

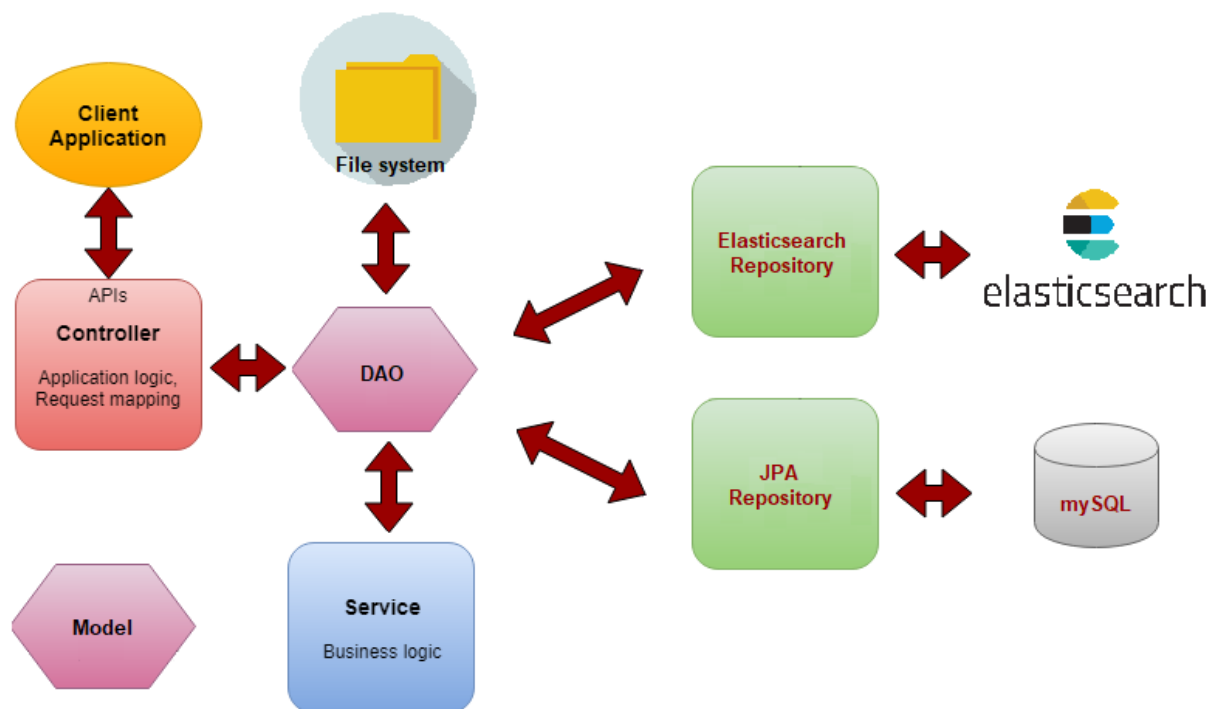
# UPRAVLJANJE DIGITALNIM DOKUMENTIMA – KT1

autor: Nikola Grujčić E2 12/2019 (RA 95/2015)

## Arhitektura aplikacije

Serverski deo aplikacije je implementiran u Javi 8, uz korišćenje Spring okvira. Klijentski deo aplikacije će se realizovati korišćenjem Angular okvira 7.1.14. Klijentski i serverski delovi će međusobno komunicirati putem REST tehnologije, oslanjajući se na HTTPS protokol. Podaci će se razmenjivati u JSON formatu.

Serverska strana će komunicirati sa bazom mySQL putem Hibernate biblioteka (server version: 8.0.16 MySQL Community Server - GPL). Kako bismo komunicirali sa elasticsearch-om koristimo Spring Data Elasticsearch.



Slika 1. Arhitektura aplikacije

Glavni delovi aplikacije (slika 1) koji se mogu identifikovati su:

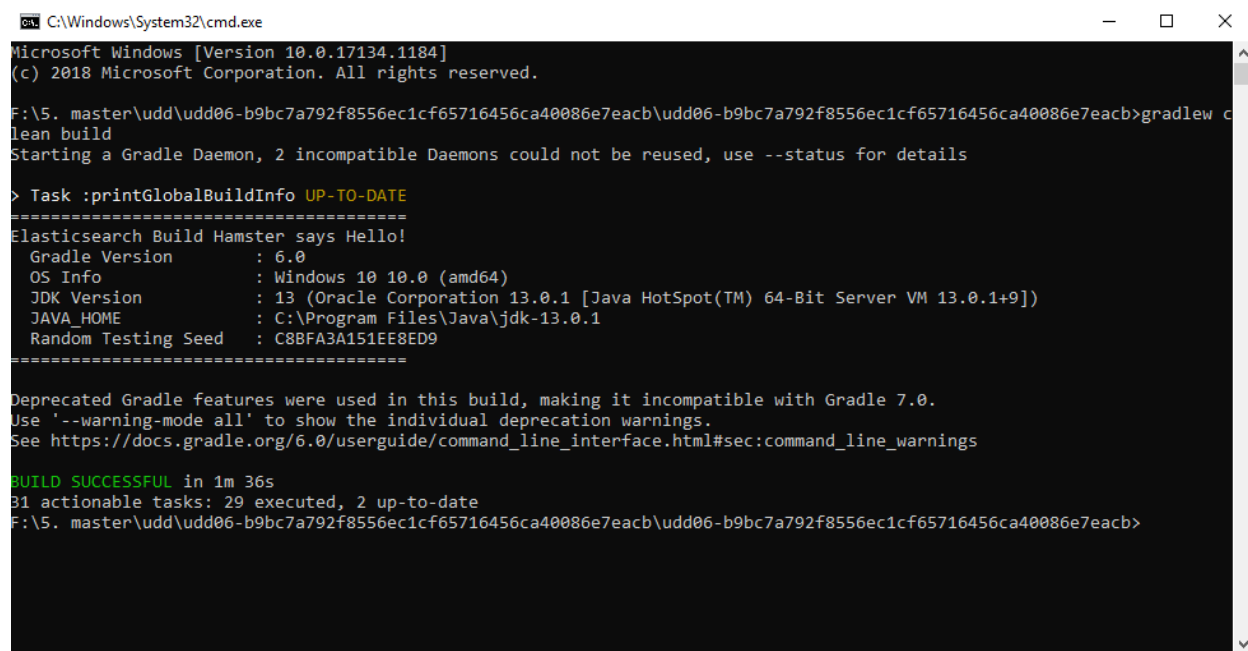
- Client Application – frontend koji će biti realizovan u Angularu;
- Spring boot aplikacija – koja ima sledeće slojeve:
  - Model – u ovom sloju će se definisati modeli podataka koji se čuvaju u mySQL bazi i modeli koji će predstavljati indexUnit;
  - Controllers – interfejsi ka klijentskoj aplikaciji;
  - DAO (Data access object) – interfejsi koji će komunicirati sa elasticsearch-om, sa bazom i sa fajl sistemom i omogućavati izolaciju *persistence* sloja i aplikativnog sloja;

