

# E-Theses

## Specifikacija aplikacije

Aplikacije E-Theses predstavlja platformu za upravljanje završnim radovima. Postoje dva tipa korisnika, administratori i obični korisnici. Administratori imaju potpunu kontrolu nad svim entitetima sistema (profili korisnika, radovi, komentari, tagovi, smerovi, nivoi studija...), dok obični korisnici imaju kontrolu samo nad svojim profilom. Obični korisnici mogu preko platforme pretraživati radove svih korisnika po raznim kriterijumima, preuzimati radove, komentarisati radove... Administratori sa druge strane mogu kačiti radove, brisati radove, brisati komentare i tagove nad radovima, upravljati drugim entitetima koji su povezani sa radovima kao što su profili korisnika, smerovi, predmeti, oblasti studija...

Back-end aplikacije implementiran je kao REST Web servis upotrebom Java programskog jezika i Spring framework-a, dok je front-end deo aplikacije implementiran upotrebom JavaScript jezika i Angular framework-a.

REST Web servis implementiran je kako bi u budućnosti mogle da se rade integracije sa brojnim drugim aplikacijama za sličnu namenu kao što je Moodle, kao i zbog razdvojenosti front-end strane od back-end strane, odnosno web servisa. Razdvojenost front-end i back-end omogućava nezavisno verzionisanje, build-ovanje i deploy-ovanje front-end i back-end aplikacija. Takođe omogućava implementiranje i web front-end aplikacije u nekom od JavaScript mvc framework-a i mobilnih aplikacija na bilo kojoj platformi.

## Izabrane tehnologije

Za implementaciju web-servisa izabrana je REST arhitektura iz nekoliko razloga. Prvenstveno REST je veoma jednostavan za razumevanje, implementiranje i dokumentovanje, pogotovo ukoliko se koristi HTTP protokol i osnovne CRUD operacije. REST veoma dobro koristi protok mreže za razliku od SOAP protokola. REST je dizajniran da ne prati stanje, tako da je omogućeno keširanje a samim tim i mnogo bolje performanse. REST podržava brojne formate poruka među kojim i JSON, što znači mnogo bolju podršku browser klijentima za razliku od SOAP-a koji koristi XML poruke. JSON format omogućava veoma jednostavno konzumiranje web servisa od strane JavaScript klijentskih aplikacija.

REST web servis za pretragu završnih radova implementiran je u Java programskom jeziku, upotrebom Spring Framework-a. Spring je izabran jer predstavlja najpopularniji Java web framework danas. Spring pruža veoma dobru podršku za implementiranje REST web servisa, zaštitu web aplikacija, implementaciju DAO (Data Access Object) sloja nad bazom podataka, kao i za brojne druge stvari.

DAO sloj implementira se kreiranjem jednostavnog interfejsa koji se vezuje za jedan od domenskih objekata, a zatim spring pruža implementaciju za preko 15 operacija nad tim domenskim objektom. Metode Java klase uz nekoliko anotacija postaju REST web servisi a Java objekti JSON ili XML poruke koje se razmenjuju putem HTTP protokola.

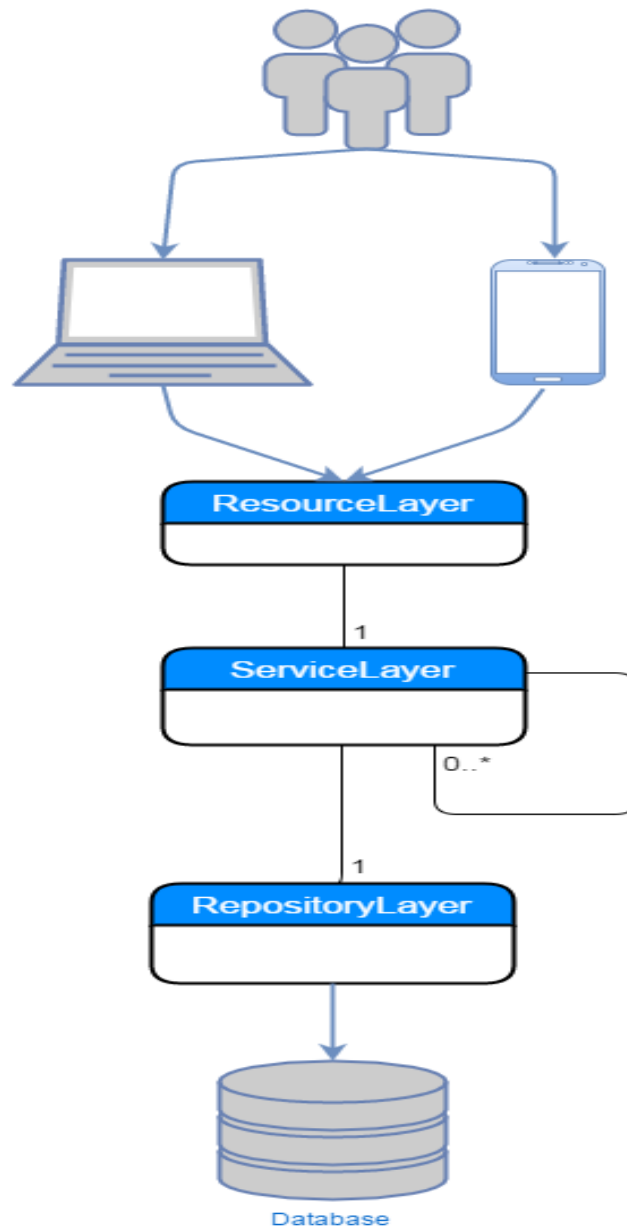
# Implementacija

## Arhitektura

Arhitektura web servisa sastoji se iz tri sloja:

