



ИМПЛЕМЕНТАЦИЈА КОМБИНОВАНЕ ГЕОПРОСТОРНЕ, ВРЕМЕНСКЕ И ТЕКСТУАЛНЕ ПРЕТРАГЕ

THE IMPLEMENTATION OF COMBINED GEOSPATIAL, TEMPORAL AND TEXTUAL SEARCH

Александар Јеремић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – У раду је приказана комбинација геопросторне, временске и текстуалне претраге у оквиру веб апликације за претраживање догађаја у неком граду. Решење је имплементирано коришћењем Еластисеарџ сервера за претрагу докумената. Такође за имплементацију је коришћен Спринг радни оквир, као и Ангулар за имплементацију клијентске стране.

Кључне речи: *Elasticsearch*, геопросторна претрага, временска претрага, текстуална претрага

Abstract – *In this work, a combined geospatial, temporal and textual search within a web application for searching events in a city was shown. The solution was implemented using Elasticsearch server for document search. Additionally, Spring framework was used for the implementation, as well as Angular for the implementation of the client's part.*

Keywords: *Elasticsearch*, geospatial search, temporal search, textual search

1. УВОД

Задатак овог рада је да се направи веб апликација која ће помоћи корисницима да брже дођу до информације који догађаји се дешавају у њихом граду или граду који желе да посете. Културни, музички, спортски и други догађаји обједињени су у једну апликацију која има циљ да крајњем кориснику, у зависности од његове претраге да информацију о времену одржавања и детаљима догађаја који ће бити исцртани на мапи, на локацији њиховог одржавања. Главни акценат овог рада дат је геопросторној, временској и текстуалној претрази помоћу *Elasticsearch* сервера.

2. ТЕОРИЈСКЕ ОСНОВЕ

2.1 Дигитални документ

Дигитални документ је рачунарски обрађена информација којом се рукује као основном јединицом обраде. Лични рачунари су данас доступни већини становништва, а поред њих и разни други уређаји који су опремљени потребним хардвером и софтвером који омогућује креирање и конзумирање различитих дигиталних докумената [1].

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Драган Ивановић, ред. проф.

Метаподаци - представљају податке о документу, односно то су подаци о подацима.

Метаподатке користимо како бисмо боље организовали колекције докумената и тако преглед докумената прилагодили корисницима. Осим тога, метаподатке можемо користити и за класификацију докумената. Они могу битно унапредити како квалитет тако и могућност претраге колекције дигиталних докумената, а можемо их користити и за сортирање и филтрирање резултата претраге [1].

2.2 Проналажење информација

Када говоримо о проналажењу информација (енг. *Information retrieval*) морамо прво рећи да је то област која се бави техникама за репрезентацију, складиштење, организацију, приступ и проналажење информација. Поменуте технике треба да обезбеде проналажење материјала неструктуриране природе, најчешће текстуалних докумената, и то у оквиру велике колекције која задовољава корисникове потребе за информацијама. Претраживање је процес обраде упита и проналажење информација које корисник тражи и укључује технике за ефикасан приступ и проналажење информација у претходно креираним индексима у процесу индексирања [1].

2.3 Текстуална претрага

Текстуални дигитални документи се састоје од речи. Речи представљају одређени број знакова који се појављује у тексту. Токен представља инстанцу речи која се појављује у тексту у оквиру области проналажења информација. Док терм представља једну класу еквиваленције речи. Уколико се речи, након примене одређених правила за претпроцесирање текста, свде на исти низ знакова, онда припадају истој класи еквиваленције. Квалитет система за претрагу текстуалних дигиталних докумената умногоме зависи од правила за претпроцесирање текста која су различита.

2.4 Геопросторна претрага

Документ може да садржи податке о локацији, док локација може да се односи на више ствари као на пример где је документ настао, где се документ налази итд. Претрагу докумената такође можемо вршити на основу тих информација. Таква претрага назива се геопросторна.

2.5. Временска претрага

Помоћу временске претраге можемо да претражујемо документе који од података имају датум тј. информацију о времену. То време може бити датум креирања документа, датум измене документа, датум отварања документа итд. Приликом претраге потребно је дефинисати временску зону, да би се датум и време на основу ког претражујемо поклопио са временском зоном документа.