- Services – implementacija DAO interfejsa, tj. poslovne logike;
- Repositories – u ovom sloju će se nalaziti repozitorijumi za elasticsearch i jpa repozitorijumi koji će ostvarivati komunikaciju sa bazom i biti zaduženi za upravljanje objektima u bazi podataka.

## Konfiguracija elasticsearch-a i instalacija plugin-a Serbian-Analyzer

Prvo je preuzet sa profesorovog linka <https://github.com/chenejac/udd06/tree/b9bc7a792f8556ec1cf65716456ca40086e7each> SerbianPlugin koji je pisan za Javu 13.0.1, za Gradle 6.0 i za Elasticsearch 7.4.0. Kako bi projekat mogao da se izbulduje, preuzeta je Java 13.0.1 sa sajta <https://www.oracle.com/technetwork/java/javase/downloads/jdk13-downloads-5672538.html>. 7.4.0.

Nakon instalacije Jave i ispravke sistemskih varijabli JAVA\_HOME i Path na novu verziju jave, bilo je potrebno pozicionirati se na korenski direktorijum projekta i ukucati komandu – *gradlew clean build* (slika 2).



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.1184]
(c) 2018 Microsoft Corporation. All rights reserved.

F:\5. master\udd\udd06-b9bc7a792f8556ec1cf65716456ca40086e7each\udd06-b9bc7a792f8556ec1cf65716456ca40086e7each>gradlew clean build
Starting a Gradle Daemon, 2 incompatible Daemons could not be reused, use --status for details

> Task :printGlobalBuildInfo UP-TO-DATE

=====
Elasticsearch Build Hamster says Hello!
Gradle Version      : 6.0
OS Info             : Windows 10 10.0 (amd64)
JDK Version         : 13 (Oracle Corporation 13.0.1 [Java HotSpot(TM) 64-Bit Server VM 13.0.1+9])
JAVA_HOME           : C:\Program Files\Java\jdk-13.0.1
Random Testing Seed : C8BFA3A151EE8ED9
=====