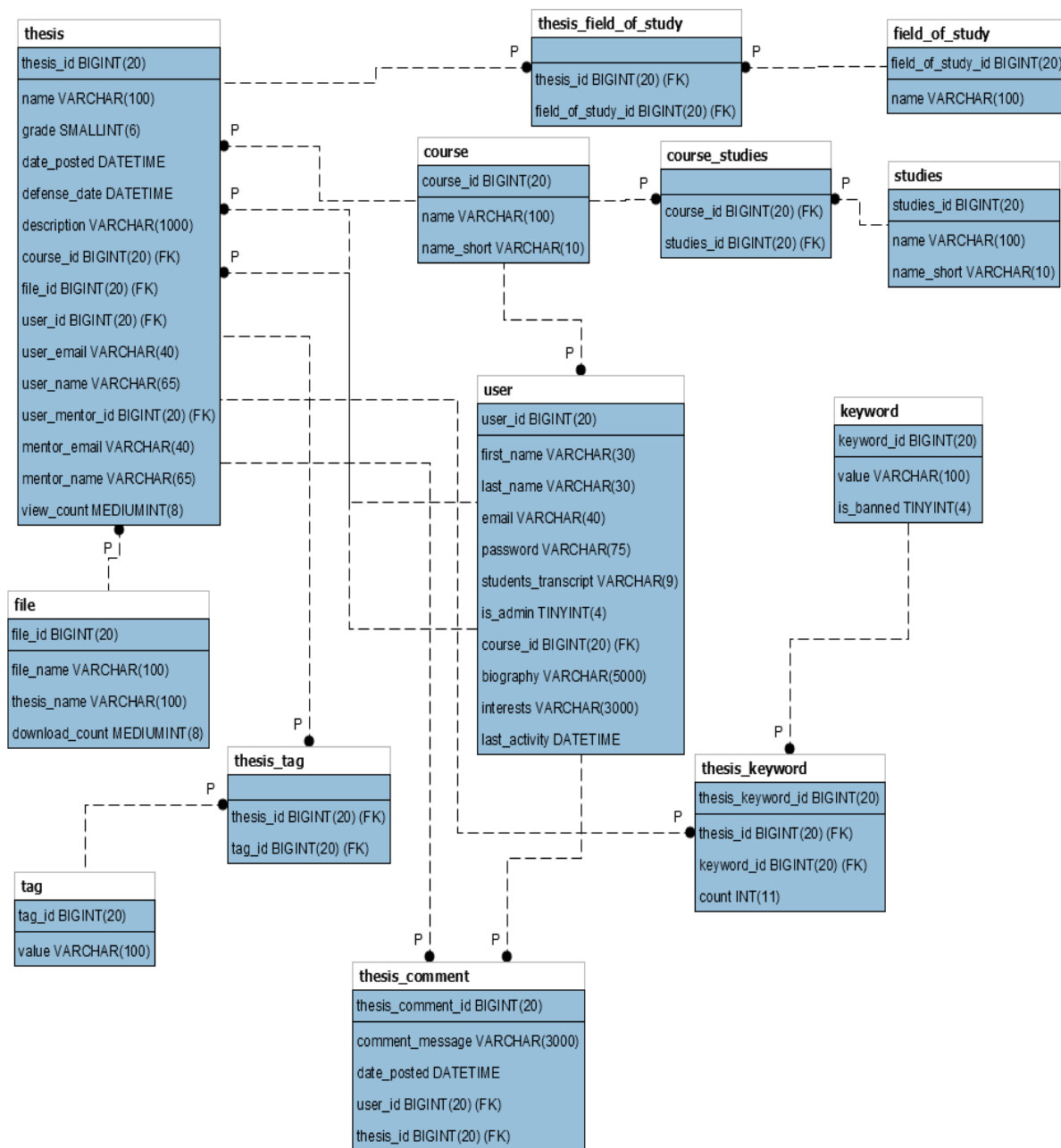
1. Sloj repozitorijuma zadužen za komunikaciju sa bazom odnosno za obezbeđivanje osnovnih operacija čuvanja, izmene, brisanja i čitanja entiteta.
2. Servisni sloj zadužen za poslovnu logiku. Postoji klasa za svaki jak entitet(domenski objekat) koja sadrži poslovnu logiku tog entiteta, koristi funkcionalnosti sloja repozitorijuma tog entiteta kao i servisnih slojeva drugih entiteta sa kojim je taj entitet vezan.
3. Resursni sloj, koji izlaže funkcionalnosti servisnog sloja u vidu REST web servisa. Takođe postoji po jedana klasa ovog sloja za svaki jak entitet.

Na slici broj 1 prikazana je arhitektura web servisa.



Slika 1: Arhitektura web servisa

Na slici broj 2 prikazan je model baze podataka(IDEF1X dijagram).



Slika 2: Model baze – IDEF1X dijagram

Na slici 2 vide se svi entiteti koje održava Sistem za pretragu završnih radova kao i veze između njih. Glavni entitet zbog kojeg sistem postoji je Rad(Thesis). Za rad se vezuje Korisnik(User) kao korisnik čiji je rad u pitanju, kao i korisnik koji je bio mentor za taj rad. Entitet fajl koji prstavlja fajl okečen uz taj rad, tagovi rada, komentari rada, nivo studija i smer na kojem je rad rađen.

### Primer koda i uputstvo za programere

Slede delovi koda sva tri sloja vezani za entitet Rad(Thesis), kao i uputstvo za izmenu i/ili dodavanje svakog sloja.

## Sloj repozitorijuma

Jedna od velikih prednosti Spring framework-a je uklanjanje nepohodnosti pisanja „boilerplate“ koda, u ovom slučaju manualnog pisanja Repository(DAO) klasa za perzistiranje objekata. Sve što je neophodno uraditi je naslediti neku od Spring-ovih Repository klasa i proslediti joj kao genričke parametre model klasu nad kojom se implementiraju CRUD operacije(Thesis) i tip id polja model klase(Long). Spring na osnovu ovoga pravi preko 15 metoda za perzistenciju objekata: findAll(), save(modelObjekat), delete(modelObjekat)... Ukoliko je neophodna neka od složenijih metoda kao što je pretraga svih entiteta po nekom polju kao što je za rad bila neophodna pretraga po nazivu rada dovoljno je samo interfejs dopuniti sa potpisom metode odgovarajućeg naziva i spring će obezbediti implementaciju. U našem primeru findByThesisName(String name). Ukoliko su neophodne još složenije metode kao u našem slučaju rada zbog napredne pretrage, moguće je u ovkiru anotacije @Query napisati custom OQL(Object query language) upit.

Za ovaj primer korišćene su automatski implementirane metode, metode dobijene potpisom odgovarajućeg naziva, kao i metode sa specifičnim upitom. Većina ostalih Repository klasa koristi samo automatski implementirane metode i eventualno metode dobijene potpisom.

```
public interface ThesisRepository extends PagingAndSortingRepository<Thesis, Long> {

    public List<Thesis> findAll();
    public List<Thesis> findByUser(User user);
    public Thesis findByName(String name);

    @Query("SELECT distinct t from Thesis t where t.name like ?1 and ?3 <= ('"
        + "SELECT count(tag2.id) from Thesis t2 left join t2.tags tag2 where t2.id=t.id and tag2.value in(?2)"
        + ") and ?5 <= ('"
        + "SELECT count(fos.id) from Thesis t3 left join t3.fieldOfStudies fos where t3.id=t.id and fos.name in(?4)"
        + ")")
    public Page<Thesis> findByNameLikeTagsAndFieldsPagable(Pageable pageRequest, String thesisName, List<String> tagValues, Long tagsMatchLimit, List<String> fieldValues, Long fieldsMatchLimit);

    @Query("SELECT distinct t from Thesis t where t.name like ?1 and t.course.name = ?4 and ?3 <= ('"
        + "SELECT count(tag2.id) from Thesis t2 left join t2.tags tag2 where t2.id=t.id and tag2.value in(?2)"
        + ") and ?6 <= ('"
        + "SELECT count(fos.id) from Thesis t3 left join t3.fieldOfStudies fos where t3.id=t.id and fos.name in(?5)"
        + ")")
    public Page<Thesis> findByNameLikeTagsFieldsAndCoursePagable(Pageable pageRequest, String thesisName, List<String> tagValues, Long matchLimit, String courseName, List<String> fieldValues, Long fieldsMatchLimit);

    @Query("SELECT distinct t from Thesis t join t.course.studies s where t.name like ?1 and s.name = ?4 and ?3 <= ('"
        + "SELECT count(tag2.id) from Thesis t2 left join t2.tags tag2 where t2.id=t.id and tag2.value in(?2)"
        + ") and ?6 <= ('"
        + "SELECT count(fos.id) from Thesis t3 left join t3.fieldOfStudies fos where t3.id=t.id and fos.name in(?5)"
        + ")")
    public Page<Thesis> findByNameLikeTagsFieldsAndStudiesPagable(Pageable pageRequest, String thesisName, List<String> tagValues, Long matchLimit, String studiesName, List<String> fieldValues, Long fieldsMatchLimit);
}
```

