrebno ga je instalirati na računaru .

1) Nakon instalacije, sledi kreiranje logstash.conf fajla gde se specificira koji plugin-i se koriste (inputs, outpust, filters) i to omogućava kreiranje Logstash pipeline-a.
2) Pokretanje Logstash-a se izvršava pozicioniranjem u *bin* folder gde je instaliran, otvaranjem terminala i pokretanjem komande *logstash -f logstash.conf*

7. Kibana

Kibana služi za vizuelizaciju podataka koja upotpunjuje ELK Stack I koristi se za vizuelizaciju ElasticSearch dokumenata. Ima svoj dashboard koji nudi različite informacije.

Kibana se takođe instalira na racunaru.

1) Nakon instalacije, potrebno je izmeniti kibana.yml konfiguracioni fajl, kako bi se povezala sa ElasticSearch-om. U našem slučaju to će biti na portu 9200.

```
elasticsearch.url: "http://localhost:9200"
```

2) Pokretanje Kibana-e se izvršava pozicioniranjem u ***bin*** folder gde je instalirana, otvaranjem terminala i pokretanjem ***kibana.bat***
3) Kibana UI-u se može pristupiti i preko adrese ***http://localhost:5601***

8. SerbianAnalyzer

ElasticSearch Analyzers podržavaju mnoštvo jezika, tako da je prilikom analize nekog teksta potrebno navesti koji će se koristiti. Da Kako bi se izvršilo pretprocesiranje teksta na srpskom jeziku, to omogućava SerbianAnalyzer plugin za srpski jezik. Podržavaju različite operacije nad tekstom kao što je konvertovanje ćirilice u latinicu, prebacivanje velikih u mala slova, izbacivanje stop reči i slično… Takođe, analyzer-i se mogu prilagođavati različitim potrebama, tj. postoji mogućnost njihove dorade. Za potrebe ovog projekta, korišćen je navedeni Guthub repozitorijum kao pomoć pri instalaciji SerbianAnalyzera.

Da bi se on ubacio u ElasticSearch, potrebno je izbildovati odgovarajući fajl pomoću Gradle-a.

1) To se radi tako što se otvori terminal u root folderu preuzetog projekta, izvrši naredba ***gradlew clean build***. Tako nastaje arhivirana distrubucija u folderu build-distributions unutar root foldera sa nazivom *serbian-analyzer-1.0-SNAPSHOT.zip.*
2) Potom sledi instaliranje dobijenog fajla kao plugin-a unutar ElasticSearch-a, a to se ostvaruje pozicioniranjem unutar ***bin*** foldera u okviru root foldera u koji je preuzet ElasticSearch. Pomoću naredbe *elasticsearch-plugin install file: <absolute path of distribution archive>* se instalira plugin.
3) Pokretanje elasticsearch servera
4) Primer upita na elasticsearch serveru (Slika 3)
5) Ukoliko je potrebno ukloniti plugin, to je moguće pozicioniranjem unutar bin foldera u okviru root foldera u koji je preuzet ElasticSearch i izvršiti komandu *elasticsearch-plugin remove serbian-analyzer*

```
curl -H 'Content-Type: application/json' -X PUT -D
'{
   "mappings":{
      "properties":{
         "content":{
            "type":"text",
            "fields":{
                "sr":{
                   "type":"text",
                   "analyzer":"serbian"},
                "en":{
                    "type":"text",
                    "analyzer":"english"}
             }
          }
       }
   }
}'
http://localhost:9200/tweet
```

Slika 3 – Primer upita na ElasticSearch serveru

9. Indexing unit

Za razliku od relacionih baza koje čuvaju podatke u tabelama, u ElasticSearch-u koji je NoSQL se čuvaju u vidu JSON objekta. Ovako sačuvani objekti su indeksirani i nad njima se može izvršiti pretraga. ElasticSearch ne čuva čitav dokument u pdf formatu, samo njegovu lokaciju, pa se tom dokumentu pristupa preko nje.

Primer jednog JSON objekta za naš konkretan slučaj prikazan je na Slici 4.

```
{
    "metadata": {
        "applicant": "Nevena Atic",
        "education": "Software engineer",
        "type":"CV",
        "date":"2022-12-20"
    },
    "content": "Something about biography, experience, etc.",
    "path":"path/cv.pdf"
}
```

Slika 4 – primer JSON objekta za CV

10.Geoprostorna pretraga

Za potrebe ovog predmeta, takođe je neophodno omogućiti geoprostornu pretragu aplikanata, za koju je neophodno čuvati informacije o imenu grada, geografskoj širini I duzini lokacije stanovanja.

ElasticSearch ima podršku za kreiranje ovakvih upita I omogućava pretragu dokumenata kako bi pronasao samo one aplikante na osnovu imena grada I radiusa. Nudi nam dve vrste pretraga – geo_point koja će u ovom slučaju biti korišćena jer nudi pretragu po geografskoj širini I dužini I geo_shape koja nudi pretragu po tačkama, linijama, krugovima… Primer jedne pretrage iz geo_point grupe gde je korišćen geo_distance query prikazan je na Slici 5.

```
GET /locations/_search
{
    "query": {
        "bool":{
            "must": {
                "match_all":{}
            }
        },
        "filter":{
            "geo_distance":{
                "distance": "70km",
                "pin.location":{
                    "lon":20,
                    "lat":-25
                }
            }
        }
    }
}
```

Slika 5 – prikaz upita pomoću geo_distance