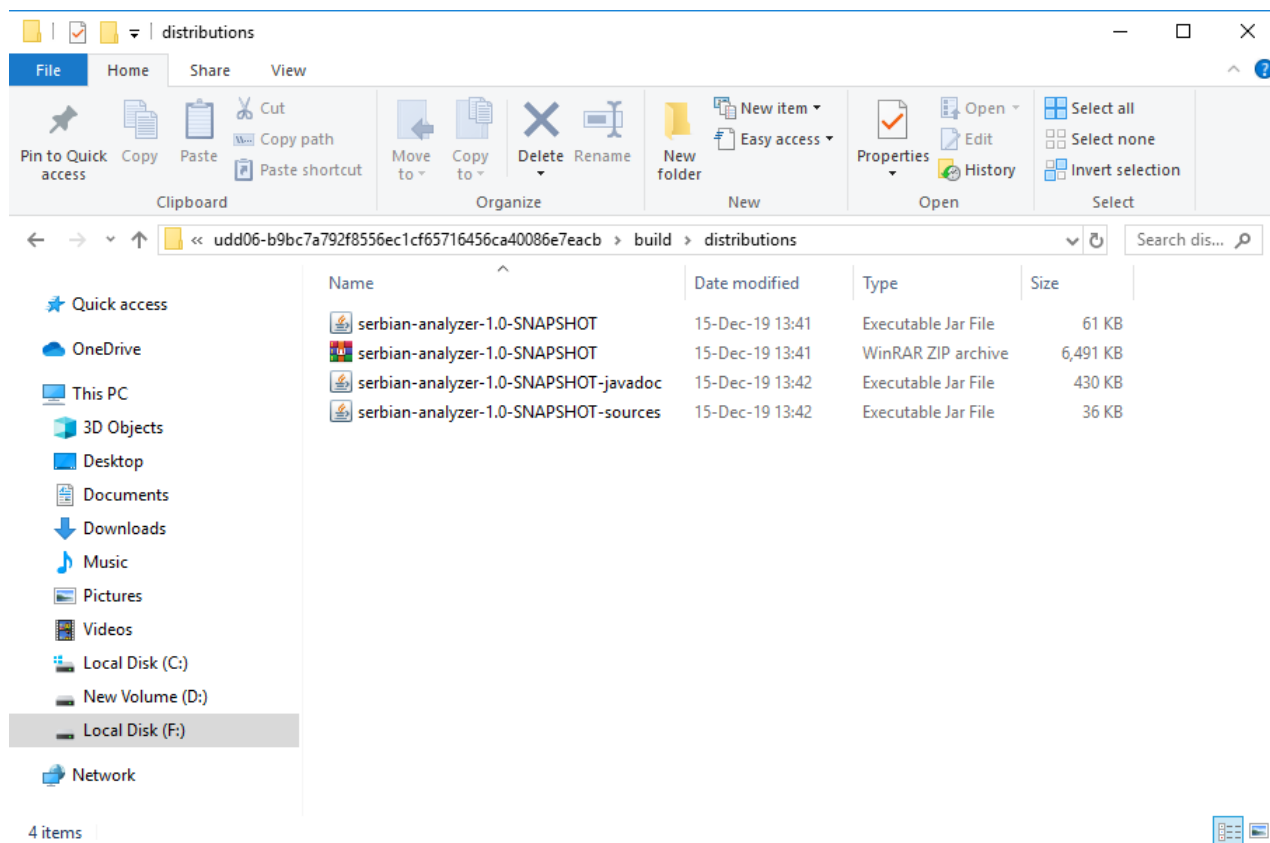
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.0/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 1m 36s
31 actionable tasks: 29 executed, 2 up-to-date
F:\5. master\udd\udd06-b9bc7a792f8556ec1cf65716456ca40086e7each\udd06-b9bc7a792f8556ec1cf65716456ca40086e7each>

```

Slika 2. Komanda za bildovanje projekta

Pošto se plugin uspešno izbuldovao, u build/distributions folderu, nalaziće se serbian-analyzer-1.0-SNAPSHOT.zip, kao što je prikazano na slici 3.



Slika 3. Folder distributions u izbldovanom projektu

Sledeći korak je bio instalacija plugin-a u elasticsearch. Pozicionirao sam se u bin direktorijum elasticsearch-a i ukucio komandu: `elasticsearch-plugin install file:F:\5.master\udd\udd06-b9bc7a792f8556ec1cf65716456ca40086e7eacb\udd06-b9bc7a792f8556ec1cf65716456ca40086e7eacb\build\distributions\serbian-analyzer-1.0-SNAPSHOT.zip`. Nakon toga sam dobio potvrdu o uspešno instaliranom plugin-u (slika 4). Prilikom narednog pokretanja elasticsearch-a može se videti da je učitao serbian-analyzer plugin (slika 5).

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.1184]
(c) 2018 Microsoft Corporation. All rights reserved.

F:\5.master\udd\elasticsearch-7.4.0-windows-x86_64\elasticsearch-7.4.0\bin>elasticsearch-plugin install file:///F:\5.master\udd\udd06-b9bc7a792f8556ec1cf65716456ca40086e7eacb\udd06-b9bc7a792f8556ec1cf65716456ca40086e7eacb\build\distributions\serbian-analyzer-1.0-SNAPSHOT.zip
-> Downloading file:///F:\5.master\udd\udd06-b9bc7a792f8556ec1cf65716456ca40086e7eacb\udd06-b9bc7a792f8556ec1cf65716456ca40086e7eacb\build\distributions\serbian-analyzer-1.0-SNAPSHOT.zip
[=====] 100%
-> Installed serbian-analyzer

F:\5.master\udd\elasticsearch-7.4.0-windows-x86_64\elasticsearch-7.4.0\bin>
```

Slika 4. Instalacija SerbianAnalyzer-a

```
C:\Windows\System32\cmd.exe - elasticsearch.bat
[2019-12-15T14:44:47,888][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [ingest-geoip]
[2019-12-15T14:44:47,889][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [ingest-user-agent]
[2019-12-15T14:44:47,891][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [lang-expression]
[2019-12-15T14:44:47,897][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [lang-mustache]
[2019-12-15T14:44:47,898][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [lang-painless]
[2019-12-15T14:44:47,899][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [mapper-extras]
[2019-12-15T14:44:47,900][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [parent-join]
[2019-12-15T14:44:47,902][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [percolator]
[2019-12-15T14:44:47,904][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [rank-eval]
[2019-12-15T14:44:47,909][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [reindex]
[2019-12-15T14:44:47,911][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [repository-url]
[2019-12-15T14:44:47,911][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [search-business-rules]
[2019-12-15T14:44:47,912][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [spatial]
[2019-12-15T14:44:47,913][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [transport-netty4]
[2019-12-15T14:44:47,914][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [vectors]
[2019-12-15T14:44:47,915][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-analytics]
[2019-12-15T14:44:47,916][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-ccr]
[2019-12-15T14:44:47,922][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-core]
[2019-12-15T14:44:47,923][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-deprecation]
[2019-12-15T14:44:47,925][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-graph]
[2019-12-15T14:44:47,926][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-ilm]
[2019-12-15T14:44:47,927][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-logstash]
[2019-12-15T14:44:47,929][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-ml]
[2019-12-15T14:44:47,932][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-monitoring]
[2019-12-15T14:44:47,934][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-rollup]
[2019-12-15T14:44:47,936][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-security]
[2019-12-15T14:44:47,938][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-sql]
[2019-12-15T14:44:47,939][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-voting-only-node]
[2019-12-15T14:44:47,940][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded module [x-pack-watcher]
[2019-12-15T14:44:47,945][INFO ][o.e.p.PluginsService] [DESKTOP-1A0VH48] loaded plugin [serbian-analyzer]
```