Slika 3: ThesisRepository klasa

Pored repository klase neophodno je napraviti i odgovarajuću model klasu sa odgovarajućim anotacijama i geterima i seterima(Java bean). Na slici 4 prikazana je model klasa za objekat Rad(Thesis)

```

@Entity
@Table(name = "thesis")
public class Thesis {

    @Id
    @Column(name = "thesis_id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    protected Long id;

    @Column(name = "name")
    protected String name;

    @Column(name="grade")
    protected Integer grade;

    @Column(name="date_posted")
    protected Date datePosted;

    @Column(name="defense_date")
    protected Date defenseDate;

    @Column(name="description")
    protected String description;

    @ManyToOne
    @JoinColumn(name = "course_id")
    protected Course course;

    @ManyToOne
    @JoinColumn(name = "file_id")
    protected TFile file;

    @OneToMany(mappedBy = "thesis", fetch = FetchType.EAGER)
    protected Set<ThesisComment> comments = new HashSet<ThesisComment>();

    @ManyToMany(fetch = FetchType.EAGER)
    @JoinTable(name = "thesis_tag", joinColumns = { @JoinColumn(name = "thesis_id") }, inverseJoinColumns = { @JoinColumn(name = "tag_id") })
    protected Set<Tag> tags = new HashSet<Tag>();

    @ManyToMany(fetch = FetchType.EAGER)
    @JoinTable(name = "thesis_field_of_study", joinColumns = { @JoinColumn(name = "thesis_id") }, inverseJoinColumns = { @JoinColumn(name = "field_of_study_id") })
    protected Set<FieldOfStudy> fieldOfStudies = new HashSet<>();

    @OneToMany(mappedBy = "thesis", fetch = FetchType.EAGER, cascade=CascadeType.ALL,orphanRemoval=true)
    private Set<ThesisKeyword> thesisKeywords = new HashSet<ThesisKeyword>();

    @ManyToOne
    @JoinColumn(name = "user_id")
    protected User user;
}

```

Slika 4 : thesis klasa

Dakle kako bi se dodao novi entitet i njegov persistentni sloj neophodno je napraviti model klasu (paket rs.elab.pzr.core.model), sa odgovarajućim anotacijama, i napraviti klasu koja implementira neki od Spring-ovih Repository interfejsa (paket rs.elab.pzr.core.repository). Takođe neophodno je dodati tabelu u bazi.

Za održavanje šeme korišćena je Spring integracija za Liquibase biblioteku za praćenje, održavanje i menjanje šeme baze:

<http://www.liquibase.org/documentation/spring.html>

Dakle cela šema se održava u okviru jednog falla:  
 \src\main\resources\context\db\db.changelog.xml.

Kreiranje tabele thesis prikazano je na slici 5.

```

<createTable tableName="thesis">
  <column name="thesis_id" type="bigint" autoIncrement="true">
    <constraints primaryKey="true" />
  </column>
  <column name="name" type="varchar(100)">
    <constraints nullable="false" unique="true" />
  </column>
  <column name="grade" type="smallint" />
  <column name="date_posted" type="datetime">
    <constraints nullable="false" />
  </column>
  <column name="defense_date" type="datetime" />
  <column name="description" type="varchar(1000)" />
  <column name="course_id" type="bigint" />
  <column name="file_id" type="bigint" />
  <column name="user_id" type="bigint">
    <constraints unique="false" />
  </column>
  <column name="user_email" type="varchar(40)" />
  <column name="user_name" type="varchar(65)" />
  <column name="user_mentor_id" type="bigint" />
  <column name="mentor_email" type="varchar(40)" />
  <column name="mentor_name" type="varchar(65)" />
  <column name="view_count" type="mediumint unsigned" />
</createTable>
<addForeignKeyConstraint baseColumnNames="course_id"
  baseTableName="thesis" constraintName="fk_thesis_course" onDelete="SET NULL"
  referencedColumnNames="course_id" referencedTableName="course" />
<addForeignKeyConstraint baseColumnNames="user_id"
  baseTableName="thesis" constraintName="fk_thesis_user" onDelete="SET NULL"
  referencedColumnNames="user_id" referencedTableName="user" />
<addForeignKeyConstraint baseColumnNames="user_mentor_id"
  baseTableName="thesis" constraintName="fk_thesis_mentor_user"
  onDelete="SET NULL" referencedColumnNames="user_id"
  referencedTableName="user" />
<addForeignKeyConstraint baseColumnNames="file_id"
  baseTableName="thesis" constraintName="fk_thesis_file" onDelete="SET NULL"
  referencedColumnNames="file_id" referencedTableName="file" />

```

Slika 5: db.changelog.xml.

### Servisni sloj

Servisni sloj sadrži poslovnu logiku entiteta kao što su naprimer ograničenja nad tim entitetom kako sustrukturalno tako i vrednosna, pravila za povezane i slabe objekte tog entiteta, održavanje fajlova kao u slučaju rada...

Servisne klase koriste funkcionalnosti prethodno objašnjenih klasa repozitorijuma kao i funkcionalnosti drugih servisnih klasa sa kojima je entitet te servisne klase povezan.

Na slici 6 prikazan je deo koda sevisne klase entiteta Rad(Thesis).

```

@Override
public void removeThesis(Long thesisId) {
    Thesis thesis = thesisRepository.findOne(thesisId);
    if (thesis == null) {
        throw new IllegalArgumentException("Rad sa id-em " + thesisId
            + " ne postoji u bazi!");
    }
    thesisRepository.delete(thesis);
}

@Override
public TFile addFile(Long thesisId, MultipartFile file) {
    if (file.isEmpty()) {
        throw new IllegalArgumentException("Fajl je prazan!");
    }
    Thesis thesis = getThesis(thesisId);
    if (thesis == null) {
        throw new IllegalArgumentException("Rad sa id-em " + thesisId
            + " ne postoji u bazi!");
    }
    TFile existingTFile = thesis.getFile();
    if (existingTFile != null) {
        throw new IllegalArgumentException("Rad " + thesis.getName()
            + " već ima upload-ovan fajl "
            + existingTFile.getFileName());
    }
    if (fileRepository.findByFileName(file.getOriginalFilename()) != null) {
        throw new IllegalArgumentException("Fajl sa imenom "
            + file.getOriginalFilename()
            + " već postoji. Molimo izaberite drugo ime fajla.");
    }
    TFile tFile = new TFile();
    tFile.setThesisName(thesis.getName());
    tFile.setFileName(file.getOriginalFilename());
    thesis.setFile(tFile);
    logger.debug(tFile.getFileName());
    try {
        byte[] bytes = file.getBytes();
        BufferedOutputStream stream = new BufferedOutputStream(
            new FileOutputStream(thesisFilesFolder
                + tFile.getFileName()));
        stream.write(bytes);
        stream.close();
    } catch (Exception e) {
        throw new PzrRuntimeException("Fajl nije uspešno postavljen: "
            + e.getMessage());
    }
    fileRepository.save(tFile);
    thesisRepository.save(thesis);
    return tFile;
}

@Override
public Set<TFile> getAllFileRecords() {
    return fileRepository.findAll();
}

```

Slika 6: ThesisServiceImpl klasa

Dakle da bi se dodala nova servisna (paket rs.elab.pzr.core.service) klasa neophodno je napraviti interfejs servisne klase (zapravo samo dobra praksa) zatim implementirati taj