3. КОРИШЋЕНЕ ТЕХНОЛОГИЈЕ

Јава је објектно-оријентисан програмски језик. Један је од најпопуларнијих програмских језика на свету.

Кажемо да је објектно-оријентисан програмски језик јер подржава концепте објектно – оријентисаног програмирања као што су: класе, објекти, учаурење, наслеђивање, полиморфизам, преклапање и прекривање метода, интерфејсе, апстрактне класе, итд.

Спринг је радни оквир за Јава платформу, који обезбеђује свеобухватни модел програмирања и конфигурације за модерне пословне апликације засноване на језику Јава. Спринг радни оквир је оквир отвореног кода са изузетно квалитетном подршком за развој апликација, што омогућује инжењерима да се фокусирају на писање пословне логике [3].

Спринг је популаран из неколико разлога, пре свега због поједностављеног тестирања кода, смањења управљачког кода и архитектурне флексибилности.

Предности Спринг радног оквира су што је минимално зависан ка другим библиотекама и оквирима. Такође предност спринга је што не тера програмера да наследи неку класу или имплементира интерфејсе, а омогућава креирање скалабилних апликација користећи *POJO (Plain old Java Object)*. Главни концепти Спринг радног оквира су инверзија контроле (енг. *Inversion of Control*) и повезивање објеката коришћењем инјекције зависности (енг. *Dependency injection*).

Elasticsearch је дистрибуиран и отворен механизам за претраживање и анализу свих врста података, укључујући текстуалне, нумеричке, геопросторне, временске, структуриране и неструктуриране податке. *Elasticsearch* је централна компонента *ElasticStack* - а, скупа алата који су бесплатни и отвореног кода и који служе за уношење података, складиштење, анализу и визуелизацију. *Elasticsearch* је базиран на *Lucene* -у, развијен у Јава програмском језику и објављен као пројекат отвореног кода под условима *Apache* лиценце. Претраживање се састоји од једног или више упита који се комбинују и шаљу *Elasticsearch*-у. Документи који одговарају упитима претраживања враћају се у поготке или резултате претраживања. Претраживање такође може садржати додатне информације које се користе за бољу обраду упита. На пример, претрага може бити ограничена на одређени индекс или само дати одређени број резултата [11].

ElasticSearch подржава два типа поља за геопросторну претрагу, поље типа *geo_point* које подржава парове географска ширина-географска дужина и *geo_shape* поље које подржава тачке, линије, полигоне, итд. [12]

Упит за опсег (*Range query*) проналази документе са пољима који имају вредности у задатом опсегу. Тип *Lucene* упита зависи од типа поља, за поља која имају тип стринг, користи се *TermRangeQuery*, док се за поља која су типа број или датум корисити *NumericRangeQuery* [14].

Ангулар представља платформу и окружење које се користи за креирање ефикасних клијентских *SPA (Single Page application)* коришћењем *HTML*-а и *typescript*-а. Код Ангулара је јако битно да се направи разлика између *Angular.js* радног оквира и *Anglura.io* радног оквира. Прва настала верзија Ангулара је *Angular.js* радни оквир, који у својој синтакси користи *JavaScript* језик. Код *Anglura.io* се за синтаксу користи *typescript* и он се посебним компајлерима преводи у *JavaScript*.

Његов основни елемент је модул који служи да обједини компоненте, директиве, сервисе, разне функције па и друге модуле у једну логичку целину која се касније може користити самостално. Модули служе да окупе сличан код по функционалности који се назива модул корен (*root module*) око кога се касније по потреби надограђују остали модули.

Leaflet је водећа *JavaScript* библиотека отвореног кода за интерактивне мапе прилагођена за рад како на десктоп рачунарима тако и на мобилним уређајима. Заузима само 39Кб и има све функције мапирања које су већини програмера потребне.

4. СПЕЦИФИКАЦИЈА

У раду је имплементирана веб апликација за претраживање догађаја на мапи. Догађаји се исцртавају на мапи на основу упита корисника. Корисник задаје место тј. град за који хоће да претражи догађаје и опционо уноси датум на основу кога ће се приказивати догађаји, као и категорију догађаја. Датум може да буде данашњи дан, неки дан у будућности или временски интервал између два унесена датума.