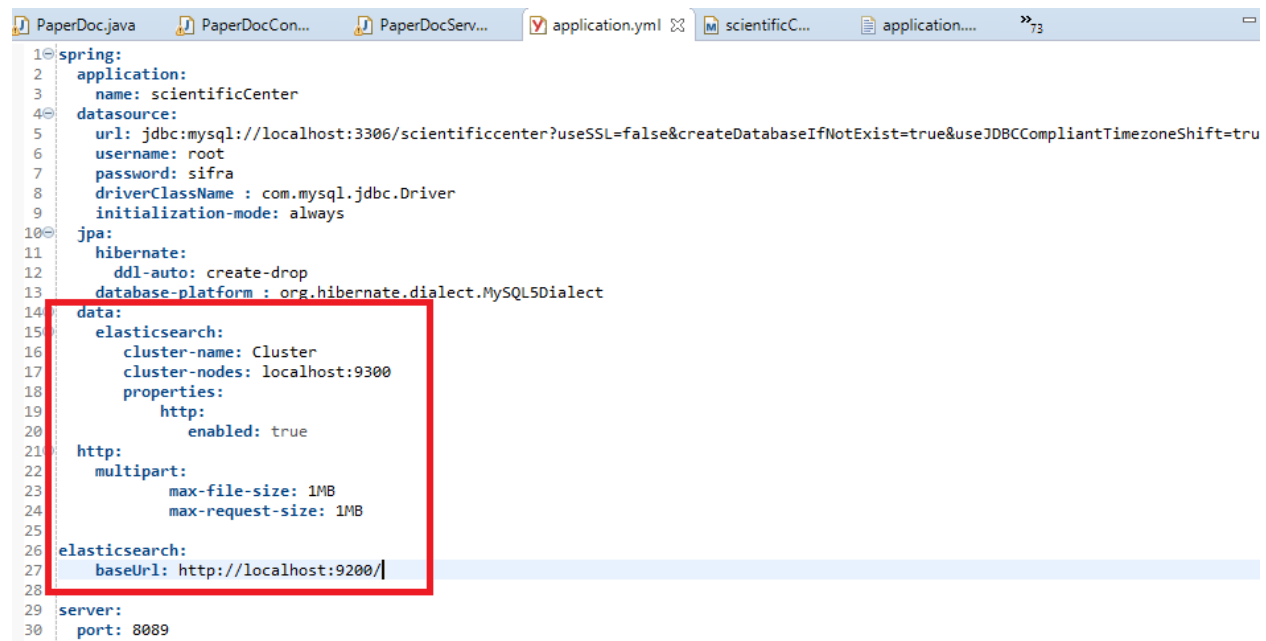
Slika 5. Potvrda o instaliranom analyzer-u

## Povezivanje Spring Boot aplikacije sa elasticsearch-om

Kako bi Spring Boot aplikacija mogla da uspostavi komunikaciju sa elasticsearch-om u pom.xml-u je dodat dependency:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
  <version>1.5.6.RELEASE</version>
</dependency>
```

U application.yml-u je bilo potrebno da se definiše url od elasticsearch-a i od cluster nodes-a (slika 6).



Slika 6. Screenshot application.yml-a

Napravljen je model rada (*PaperDoc*), on predstavlja dokument koji se indeksira. Definisan je indeks pod nazivom „*paperlibrary*“ i tip koji će taj indeks da sadrži „*paper*“. Za *shards* je postavljen broj 1, jer nije potrebno da se smanjuje indeks na više delova, a za *replicas* takođe trenutno nije postavljeno da je potrebna kopija podataka. Ubačeni su osnovni podaci koji su neophodni za osnovnu pretragu i MoreLikeThis pretragu. Za polja je definisan serbian analyzer koji smo instalirali u elasticsearch-u. Prethodno opisan model rada je prikazan na slici 7.

```

20 @Document(indexName = PaperDoc.INDEX_NAME, type = PaperDoc.TYPE_NAME, shards = 1, replicas = 0)
21 public class PaperDoc {
22
23     public static final String INDEX_NAME = "paperlibrary";
24     public static final String TYPE_NAME = "paper";
25
26     public static final String DATE_PATTERN = "yyyy-MM-dd";
27
28
29     @Id
30     @Field(type = FieldType.Text, store = true)
31     private Long id;
32
33     @Field(type = FieldType.Text, analyzer="serbian", store = true)
34     private String title;
35
36     @Field(type = FieldType.Text, analyzer="serbian", store = true)
37     private String journaltitle;
38
39     @Field(type = FieldType.Text, analyzer="serbian", store = true)
40     private String author;
41
42     @Field(type = FieldType.Text, analyzer="serbian", store = true)
43     private String coauthors;
44
45     @Field(type = FieldType.Text, analyzer="serbian", store = true)
46     private String keywords;
47
48     @Field(type = FieldType.Text, analyzer="serbian", store = true)
49     private String area;
50
51     @Field(type = FieldType.Boolean, store = true)
52     private Boolean status;
53

```

Slika 7. Deo modela indexUnit-a za radove

Pored repozitorijuma JPA koji će se koristiti za *mySQL* bazu, napravljen je i repozitorijum za dokumente koji nasleđuje *ElasticsearchRepository*.

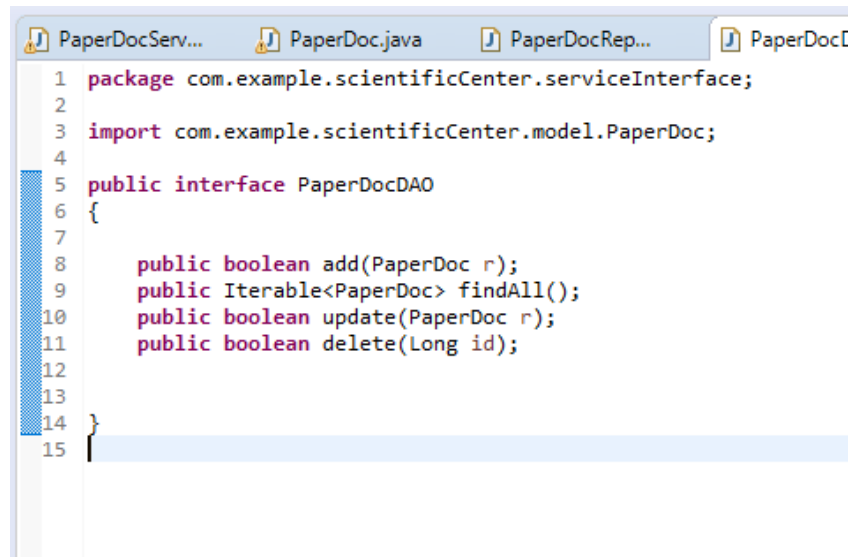
```