interfejs i deklarisanе metode kao na slici 6. Dodati druge servisne klase kao i klasu repozitorijuma koje ova servisna klasa koristi kao attribute te klase i obezbediti getere i setere za te klase(ne instancirati te klase!!! – Spring Dependency Injection) a zatim kroz xml fajl za servisni sloj(/src/main/resources/context/core/core-services-context.xml) kreirati Java objekat(Java bean) sa odgovarajućim zavisnostima(repository i druge servisne klase) kao na slici 7:

```
<!-- authentication related service -->
<beans:bean id="authenticationService" class="rs.fon.elab.pzr.core.service.authentication.AuthenticationServiceImpl">
  <beans:property name="ticketService" ref="ticketService"></beans:property>
  <beans:property name="userRepository" ref="userRepository"></beans:property>
</beans:bean>
<beans:bean id="ticketService" class="rs.fon.elab.pzr.core.service.authentication.TicketService">
  <beans:property name="ticketCash" ref="ticketCash"></beans:property>
</beans:bean>
<beans:bean id="customUserDetailsService" class="rs.fon.elab.pzr.core.service.authentication.CustomUserDetailService">
  <beans:property name="userRepository" ref="userRepository"></beans:property>
</beans:bean>
<beans:bean id="ticketCash" class="rs.fon.elab.pzr.core.model.ticket.InMemoryTicketCash" scope="singleton">
</beans:bean>

<!-- rest services -->
<beans:bean id="courseService" class="rs.fon.elab.pzr.core.service.CourseServiceImpl">
  <beans:property name="courseRepository" ref="courseRepository"></beans:property>
</beans:bean>
<beans:bean id="studiesService" class="rs.fon.elab.pzr.core.service.StudiesServiceImpl">
  <beans:property name="studiesRepository" ref="studiesRepository"></beans:property>
</beans:bean>
<beans:bean id="keywordService" class="rs.fon.elab.pzr.core.service.KeywordServiceImpl">
  <beans:property name="keywordRepository" ref="keywordRepository"></beans:property>
  <beans:property name="thesisService" ref="thesisService"></beans:property>
</beans:bean>
<beans:bean id="userService" class="rs.fon.elab.pzr.core.service.UserServiceImpl">
  <beans:property name="thesisRepository" ref="thesisRepository"></beans:property>
  <beans:property name="userRepository" ref="userRepository"></beans:property>
</beans:bean>
<beans:bean id="thesisService" class="rs.fon.elab.pzr.core.service.ThesisServiceImpl">
  <beans:property name="thesisRepository" ref="thesisRepository"></beans:property>
  <beans:property name="commentRepository" ref="commentRepository"></beans:property>
  <beans:property name="fileRepository" ref="fileRepository"></beans:property>
  <beans:property name="userRepository" ref="userRepository"></beans:property>
  <beans:property name="userService" ref="userService"></beans:property>
  <beans:property name="thesisFilesFolder" value="${location.thesis.files}"></beans:property>
</beans:bean>

<beans:bean id="tagService" class="rs.fon.elab.pzr.core.service.TagServiceImpl">
  <beans:property name="tagRepository" ref="tagRepository"></beans:property>
</beans:bean>
<beans:bean id="fieldOfStudyService" class="rs.fon.elab.pzr.core.service.FieldOfStudyServiceImpl">
  <beans:property name="fieldOfStudyRepository" ref="fieldOfStudyRepository"></beans:property>
</beans:bean>
```

Slika 7: core-services-context.xml

### Resursni sloj

Kao što je već bilo reči resursni sloj izlaže funkcije servisnog sloja kao REST servise drugim aplikacijama. Takođe veoma velika prednost Spring-a je izuzetna lakoća kreiranja REST servisa. Neophodno je samo kreirati klasu i uz par anotacija pretvoriti njene metode u REST end-point-e. Takođe spring vrši konverziju(zapravo to je funkcionalnost drugih biblioteka za konverziju Java objekata u json format i obrnuto za koje je spring obezbedio integraciju, u našem slučaju jackson biblioteka) java objekata u json poruke koje se prosledjuju kao parametri web servisu i koje web servis vraća kao rezultat izvršene operacije.

Na slici 8 prikazan je deo klase ThesisResource:



```

@RequestMapping(method = RequestMethod.GET)
public @ResponseBody List<ThesisResponseLevel1> getThesises(
    @RequestParam(value = "userID", required = false) Long userID) {
    List<ThesisResponseLevel1> thesisResponseList = new ArrayList<ThesisResponseLevel1>();

    if (userID != null) {
        List<Thesis> userThesis = thesisService.getThesisByUserId(userID);
        for (Thesis thesis : userThesis) {
            thesisResponseList.add(RestFactory
                .createThesisResponseLevel1(thesis));
        }
        return thesisResponseList;
    }

    List<Thesis> thesisList = thesisService.getAllThesis();

    for (Thesis thesis : thesisList) {
        thesisResponseList.add(RestFactory
            .createThesisResponseLevel1(thesis));
    }
    return thesisResponseList;
}

@RequestMapping(method = RequestMethod.GET, value = "/advanced_search")
public @ResponseBody ThesisPageResponse advancedSearch(
    @RequestParam(value = "pageNumber", required = false) Integer pageNumber,
    @RequestParam(value = "pageSize", required = false) Integer pageSize,
    @RequestParam(value = "thesisName", required = false) String thesisName,
    @RequestParam(value = "tagValues", required = false) List<String> tagValues,
    @RequestParam(value = "matchLimit", required = false) Long matchLimit,
    @RequestParam(value = "courseName", required = false) String courseName,
    @RequestParam(value = "studiesName", required = false) String studiesName,
    @RequestParam(value = "sortField", required = false) String sortField,
    @RequestParam(value = "fieldValues", required = false) List<String> fieldValues,
    @RequestParam(value = "fieldMatchLimit", required = false) Long fieldMatchLimit,
    @RequestParam(value = "descriptionKeys", required = false) List<String> descriptionKeys,
    @RequestParam(value = "descriptionMatchLimit", required = false) Long descriptionMatchLimit) {
    Page<Thesis> thesisPage = thesisService.advancedSearch(pageNumber,
        pageSize, thesisName, tagValues, matchLimit, courseName,
        studiesName, sortField, fieldValues, fieldMatchLimit,
        descriptionKeys, descriptionMatchLimit);
    return RestFactory.createThesisPageResponse(thesisPage);
}

@RequestMapping(method = RequestMethod.GET, value =("/{thesisID}")
public @ResponseBody ThesisResponseLevel1 getThesis(
    @PathVariable("thesisID") Long thesisId) {
    Thesis thesis = thesisService.getThesis(thesisId);
    thesis.setViewCount(thesis.getViewCount() + 1);
    thesis = thesisService.updateThesis(thesis);
    return RestFactory.createThesisResponseLevel1(thesis);
}

@RequestMapping(method = RequestMethod.DELETE, value =("/{thesisID}")
public void removeThesis(@PathVariable("thesisID") Long thesisID) {
    thesisService.removeThesis(thesisID);
}

```

Slika 8: ThesisResource klasa

Dakle da bi se izložio web servis neophodno je napraviti prethodna dva sloja(sloj repozitorijuma i servisni sloj) a zatim napraviti klasu koja će da predstavlja web servise nad odgovarajućim entitetom za koje su napravljena prethodna dva sloja. Dodati servisne klase koje ova klasa koristi kao attribute te klase i obezbediti getere i setere (ne instancirati te klase!!! – Spring Dependency Injection) a zatim kroz xml fajl za resursni sloj(/src/main/resources/context/rest/rest-context.xml) kreirati java objekat(java bean) sa odgovarajućim zavisnostima kao na slici 9:

```

<!-- this is needed i order for @RequestBody and @ResponseBody to work spring
will look at classpath and find jackson jars and will convert java beans
into json @RestController annotation is not needed becaouse this context
is managed by SpringDispatcherServlet and beans declared here is searched
for every request for @RequestMapping and other annotations -->
<mvc:annotation-driven />

<bean id="authenticationResource" class="rs.fon.elab.pzr.rest.resources.AuthenticationResource">
    <property name="authenticationService" ref="authenticationService" />
    <property name="userService" ref="userService"></property>
    <property name="thesisService" ref="thesisService" />
</bean>
<bean id="courseResource" class="rs.fon.elab.pzr.rest.resources.CourseResource">
    <property name="courseService" ref="courseService" />
    <property name="studiesService" ref="studiesService" />
</bean>
<bean id="studiesResource" class="rs.fon.elab.pzr.rest.resources.StudiesResource">
    <property name="studiesService" ref="studiesService" />
</bean>
<bean id="userResource" class="rs.fon.elab.pzr.rest.resources.UserResource">
    <property name="userService" ref="userService" />
    <property name="courseService" ref="courseService" />
    <property name="thesisService" ref="thesisService" />
</bean>
<bean id="thesisResource" class="rs.fon.elab.pzr.rest.resources.ThesisResource">
    <property name="thesisService" ref="thesisService" />
    <property name="userService" ref="userService" />
    <property name="courseService" ref="courseService" />
    <property name="tagService" ref="tagService" />
    <property name="fieldOfStudyService" ref="fieldOfStudyService" />
    <property name="keywordService" ref="keywordService"/>
</bean>
<bean id="tagResource" class="rs.fon.elab.pzr.rest.resources.TagResource">
    <property name="tagService" ref="tagService" />
</bean>
<bean id="keywordResource" class="rs.fon.elab.pzr.rest.resources.KeywordResource">
    <property name="keywordService" ref="keywordService" />
</bean>
<bean id="fieldOfStudyResource" class="rs.fon.elab.pzr.rest.resources.FieldOfStudyResource">
    <property name="fieldOfStudyService" ref="fieldOfStudyService" />
</bean>
</beans>