Корисник такође има могућност претраге самих догађаја на основу назива или описа догађаја. Након таквог упита кориснику ће бити излистани догађаји који задовољавају упит, након чега он бира догађај који жели да прегледа.

Приликом креирања догађаја задаје се његова географска локација тј. географска ширина и географска дужина места на којем се одржава. На тај начин се догађај на мапи исцртава на месту свог одржавања. Апликацију могу да користе и корисници који нису регистровани на систем, односно могу да претражују и прегледају догађаје. Додавање догађаја и њихову измену могу вршити само корисници који су регистровани на систем и који имају улогу уредника у том систему. Главни акценат апликације јесте на претраживању догађаја на основу геопросторне и временске претраге, односно претраживање на основу локације и датума одржавања догађаја.

Архитектура система апликације за претраживање догађаја на мапи је следећа:

- *Backend* део тј. апликација на серверској страни,

- *Frontend* део тј. апликација на клијентској страни,
- *Elasticsearch* и
- Релациона база података.

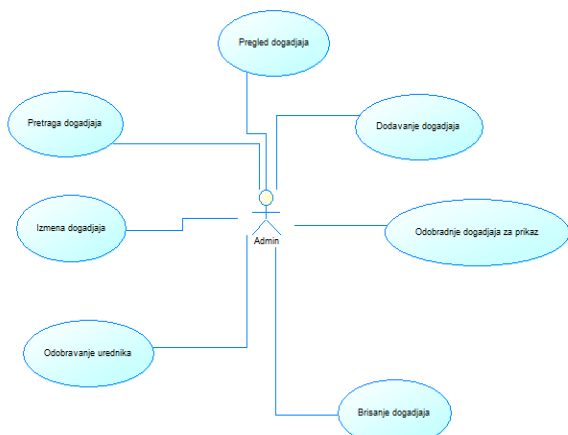
Backend апликација имплементирана је у Јава програмском језику уз помоћ Спринг радног оквира. Представља главни део апликације јер је овај део задужен за комуникацију са свим деловима апликације.

Клијентска страна апликације имплементирана је уз помоћ Ангулар радног оквира. Она је задужена за обраду података добијених са серверске стране и њихово приказивање крајњем кориснику. Тако се на клијентској страни налази мапа за преглед догађаја који су добијени као резултат корисничког упита. Мапа је имплементирана уз помоћ *Leaflet* библиотеке.

Што се тиче *Elasticsearch* он садржи индексираних податке о градовима и догађајима. Комуницира са *Backend* апликацијом и враћа резултате упита.

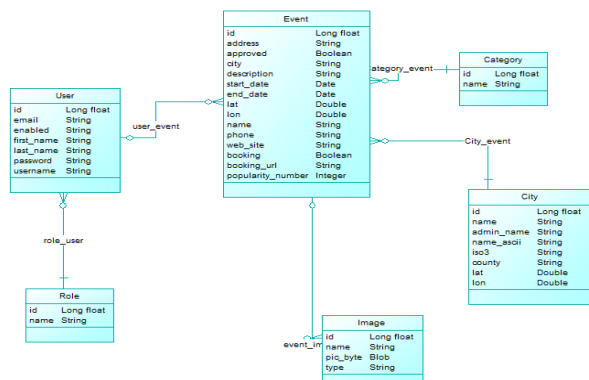
Релациона база података која је коришћена приликом имплементације апликације је *MySQL*.

На слици 1 приказан је случај коришћења админа у систему.



Слика 1 Случај коришћења админа у систему

На слици 2 приказан је модел података апликације



Слика 2 Модел података

5. ИМПЛЕМЕНТАЦИЈА

На слици 3 приказана је имплементација упита ка *Elasticsearch* серверу. Прво је неопходно упутити