1 package com.example.scientificCenter.repository;
2
3 import org.springframework.data.elasticsearch.repository.ElasticsearchRepository;
4
5
6
7 public interface PaperDocRepository extends ElasticsearchRepository<PaperDoc, Long> {
8
9 }
10
11
12

```

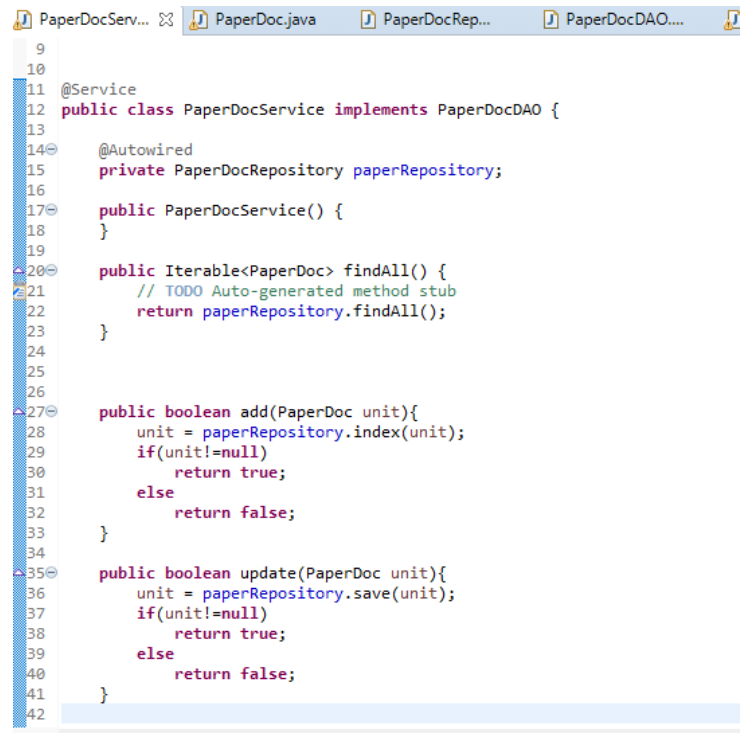
Slika 8. Elasticsearch repozitorijum za radove

Napravljen je poseban servis za manipulisanje repozitorijumom za *elasticsearch* (slika 10). Servisom manipulišemo iz klase *PaperDocDAO* (slika 9). Trenutni kontroler za radove je prikazan na slici 11.



```
1 package com.example.scientificCenter.serviceInterface;
2
3 import com.example.scientificCenter.model.PaperDoc;
4
5 public interface PaperDocDAO
6 {
7
8     public boolean add(PaperDoc r);
9     public Iterable<PaperDoc> findAll();
10    public boolean update(PaperDoc r);
11    public boolean delete(Long id);
12
13 }
14
15
```

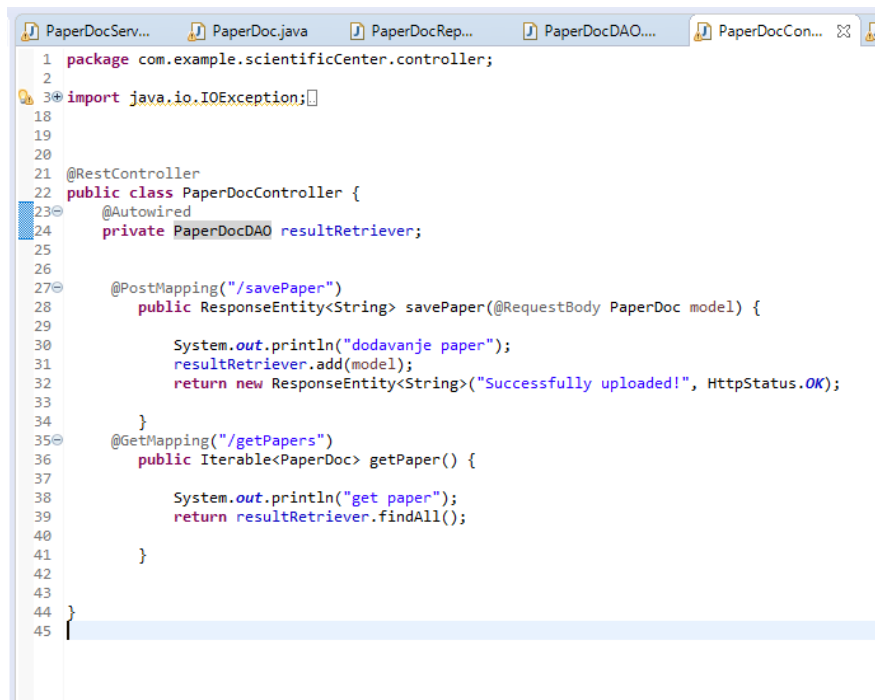
Slika 9. Interfejs DAO za radove



```
9
10
11 @Service
12 public class PaperDocService implements PaperDocDAO {
13
14     @Autowired
15     private PaperDocRepository paperRepository;
16
17     public PaperDocService() {
18     }
19
20     public Iterable<PaperDoc> findAll() {
21         // TODO Auto-generated method stub
22         return paperRepository.findAll();
23     }
24
25
26     public boolean add(PaperDoc unit){
27         unit = paperRepository.index(unit);
28         if(unit!=null)
29             return true;
30         else
31             return false;
32     }
33
34     public boolean update(PaperDoc unit){
35         unit = paperRepository.save(unit);
36         if(unit!=null)
37             return true;
38         else
39             return false;
40     }
41 }
42
```

Slika 10. Servis za radove