```

Slika 9: rest-context.xml

## API Dokumentacija

### Resurs za autentikaciju

#### Zahtev

Metoda	URL	Opis
POST	zavrzni_radovi/webapi/authentication/	Obezbeđuje token na osnovu odgovarajućih kredencijala korisnika. Token može nositi dve vrste privilegija od kojih zavisi kojim resursima će korisnik moći da pristupi ukoliko u hederu ima obezbeđeni token: <ul style="list-style-type: none"> <li>- Korisnički token</li> <li>- Administratorski token</li> </ul>

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
json field	email	string
json field	password	string

Odgovor

Status	Response
200	<pre>{   "ticket": "inhq9g7n7t8mplr7et4nprvuj4" }</pre>
200	<pre>{   "error": "Bad credentials" }</pre>

Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/authentication/	Vraća informacije o korisniku koji je poslao zahtev

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string

Odgovor

Status	Response
200	<pre>{   "id": 2,   "firstName": "Petar",   "lastName": "Petrovic",   "email": "petarp@gmail.com",   "studentsTranscript": "11/11",   "course": {     "id": 1,     "name": "Informacioni sistemi i tehnologije",     "nameShort": "ISIT",     "studiesIDs": [       1,       2     ]   },   "theses": [ ],   "biography": null,   "interests": null,   "admin": false }</pre>
401	<pre>{   "error": "Must provide a valid ticket" }</pre>

	}
--	---

Preduslovi

#### Korisnički token

**ticket** header mora biti prisutan sa stringom koji predstavlja korisnički ili administratorski token.

#### Resurs za smerove

Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/courses/	Vraća sve smerove

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra

Odgovor

Status	Response
200	<pre>[   {     "id": 1,     "name": "Informacioni sistemi i tehnologije",     "nameShort": "ISIT",     "studies": [       {         "id": 2,         "name": "Diplomske studije",         "nameShort": "DS",         "courseIDs": [           1         ]       },       {         "id": 1,         "name": "Osnovne studije",         "nameShort": "OS",         "courseIDs": [           1,           2         ]       }     ]   },   {     "id": 2,     "name": "Menadžment",     "nameShort": "MNG",     "studies": [       {         "id": 1,         "name": "Osnovne studije",         "nameShort": "OS",</pre>

	<pre>         "courseIDs": [             1,             2         ]     },     {         "id": 3,         "name": "Programiranje",         "nameShort": "PRG",         "studies": [ ]     } ] </pre>
--	--

#### Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/courses/{courseID}	Vraća smer za zadati id

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
PATH PARAM	courseID	long

#### Odgovor

Status	Response
200	<pre> {     "id": 1,     "name": "Informacioni sistemi i tehnologije",     "nameShort": "ISIT",     "studies": [         {             "id": 2,             "name": "Diplomske studije",             "nameShort": "DS",             "courseIDs": [                 1             ]         },         {             "id": 1,             "name": "Osnovne studije",             "nameShort": "OS",             "courseIDs": [                 1,                 2             ]         }     ] } </pre>

## Zahtev

Metoda	URL	Opis
POST	zavrzni_radovi/webapi/courses	Kreira novi smer

## Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
JSON FIELD	name	string
JSON FIELD	nameShort	string
JSON FIELD	studiesIDs	List<Long>

## Odgovor

Status	Response
200	<pre>{   "id": 1,   "name": "Informacioni sistemi i tehnologije",   "nameShort": "ISIT",   "studies": [     {       "id": 2,       "name": "Diplomske studije",       "nameShort": "DS",       "courseIDs": [         1       ]     },     {       "id": 1,       "name": "Osnovne studije",       "nameShort": "OS",       "courseIDs": [         1,         2       ]     }   ] }</pre>
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

## Preduslovi

### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

## Zahtev

Metoda	URL	Opis
PUT	zavrsni_radovi/webapi/courses/{courseID}	Ažurira postojeći smer

## Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	courseID	long
JSON FIELD	name	string
JSON FIELD	nameShort	string
JSON FIELD	studiesIDs	List<Long>

## Odgovor

Status	Response
200	<pre>{   "id": 1,   "name": "Informacioni sistemi i tehnologije",   "nameShort": "ISIT",   "studies": [     {       "id": 2,       "name": "Diplomske studije",       "nameShort": "DS",       "courseIDs": [         1       ]     },     {       "id": 1,       "name": "Osnovne studije",       "nameShort": "OS",       "courseIDs": [         1,         2       ]     }   ] }</pre>
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

## Preduslovi

### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

#### Zahtev

Metoda	URL	Opis
DELETE	zavrzni_radovi/webapi/courses/{courseID}	Briše postojeći smer

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	courseID	long

#### Odgovor

Status	Response
200	
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

#### Preduslovi

### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

### Resurs za nivoe studija

#### Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/studies/	Vraća sve nivoe studija

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra

#### Odgovor

Status	Response
200	[



	<pre> {   "id": 1,   "name": "Osnovne studije",   "nameShort": "OS",   "courses": [     {       "id": 2,       "name": "Menadžment",       "nameShort": "MNG",       "studiesIDs": [         1       ]     },     {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     }   ] }, {   "id": 2,   "name": "Diplomske studije",   "nameShort": "DS",   "courses": [     {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     }   ] }, {   "id": 3,   "name": "Post diplomske studije",   "nameShort": "PDS",   "courses": [ ] } ] </pre>
--	--

#### Zahtev

Metoda	URL	Opis
GET	zavrsni_radovi/webapi/studies/{studiesID}	Vraća nivo studija za zadati id

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
PATH PARAM	studiesID	long

Odgovor

Status	Response
200	<pre>{   "id": 2,   "name": "Diplomske studije",   "nameShort": "DS",   "courses": [     {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     }   ] }</pre>

Zahtev

Metoda	URL	Opis
POST	zavrzni_radovi/webapi/studies	Kreira novi nivo studija

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
JSON FIELD	name	string
JSON FIELD	nameShort	string

Odgovor

Status	Response
200	<pre>{   "id": 2,   "name": "Diplomske studije",   "nameShort": "DS",   "courses": [     {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     }   ] }</pre>

	<pre>         ]       }     ]   } </pre>
404	<pre> {   "error": "poruka greške" } </pre>
401	
403	

## Preduslovi

### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

## Zahtev

Metoda	URL	Opis
PUT	zavrzni_radovi/webapi/studies/{studiesID}	Ažurira postojeći nivo studija

## Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	studiesID	long
JSON FIELD	name	string
JSON FIELD	nameShort	string

## Odgovor

Status	Response
200	<pre> {   "id": 2,   "name": "Diplomske studije",   "nameShort": "DS",   "courses": [     {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     }   ] } </pre>