упит *Elasticsearch*-у за добијање информација о задатом граду. У *Elasticsearch*-у је претходно индексирано преко 26.000 градова. Поља која су коришћена приликом индексирања градова су назив града, држава, *iso* код, *geoPoint* поље које садржи географску ширину и географску дужину града, као и оригиналан назив града у држави у којој се град налази. На пример ако је назив града индексирани као *Belgrade*, оригиналан назив града или *adminName* ће бити Београд. Приликом претраге није битно да ли уносимо *Belgrade* или Београд, биће приказан исти град.

```

QueryBuilder queryBuilderGeo = QueryBuilders.geoDistanceQuery("geoPoint")
    .distance(20, DistanceUnit.KILOMETERS)
    .point(new GeoPoint(resultDataCity.getGeoPoint().getLat(),resultDataCity.getGeoPoint().getLon()));
QueryBuilder queryBuilderApproved = QueryBuilders.matchQuery("approved",true);
QueryBuilder queryBuilderOnlyStartDate = QueryBuilders.rangeQuery("startDate")
    .gte(geoAndTimeQuery.getStartDate()).lte(geoAndTimeQuery.getEndDate());
QueryBuilder queryBuilderOnlyStartDateQuery = QueryBuilders.boolQuery().must(queryBuilderGeo)
    .must(queryBuilderApproved).must(queryBuilderOnlyStartDate);

QueryBuilder queryBuilderOnlyEndDate = QueryBuilders.rangeQuery("endDate")
    .gte(geoAndTimeQuery.getStartDate()).lte(geoAndTimeQuery.getEndDate());
QueryBuilder queryBuilderOnlyEndDateQuery = QueryBuilders.boolQuery().must(queryBuilderGeo)
    .must(queryBuilderApproved).must(queryBuilderOnlyEndDate);

QueryBuilder queryBuilderStartDateBefore = QueryBuilders.rangeQuery("startDate")
    .lte(geoAndTimeQuery.getStartDate());
QueryBuilder queryBuilderEndDateAfter = QueryBuilders.rangeQuery("endDate")
    .gte(geoAndTimeQuery.getEndDate());
QueryBuilder queryBuilderDateBeforeAndAfter = QueryBuilders.boolQuery().must(queryBuilderGeo)
    .must(queryBuilderApproved).must(queryBuilderStartDateBefore).must(queryBuilderEndDateAfter);
QueryBuilder dateNullQuery = QueryBuilders.boolQuery().mustNot(QueryBuilders.existsQuery("startDate"))
    .must(queryBuilderGeo).must(queryBuilderApproved);
QueryBuilder finalQuery = QueryBuilders.boolQuery().should(queryBuilderOnlyStartDateQuery)
    .should(dateNullQuery).should(queryBuilderOnlyEndDateQuery).should(queryBuilderDateBeforeAndAfter);
if(geoAndTimeQuery.getCategory()!=null){
    finalQuery = QueryBuilders.boolQuery().must(finalQuery)
        .must(QueryBuilders.matchQuery("category",geoAndTimeQuery.getCategory()));
}

```

Слика 3 Приказана је имплементација упита

Финални упит је помоћу *sourceBuilder* прослеђен *restHighLevelClient* који враћа резултате претраге. Резултати претраге су враћени као погоци који се помоћу мапера мапирају на објекат који ће бити прослеђен клијентској страни. *SourceBuilder* је проширен да врати 100 погодака претраге. На слици 4 приказана је имплементација за добијање погодака и њихово мапирање.

```

SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
SearchRequest searchRequest = new SearchRequest("event");
ObjectMapper objectMapper = new ObjectMapper();
objectMapper.configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES,false);

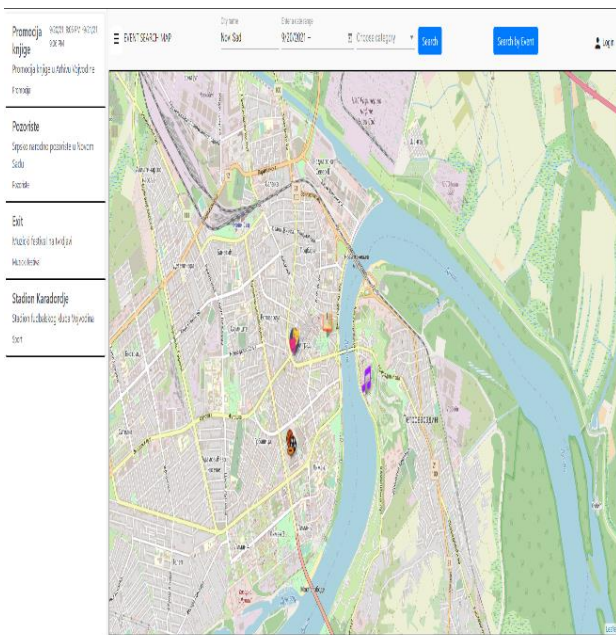
searchSourceBuilder.query(queryBuilderBoolQuery);
searchRequest.source(searchSourceBuilder);
searchSourceBuilder.size(100);

SearchResponse searchResponse = restHighLevelClient.search(searchRequest, RequestOptions.DEFAULT);
SearchHit[] hits = searchResponse.getHits().getHits();
for(SearchHit searchHit: hits){
    ResultDataEvent resultDataEvent = objectMapper
        .readValue(searchHit.getSourceAsString(),ResultDataEvent.class);
    if(resultDataEvent.getStartDate()!=null){
        String date = resultDataEvent.getStartDate().toString();
        if(resultDataEvent.getEndDate()!= null){
            date+=" - " + resultDataEvent.getEndDate().toString();
        }
        resultDataEvent.setDate(date);
    }
    resultDataEvents.add(resultDataEvent);
}
}