```
1 package com.example.scientificCenter.controller;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 @RestController
22 public class PaperDocController {
23     @Autowired
24     private PaperDocDAO resultRetriever;
25
26
27     @PostMapping("/savePaper")
28     public ResponseEntity<String> savePaper(@RequestBody PaperDoc model) {
29
30         System.out.println("dodavanje paper");
31         resultRetriever.add(model);
32         return new ResponseEntity<String>("Successfully uploaded!", HttpStatus.OK);
33     }
34
35     @GetMapping("/getPapers")
36     public Iterable<PaperDoc> getPaper() {
37
38         System.out.println("get paper");
39         return resultRetriever.findAll();
40     }
41
42
43
44 }
45 }
```

*Slika 11. Kontroler za radove*

## Čuvanje podataka

Metapodaci će se čuvati u mySQL bazi, a PDF fajl će se čuvati u fajl sistemu. U bazi će se čuvati lokacija do PDF fajla. U konfiguraciji elasticsearch-a je podešeno gde će se čuvati indeksi, trenutno je podešeno da se čuvaju na lokaciji projekta (slika 12). Na slici 13 je prikazano stablo projekta u eclipse-u u kojem se nalazi nodes. Kako bi se svi podaci, iako na različitim lokacijama, jedinstveno ažurirali, predviđeno je, kao što je ranije prikazano, korišćenje Data Access Object pattern-a.

```

elasticsearch.yml x
7  # The primary way of configuring a node is via this file. This template lists
8  # the most important settings you may want to configure for a production cluster.
9  #
10 # Please consult the documentation for further information on configuration options:
11 # https://www.elastic.co/guide/en/elasticsearch/reference/index.html
12 #
13 # ----- Cluster -----
14 #
15 # Use a descriptive name for your cluster:
16 #
17 cluster.name: Cluster
18 #
19 # ----- Node -----
20 #
21 # Use a descriptive name for the node:
22 #
23 #node.name: node-1
24 #
25 # Add custom attributes to the node:
26 #
27 #node.attr.rack: r1
28 #
29 # ----- Paths -----
30 #
31 # Path to directory where to store the data (separate multiple locations by comma):
32 #
33 path.data: D:\naucnacentrala\naucnaCentrala\scientificCenter\scientificCenter
34 #
35 # Path to log files:
36 #
37 #path.logs: /path/to/logs
38 #
39 # ----- Memory -----
40 #
41 # Lock the memory on startup:
42 #
43 #bootstrap.memory_lock: true
44 #

```

Slika 12. Application.yml od elasticsearch-a