404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

Preduslovi

**Administratorski token**

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

Zahtev

Metoda	URL	Opis
DELETE	zavrzni_radovi/webapi/studies/{studiesID}	Briše postojeći nivo studija

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	studiesID	long

Odgovor

Status	Response
200	
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

Preduslovi

**Administratorski token**

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

**Resurs za oblasti**

Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/fields-of-study/	Vraća sve oblasti

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra

Odgovor

Status	Response
200	<pre>[   {     "id": 2,     "name": "JAVA"   },   {     "id": 6,     "name": "NoSQL"   },   {     "id": 4,     "name": "Nuklearna fizika"   },   {     "id": 1,     "name": "SQL"   },   {     "id": 3,     "name": "SUBP"   } ]</pre>

Zahtev

Metoda	URL	Opis
POST	zavrsni_radovi/webapi/fields-of-study/	Kreira novu oblast

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
JSON FIELD	name	string

Odgovor

Status	Response
200	<pre>{   "id": 2,   "name": "JAVA" }</pre>
404	<pre>{   "error": "poruka greške" }</pre>

	}
401	
403	

Preduslovi

#### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

Zahtev

Metoda	URL	Opis
PUT	<code>zavrsni_radovi/webapi/fields-of-study/{fieldsOfStudyID}</code>	Ažurira postojeću oblast

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string
PATH PARAM	fieldsOfStudyID	long
JSON FIELD	name	string

Odgovor

Status	Response
200	<pre>{   "id": 2,   "name": "JAVA 8" }</pre>
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

Preduslovi

#### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

Zahtev

Metoda	URL	Opis
--------	-----	------

<b>DELETE</b>	zavrsni_radovi/webapi/fields-of-study/{fieldsOfStudyID}	Briše postojeći nivo studija
---------------	---	------------------------------

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	fieldsOfStudyID	long

Odgovor

Status	Response
200	
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

Preduslovi

**Administratorski token**

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

**Resurs za tagove**

Zahtev

Metoda	URL	Opis
<b>GET</b>	zavrsni_radovi/webapi/tags/	Vraća sve tagove

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra

Odgovor

Status	Response
200	<pre>[   {     "id": 2,     "value": "fastsearch"   },   {     "id": 1,     "value": "nosql"   },   { </pre>

	<pre>       "id": 5,       "value": "spring"     },     {       "id": 4,       "value": "springboot"     },     {       "id": 3,       "value": "springdata"     }   ] </pre>
--	---

#### Zahtev

Metoda	URL	Opis
<b>DELETE</b>	zavrsni_radovi/webapi/tags/{tagID}	Briše postojeći tag

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	tagID	long

#### Odgovor

Status	Response
200	
404	<pre> {   "error": "poruka greške" } </pre>
401	
403	

#### Preduslovi

##### Administratorski token

**ticket** header mora biti prisutan sa stringom koji predstavlja administratorski token.

#### Resurs za korisnike

##### Zahtev

Metoda	URL	Opis
<b>GET</b>	zavrsni_radovi/webapi/users/	Vraća sve korisnike



## Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra

## Odgovor

Status	Response
200	<pre>[   {     "id": 1,     "firstName": "user10",     "lastName": "user10 ",     "email": "user10@gmail.com",     "studentsTranscript": 11/11,     "course": null,     "theses": [       {         "id": 1,         "name": "Elastic Search",         "grade": "10",         "datePosted": 1461100269000,         "defenseDate": null,         "description": "Mnogo brza pretraga.",         "courseId": 1,         "userId": 1,         "userName": "",         "userEmail": "",         "viewCount": 2,         "mentorId": null,         "mentorName": "Petar Petrović",         "mentorEmail": "petar@gmail.com",         "tags": [           {             "id": 1,             "value": "#nosql"           },           {             "id": 2,             "value": "#fastsearch"           }         ],         "fieldsOfStudy": [           {             "id": 2,             "name": "JAVA"           }         ],         "commentIDs": [ ],         "file": null       }     ],     "biography": "Rodjen, zavrrio skolu...",     "interests": "Programiranje, sport...",     "admin": false   },   {     "id": 2,     "firstName": null,</pre>

	<pre> "lastName": null, "email": "user2@gmail.com", "studentsTranscript": null, "course": {   "id": 1,   "name": "Informacioni sistemi i tehnologije",   "nameShort": "ISIT",   "studiesIDs": [     1,     2   ] }, "theses": [ ], "biography": null, "interests": null, "admin": true } ] </pre>
--	---

#### Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/users/{userID}	Vraća korisnika za zadati id

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
PATH PARAM	userID	long

#### Odgovor

Status	Response
200	<pre> {   "id": 1,   "firstName": "user10",   "lastName": "user10 ",   "email": "user10@gmail.com",   "studentsTranscript": 11/11,   "course": null,   "theses": [     {       "id": 1,       "name": "Elastic Search",       "grade": "10",       "datePosted": 1461100269000,       "defenseDate": null,       "description": "Mnogo brza pretraga.",       "courseId": 1,       "userId": 1,       "userName": "",       "userEmail": "",       "viewCount": 2,       "mentorId": null,     }   ] } </pre>

	<pre> "mentorName": "Petar Petrović", "mentorEmail": "petar@gmail.com", "tags": [   {     "id": 1,     "value": "#nosql"   },   {     "id": 2,     "value": "#fastsearch"   } ], "fieldsOfStudy": [   {     "id": 2,     "name": "JAVA"   } ], "commentIDs": [ ], "file": null }, "biography": "Rodjen, završio skolu...", "interests": "Programiranje, sport...", "admin": false } </pre>
--	--

#### Zahtev

Metoda	URL	Opis
POST	završni_radovi/webapi/users	Kreira novog korisnika

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
JSON FIELD	email	string
JSON FIELD	password	string

#### Odgovor

Status	Response
200	<pre> {   "id": 4,   "firstName": null,   "lastName": null,   "email": "user2@gmail.com",   "studentsTranscript": null,   "course": null,   "theses": null,   "biography": null,   "interests": null,   "admin": false } </pre>

404	<pre>{   "error": "poruka greške" }</pre>

#### Zahtev

Metoda	URL	Opis
PUT	zavrzni_radovi/webapi/courses/{userID}	Ažurira postojećeg korisnika

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	Ticket	string
PATH PARAM	userID	long
JSON FIELD	firstName	string
JSON FIELD	lastName	string
JSON FIELD	studentsTranscript	string
JSON FIELD	courseName	string
JSON FIELD	biography	string
JSON FIELD	interests	string

#### Odgovor

Status	Response
200	<pre>{   "id": 4,   "firstName": "Jovana",   "lastName": "Miličić",   "email": "user2@gmail.com",   "studentsTranscript": "12/12",   "course": null,   "theses": null,   "biography": "3 yeras of experience...",   "interests": "Tenis, programiranje...",   "admin": false }</pre>
404	<pre>{   "error": "poruka greške" }</pre>
401	<pre>{</pre>

	<pre>"error": "Morate biti ulogovani kako bi menjali profil" }</pre>
--	--

Preduslovi

#### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token ili token korisnika čiji se profil menja.