```

Слика 4 Имплементација за добијање погодака

На слици 5 приказана је мапа са догађајима који представљају резултат претраге за Град Нови Сад са дефинисаним почетним датумом.



Слика 5 Мапа са догађајима

6. ЗАКЉУЧАК

У раду је описана веб апликација за претраживање догађаја на основу текстуалне, геопросторне и временске претраге над *Elasticsearch* сервером. Представљене су поједине теоријске основе као што су дигитални документ и објашњен је сам појам проналажења информација. Описане су технологије које су коришћене приликом имплементације веб апликације, као што су Спринг и Ангулар, са посебним акцентом на *Elasticsearch* и начин функционисања геопросторне и временске претраге.

Дат је опис система, његова архитектура и случајеви коришћења, а затим је објашњена сама имплементација веб апликације. Приказано је на који начин су имплементирани поједине функционалности веб апликације као што су упити ка *Elasticsearch* серверу, индексирање градова и догађаја и приказивање догађаја на мапи. Приказано је коришћење апликације пропраћено сликама екрана.

Циљ овог рада био је да се направи апликација која ће олакшати претраживање и сам преглед догађаја који се дешавају у граду. Такође и да олакша промоцију организаторима догађаја, тако што ће се улоговати на систем и креирати свој догађај који ће се приказивати на мапи у зависности од претраге. Приказ рада геопросторне претраге, временске претраге, као и текстуалне претраге била је главна тема овог рада.

Апликацију је могуће проширити тако да се геопросторна претрага не односи само на градове, већ и на улице или чак на неке хотеле или места од туристичког значаја. И ако је у апликацији обухваћен велики део категорија за догађаје, постоји могућност за њихово додавање. Такође упите претраге је могуће проширити додавањем још опција претраге као и додавањем претпроцесирања текста називима и описима догађаја.

7. ЛИТЕРАТУРА

- [1] Д.Ивановић, Б. Милосављевић. Управљање дигиталним документима. Фатултетт техничких наука, 2015.
- [2] Спринг радни оквир. <https://spring.io/>. Посећено: 22.08.2021
- [3] *Search your Data- Elasticsearch*, <https://www.elastic.co/guide/en/elasticsearch/reference/current/search-your-data.html> Посећено: 05.09.2021
- [4] *Geo queries* <https://www.elastic.co/guide/en/elasticsearch/reference/current/geo-queries.html> Посећено: 05.09.2021
- [5] *Range query* <https://www.elastic.co/guide/en/elasticsearch/reference/6.8/query-dsl-range-query.html> Посећено 06.09.2021.

Кратка биографија:



Александар Јеремич рођен је 17.05. 1996. у Нишу. Основну школу „Михајло Пупин“ у Земуну завршио је 2011. године. Уписује Земунску гимназију, природно-математички смер, коју завршава 2015. године. Исте године уписује Факултет техничких наука у Новом Саду, одсек рачунарство и аутоматика. Године 2018. уписао се на смер Примењене рачунарске науке и информатика, а 2019. године усмерава се на модул Софтверско инжењерство. На мастер академске студије Факултета техничких наука уписује се школске 2019/20 на студијском програму рачунарство и аутоматика. Мастер рад одбранио је у октобру 2021.