```

> JRE System Library [JavaSE-1.8]
> Maven Dependencies
v > nodes
  v > 0
    _state
  v > indices
    v > ijoSeAUKQD-eTDeqoMvMSA
      _state
      v > 0
        _state
        > index
        > translog
      v > qtSYFdZlQtqhKzeEOR6vOg
        > _state
        > 0
        node.lock

```

Slika 13. Prikaz foldera nodes u direktorijumu projekta u eclipse-u

## Indeksne strukture

Odlučeno je da postoje tri indeksne strukture:

- prihvaćeni radovi;
- neprihvaćeni radovi;
- recenzenti.

Za potrebe javne pretrage bih pretraživao samo prihvaćene radove. Na ovaj način će se ova pretraga izvršiti veoma brzo, jer ova struktura neće sadržati neprihvaćene radove.

Prilikom traženja recenzenata koji su ranije pregledali slične radove prolaziće se kroz dve strukture, jednu koja sadrži samo prihvaćene radove i drugu koja sadrži neprihvaćene radove (u fazi obrade ili odbijene). Ovde će se obavljati MoreLikeThis pretraga. Tu je problem dupliranja podataka i pretraga će biti sporija. Pošto će svaki recenzent imati jedinstveni identifikator, duplirani rezultati će se lako otkloniti.

Poslednja indeksna struktura – recenzenti, koristiće se za potrebe geoprostorne pretrage recenzenata.

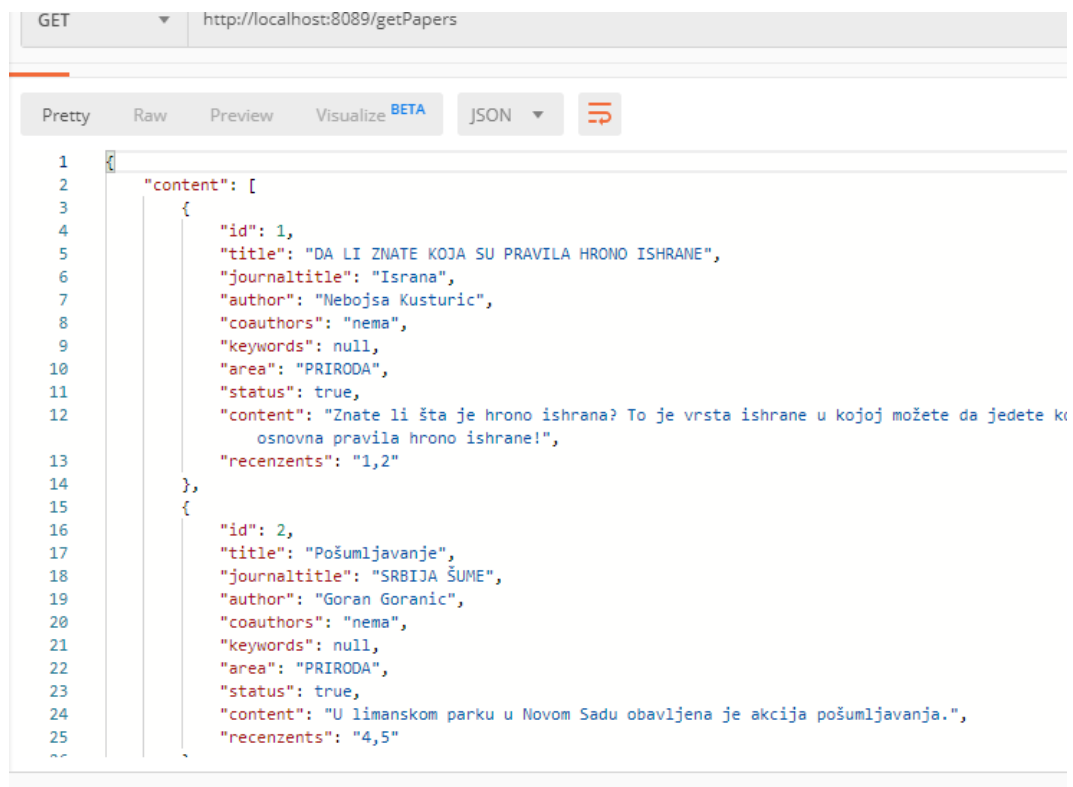
Prilikom razmatranja da li je dovoljno imati samo jednu indeksnu strukturu sa svim radovima (prihvaćenim i neprihvaćenim), došao sam do zaključka da je bolje rešenje da se razdvoje ove strukture. Pošto prilikom ostavljanja mogućnosti korisniku da *download*-uje *open-access* rad, ovo dugme će se brže prikazati ukoliko se prolazi samo kroz strukturu koja sadrži prihvaćene radove i proverava se status rada (da li je *open-access* ili nije), nego da prolazi kroz strukturu koja će imati i prihvaćene i neprihvaćene radove.

## Zahtevi za čuvanje dokumenata i za izlistavanje sadržaja indeksa

Na slici 14. je prikazan POST zahtev prilikom kojeg čuvamo u indeksnoj strukturi novi rad. Trenutno se čuvaju samo oni podaci koji su relevantni za pretrage. Na slici 15. je prikazan deo odgovora prilikom slanja GET zahteva za pribavljanje radova koji su prethodno sačuvani u indeksnoj strukturi.



Slika 14. Čuvanje novog rada



Slika 15. Deo odgovora za zahtev - listanje sačuvanih radova

## More like this pretraga

Prilikom traženja sličnih prethodno obrađenih radova i recenzenata koji su učestvovali u njihovom recenziranju, koristiće se *More Like This* pretraga. Query od ove pretrage će se formirati tako što će se preuzeti tekst podnetog rada. Izvršiće se upit koji će vratiti željeni rezultat. “*Fields*”, tj. polje koje će se ispitivati je sadržaj rada, a u okviru “*like*” će se naći sadržaj rada koji se prilaže.

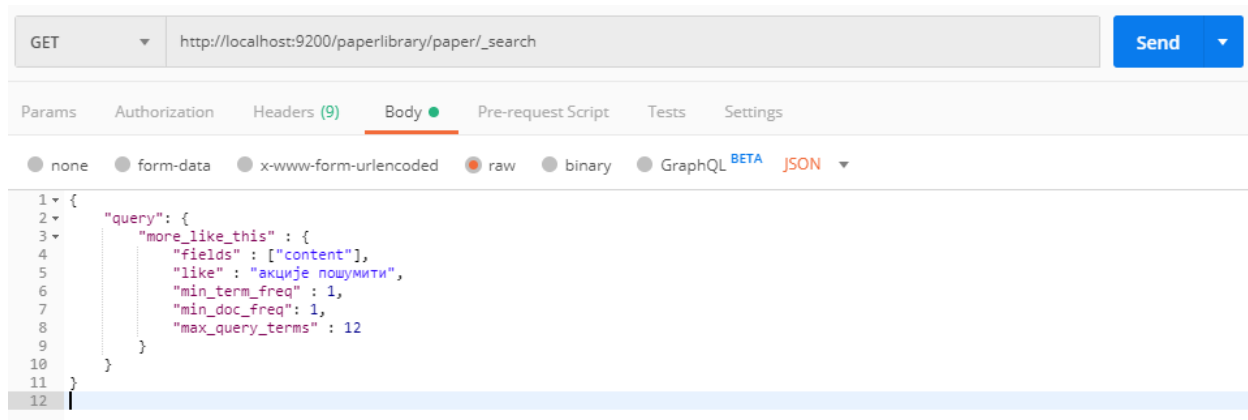
Na slici 16 je prikazan upit nad indeksom “*paperlibrary*” i tipom “*paper*”. Traženi su dokumenti čiji je tekst nalik tekstu u “*like*” propertyju. U indeksu se trenutno nalaze 5 radova, tri o zdravoj ishrani i dva o pošumljavanju. Pretraga je samo vratila radove koji su nalik podnetom sadržaju (slika 17). Radi demonstracije instaliranog plugin-a za Serbianalyzer, tražene reči su bile: “акције пошумити”, a pronađeni su radovi koji sadrže reči “акција”, “пошумити”.

Polje “*min\_term\_freq*” predstavlja minimalnu učestalost navedenog sadržaja u radovima. Trenutno je setovano na 1, ali će se ovaj broj uskladiti kada bude bilo više reči u zahtevu.

Polje “*min\_doc\_freq*” označava minimalnu učestalost dokumenata. Trenutno je postavljeno na jedinicu, jer indeks sadrži samo pet dokumenata.

Polje “*max\_query\_terms*” predstavlja maksimalan broj termova u query-ju koji će se uzeti u obzir prilikom pretrage.

Ova pretraga će, kao što je ranije navedeno, biti implementirana za prolazak kroz dve indeksne strukture, jednu sa prihvaćenim radovima i drugu sa neprihvaćenim radovima.



Slika 16. Upit za *MoreLikeThis* pretragu nad radovima