Zahtev

Metoda	URL	Opis
DELETE	zavrzni_radovi/webapi/users/{userID}	Briše postojećeg korisnika

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string
PATH PARAM	userID	long

Odgovor

Status	Response
200	
404	<pre>{   "error": "poruka greške" }</pre>
401	
403	

Preduslovi

#### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

### Resurs za radove

Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/theses/	Vraća sve radove ili sve radove određenog korisnika ukoliko je prosleđen parametar userID

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
---------------	-----------------	-------------------------

QUERY PARAM	userID	long
-------------	--------	------

Odgovor

Status	Response
200	<pre>[   {     "id": 1,     "name": "Elastic Search",     "grade": null,     "datePosted": 1461100269000,     "defenseDate": null,     "description": "Mnogo brza pretraga.",     "course": {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     },     "user": {       "id": 1,       "firstName": null,       "lastName": null,       "email": "user10@gmail.com",       "studentsTranscript": null,       "courseId": null,       "thesisIDs": [         1       ],       "biography": null,       "interests": null,       "admin": true     },     "userName": "",     "userEmail": "",     "viewCount": 2,     "mentor": null,     "mentorName": "",     "mentorEmail": "",     "tags": [       {         "id": 2,         "value": "#fastsearch"       },       {         "id": 1,         "value": "#nosql"       }     ],     "fieldsOfStudy": [       {         "id": 2,         "name": "JAVA"       }     ],     "comments": [ ]   } ]</pre>

```
"file": null
},
{
  "id": 3,
  "name": "Spring Boot",
  "grade": null,
  "datePosted": 1461100368000,
  "defenseDate": null,
  "description": "",
  "course": null,
  "user": null,
  "userName": "",
  "userEmail": "",
  "viewCount": 0,
  "mentor": null,
  "mentorName": "",
  "mentorEmail": "",
  "tags": [
    {
      "id": 3,
      "value": "springdata"
    },
    {
      "id": 5,
      "value": "spring"
    },
    {
      "id": 4,
      "value": "springboot"
    }
  ],
  "fieldsOfStudy": [
    {
      "id": 1,
      "name": "SQL"
    },
    {
      "id": 2,
      "name": "JAVA"
    }
  ],
  "comments": [ ],
  "file": null
},
{
  "id": 4,
  "name": "PHP Microservices",
  "grade": null,
  "datePosted": 1461100436000,
  "defenseDate": null,
  "description": "",
  "course": null,
  "user": null,
  "userName": "",
  "userEmail": "",
  "viewCount": 0,
  "mentor": null,
  "mentorName": "",
  "mentorEmail": "",
  "tags": [ ],
```

	<pre> "fieldsOfStudy": [ ], "comments": [ ], "file": null } ] </pre>
--	--

#### Zahtev

Metoda	URL	Opis
GET	zavrsni_radovi/webapi/theses/{thesisID}	Vraća rad sa zadatim id-em

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
PATH PARAM	thesisID	long

#### Odgovor

Status	Response
200	<pre> {   "id": 3,   "name": "Spring Boot",   "grade": null,   "datePosted": 1461100368000,   "defenseDate": null,   "description": "Rad o spring boot-u",   "course": null,   "user": null,   "userName": "User1",   "userEmail": "user1@gmail.com",   "viewCount": 0,   "mentor": null,   "mentorName": "",   "mentorEmail": "",   "tags": [     {       "id": 3,       "value": "springdata"     },     {       "id": 5,       "value": "spring"     },     {       "id": 4,       "value": "springboot"     }   ],   "fieldsOfStudy": [     {       "id": 1,       "name": "SQL"     }   ] } </pre>



	<pre>         },         {             "id": 2,             "name": "JAVA"         }     ],     "comments": [ ],     "file": null } </pre>
--	--

## Zahtev

Metoda	URL	Opis
<b>GET</b>	zavrzni_radovi/webapi/theses/advanced_search	Vraća sve radove koji ispunjavaju prosleđene parametre pretarge. Nijedan od parametara nije obavezan. matchLimit je broj minimalan broj tagova koje rad mora da poseduje od prosleđenih(tagValues), fieldMatchLimit je minimalan broj oblasti koje rad mora da poseduje od prosleđenih(fieldValues), descriptionMatchLimit je minimalan broj ključnih reči opisa koje rad mora da poseduje od prosleđenih(descriptionKeys)

## Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
QUERY PARAM	pageNumber	Int
QUERY PARAM	pageSize	Int
QUERY PARAM	thesisName	String
QUERY PARAM	tagValues	List<String>
QUERY PARAM	matchLimit	Long
QUERY PARAM	courseName	String
QUERY PARAM	studiesName	String
QUERY PARAM	sortField	String
QUERY PARAM	fieldValues	List<String>
QUERY PARAM	fieldMatchLimit	Long
QUERY PARAM	descriptionKeys	List<String>
QUERY PARAM	descriptionMatchLimit	Long

## Odgovor

Status	Response
200	<pre>[   {     "id": 1,     "name": "Elastic Search",     "grade": null,     "datePosted": 1461100269000,     "defenseDate": null,     "description": "Mnogo brza pretraga.",     "course": {       "id": 1,       "name": "Informacioni sistemi i tehnologije",       "nameShort": "ISIT",       "studiesIDs": [         1,         2       ]     },     "user": {       "id": 1,       "firstName": null,       "lastName": null,       "email": "user10@gmail.com",       "studentsTranscript": null,       "courseId": null,       "thesisIDs": [         1       ],       "biography": null,       "interests": null,       "admin": true     },     "userName": "",     "userEmail": "",     "viewCount": 2,     "mentor": null,     "mentorName": "",     "mentorEmail": "",     "tags": [       {         "id": 2,         "value": "#fastsearch"       },       {         "id": 1,         "value": "#nosql"       }     ],     "fieldsOfStudy": [       {         "id": 2,         "name": "JAVA"       }     ],     "comments": [ ],     "file": null   },   {</pre>

	<pre>       "id": 3,       "name": "Spring Boot",       "grade": null,       "datePosted": 1461100368000,       "defenseDate": null,       "description": "",       "course": null,       "user": null,       "userName": "",       "userEmail": "",       "viewCount": 0,       "mentor": null,       "mentorName": "",       "mentorEmail": "",       "tags": [         {           "id": 3,           "value": "springdata"         },         {           "id": 5,           "value": "spring"         },         {           "id": 4,           "value": "springboot"         }       ],       "fieldsOfStudy": [         {           "id": 1,           "name": "SQL"         },         {           "id": 2,           "name": "JAVA"         }       ],       "comments": [ ],       "file": null     }   ] </pre>
--	--

#### Zahtev

Metoda	URL	Opis
POST	zavrsni_radovi/webapi/theses	Kreira novi rad

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string

JSON FIELD	name	string
JSON FIELD	grade	Int
JSON FIELD	defenseDate	Date
JSON FIELD	description	String
JSON FIELD	courseName	String
JSON FIELD	userId	Long
JSON FIELD	userName	String
JSON FIELD	userEmail	String
JSON FIELD	mentorId	Long
JSON FIELD	mentorName	String
JSON FIELD	mentorEmail	String
JSON FIELD	tags	List<String>
JSON FIELD	fieldsOfStudy	List<String>

Odgovor

Status	Response
200	<pre> {   "id": 1,   "name": "Elastic Search",   "grade": null,   "datePosted": 1461100269000,   "defenseDate": null,   "description": "Mnogo brza pretraga.",   "course": {     "id": 1,     "name": "Informacioni sistemi i tehnologije",     "nameShort": "ISIT",     "studiesIDs": [       1,       2     ]   },   "user": {     "id": 1,     "firstName": null,     "lastName": null,     "email": "user10@gmail.com",     "studentsTranscript": null,     "courseId": null,     "thesisIDs": [       1     ],     "biography": null, </pre>

	<pre>         "interests": null,         "admin": true     },     "userName": "",     "userEmail": "",     "viewCount": 2,     "mentor": null,     "mentorName": "",     "mentorEmail": "",     "tags": [         {             "id": 2,             "value": "#fastsearch"         },         {             "id": 1,             "value": "#nosql"         }     ],     "fieldsOfStudy": [         {             "id": 2,             "name": "JAVA"         }     ],     "comments": [ ],     "file": null }, </pre>
404	<pre> {     "error": "poruka greške" } </pre>
401	
403	