```
1 {
2   "took": 5,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 1.8032078,
16    "hits": [
17      {
18        "_index": "paperlibrary3",
19        "_type": "paper",
20        "_id": "5",
21        "_score": 1.8032078,
22        "_source": {
23          "id": 5,
24          "title": "Procena šumovitosti u Vojvodini",
25          "journaltitle": "Vojvodina šume",
26          "author": "Mirko Mirković",
27          "coauthors": "nema",
28          "keywords": null,
29          "area": "ŠUME",
30          "status": true,
31          "content": "Optimalna površina pod šumom i Zaštitnim zasadima procenjena je na osnovu postavljenog cilja obezbeđenja
32                    prosekne površine šuma i zaštitnog zelenila na nivou od 0,16 ha po glavi stanovnika. Potrebno je izvršiti dovoljno
33                    akcija i pošumiti veći deo Srbije",
34          "recenzents": "4,5"
35        }
36      },
37      {
38        "_index": "paperlibrary3",
39        "_type": "paper",
40        "_id": "2",
41        "_score": 1.1568694,
42        "_source": {
43          "id": 2,
44          "title": "Pošumljavanje",
45          "journaltitle": "SRBIJA ŠUME",
46          "author": "Goran Goranic",
47          "coauthors": "nema",
48          "keywords": null,
49          "area": "PRIRODA",
50          "status": true,
51          "content": "U limanskom parku u Novom Sadu obavljena je akcija pošumljavanja.",
52          "recenzents": "4,5"
53        }
54      }
55    ]
56  }
57 }
```

Slika 17. Odgovor na MoreLikeThis upit

## Geoprostorna pretraga

Kako bi se pronašli recenzenti koji su udaljeni barem 100 km od grada u kojem se nalazi autor podnetog rada, sastavio sam poseban query u kojem sam u svojstvu „*geo\_distance*“ postavio svojstvu „*location*“ na koordinate grada u kojem živi autor, a u svojstvu „*distance*“ sam postavio udaljenost od 100 km. U upitu koji sam naveo, naznačio sam da se izlistaju oni gradovi koji se ne nalaze u radijusu od 100 km od navedenog grada (slika 19).

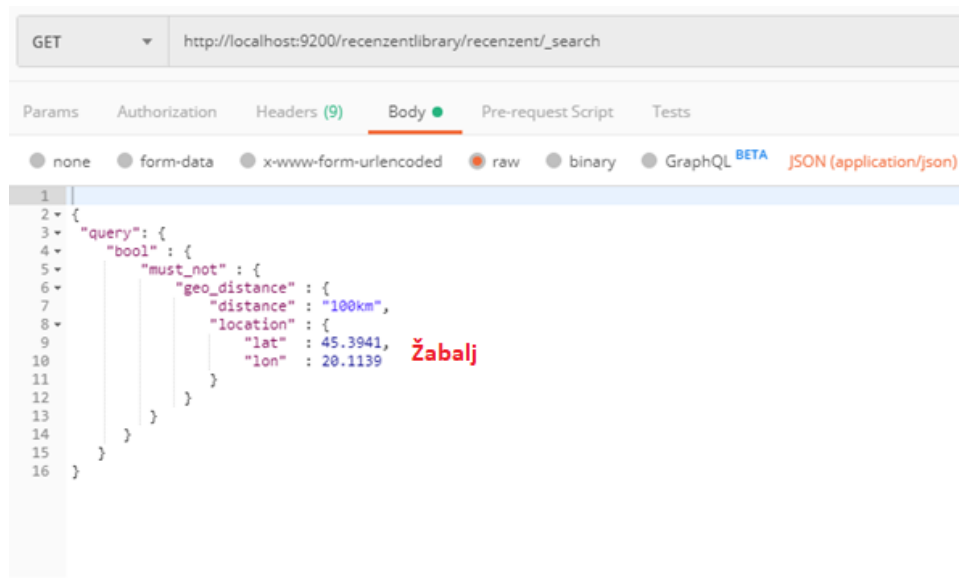
Na slici 18. je prikazan model indexunit-a recenzenta u javi. Polje „*location*“ je tipa *GeoPoint* i koristi se istoimena notacija.

U indeksu su se nalazili recenzenti iz Novog Sada, Beograda, Zrenjanina, Kragujevca i Niša. Prilikom upita koji se odnosi na autora iz Žablja, prikazani su samo recenzenti koji su iz Kragujevca i Niša (slika 20). Ostali gradovi su u radijusu od 100 km od Žablja.



```
1 package com.example.scientificCenter.model;
2
3 import javax.persistence.Id;
10 @Document(indexName = RecenzentDoc.INDEX_NAME, type = RecenzentDoc.TYPE_NAME, shards = 1, replicas = 0)
11 public class RecenzentDoc {
12     public static final String INDEX_NAME = "recenzentlibrary";
13     public static final String TYPE_NAME = "recenzent";
14
15     public static final String DATE_PATTERN = "yyyy-MM-dd";
16
17     @Id
18     @Field(type = FieldType.Text, store = true)
19     private Long id;
20
21     @Field(type = FieldType.Text, store = true)
22     private String username;
23
24     @GeoPointField
25     private GeoPoint location;
26
27
28
29     public Long getId() {
30         return id;
31     }
32
33     public void setId(Long id) {
34         this.id = id;
35     }
36
37     public String getUsername() {
38         return username;
39     }
}
```

Slika 18. Deo modela indexunit-a za recenzenta



Slika 19. Upit za geoprostornu pretragu nad recenzentima



```
GET http://localhost:9200/recenzentlibrary/recenzent/_search Send

Pretty Raw Preview JSON

1 {
2   "took": 15,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 0,
16    "hits": [
17      {
18        "_index": "recenzentlibrary",
19        "_type": "recenzent",
20        "_id": "2",
21        "_score": 0,
22        "_source": {
23          "id": 2,
24          "username": "recenzent2",
25          "location": {
26            "lat": 43.3209,
27            "lon": 21.8958      Niš
28          }
29        }
30      },
31      {
32        "_index": "recenzentlibrary",
33        "_type": "recenzent",
34        "_id": "4",
35        "_score": 0,
36        "_source": {
37          "id": 4,
38          "username": "recenzent4",
39          "location": {
40            "lat": 44.0128,
41            "lon": 20.9114      Kragujevac
42          }
43        }
44      }
45    ]
46  }
47 }
```

Slika 20. Odgovor na upit za geoprostornu pretragu