Preduslovi

#### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token

Zahtev

Metoda	URL	Opis
POST	zavrsni_radovi/webapi/theses/{thesisID}	Ažurira postojeći rad

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string
PATH PARAM	thesisID	long
JSON FIELD	grade	Int

JSON FIELD	defenseDate	Date
JSON FIELD	description	String
JSON FIELD	courseName	String
JSON FIELD	userId	Long
JSON FIELD	userName	String
JSON FIELD	userEmail	String
JSON FIELD	mentorId	Long
JSON FIELD	mentorName	String
JSON FIELD	mentorEmail	String
JSON FIELD	tags	List<String>
JSON FIELD	fieldsOfStudy	List<String>

Odgovor

Status	Response
200	<pre>{   "id": 1,   "name": "Elastic Search",   "grade": "10",   "datePosted": 1461100269000,   "defenseDate": 1461100269000,   "description": "Mnogo brza pretraga.",   "course": {     "id": 1,     "name": "Informacioni sistemi i tehnologije",     "nameShort": "ISIT",     "studiesIDs": [       1,       2     ]   },   "user": {     "id": 1,     "firstName": null,     "lastName": null,     "email": "user10@gmail.com",     "studentsTranscript": null,     "courseId": null,     "thesisIDs": [       1     ],     "biography": null,     "interests": null,     "admin": true   },   "userName": "",   "userEmail": "", }</pre>

	<pre> "viewCount": 2, "mentor": null, "mentorName": "Petar", "mentorEmail": "petar@gmail.com", "tags": [   {     "id": 2,     "value": "#fastsearch"   },   {     "id": 1,     "value": "#nosql"   } ], "fieldsOfStudy": [   {     "id": 2,     "name": "JAVA"   } ], "comments": [ ], "file": null }, </pre>
404	<pre> {   "error": "poruka greške" } </pre>
401	
403	

#### Preduslovi

##### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token

#### Zahtev

Metoda	URL	Opis
<b>DELETE</b>	zavrsni_radovi/webapi/theses/{thesisID}	Briše postojeći rad

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string
PATH PARAM	thesisID	long

#### Odgovor

Status	Response
200	

404	{ "error": "poruka greške" }
401	
403	

Preduslovi

#### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token.

Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/theses/{thesisID}/comments	Vraća sve komentare rada čiji je id prosledje kao deo url-a

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
PATH PARAM	thesisID	long

Odgovor

Status	Response
200	<pre>{   "id": 3,   "message": "nije loše",   "datePosted": 1466277503000,   "author": {     "id": 2,     "firstName": null,     "lastName": null,     "email": " user1@gmail.com",     "studentsTranscript": null,     "courseId": 1,     "thesisIDs": [ ],     "biography": null,     "interests": null,     "admin": true   },   "thesis": {     "id": 1,     "name": "Elastic Search",     "grade": null,     "datePosted": 1461100269000,     "defenseDate": null,     "description": "Mnogo brza pretraga.",     "courseId": 1,     "userId": 1,     "userName": "",     "userEmail": "",     "viewCount": 2,</pre>



```

"mentorId": null,
"mentorName": "",
"mentorEmail": "",
"tags": [
  {
    "id": 1,
    "value": "#nosql"
  },
  {
    "id": 2,
    "value": "#fastsearch"
  }
],
"fieldsOfStudy": [
  {
    "id": 2,
    "name": "JAVA"
  }
],
"commentIDs": [
  1,
  2,
  3
],
"file": null
}
},
{
  "id": 1,
  "message": "mnogo dobar rad",
  "datePosted": 1466277490000,
  "author": {
    "id": 2,
    "firstName": null,
    "lastName": null,
    "email": " user1@gmail.com",
    "studentsTranscript": null,
    "courseId": 1,
    "thesisIDs": [ ],
    "biography": null,
    "interests": null,
    "admin": true
  },
  "thesis": {
    "id": 1,
    "name": "Elastic Search",
    "grade": null,
    "datePosted": 1461100269000,
    "defenseDate": null,
    "description": "Mnogo brza pretraga.",
    "courseId": 1,
    "userId": 1,
    "userName": "",
    "userEmail": "",
    "viewCount": 2,
    "mentorId": null,
    "mentorName": "",
    "mentorEmail": "",
    "tags": [
      {

```

```

        "id": 1,
        "value": "#nosql"
    },
    {
        "id": 2,
        "value": "#fastsearch"
    }
],
"fieldsOfStudy": [
    {
        "id": 2,
        "name": "JAVA"
    }
],
"commentIDs": [
    1,
    2,
    3
],
"file": null
}
},
{
    "id": 2,
    "message": "savrsen rad",
    "datePosted": 1466277498000,
    "author": {
        "id": 2,
        "firstName": null,
        "lastName": null,
        "email": "user1@gmail.com",
        "studentsTranscript": null,
        "courseId": 1,
        "thesisIDs": [ ],
        "biography": null,
        "interests": null,
        "admin": true
    },
    "thesis": {
        "id": 1,
        "name": "Elastic Search",
        "grade": null,
        "datePosted": 1461100269000,
        "defenseDate": null,
        "description": "Mnogo brza pretraga.",
        "courseId": 1,
        "userId": 1,
        "userName": "",
        "userEmail": "",
        "viewCount": 2,
        "mentorId": null,
        "mentorName": "",
        "mentorEmail": "",
        "tags": [
            {
                "id": 1,
                "value": "#nosql"
            },
            {
                "id": 2,

```

	<pre>         "value": "#fastsearch"       }     ],     "fieldsOfStudy": [       {         "id": 2,         "name": "JAVA"       }     ],     "commentIDs": [       1,       2,       3     ],     "file": null   } ] </pre>
--	--

#### Zahtev

Metoda	URL	Opis
POST	zavrzni_radovi/webapi/theses/{thesisID}/comments	Dodaje komentar na rad čiji je id prosledjen kao deo url-a

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string
PATH PARAM	thesisID	long
JSON FIELD	Message	Sting

#### Odgovor

Status	Response
200	<pre> {   "id": 1,   "message": "mnogo dobar rad",   "datePosted": 1466277490000,   "author": {     "id": 2,     "firstName": null,     "lastName": null,     "email": "user1@gmail.com",     "studentsTranscript": null,     "courseId": 1,     "thesisIDs": [ ],     "biography": null,     "interests": null,     "admin": true   }, </pre>

	<pre> "thesis": {   "id": 1,   "name": "Elastic Search",   "grade": null,   "datePosted": 1461100269000,   "defenseDate": null,   "description": "Mnogo brza pretraga.",   "courseId": 1,   "userId": 1,   "userName": "",   "userEmail": "",   "viewCount": 2,   "mentorId": null,   "mentorName": "",   "mentorEmail": "",   "tags": [     {       "id": 1,       "value": "#nosql"     },     {       "id": 2,       "value": "#fastsearch"     }   ],   "fieldsOfStudy": [     {       "id": 2,       "name": "JAVA"     }   ],   "commentIDs": [     1,     2,     3   ],   "file": null } </pre>
404	<pre> {   "error": "poruka greške" } </pre>
401	

Preduslovi

**Administratorski token**

`ticket` header mora biti prisutan sa stringom koji predstavlja korisnički token.

Zahtev

Metoda	URL	Opis
DELETE	zavrsni_radovi/webapi/theses/comments/{commentID}	Briše komentar čiji je id prosledjen kao deo url-a

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	string
PATH PARAM	commentID	long

Odgovor

Status	Response
200	
401	<pre>{   "error": "morate biti ulogovani kako bi obrisali komentar" }</pre>

Preduslovi

#### Administratorski token

**ticket** header mora biti prisutan sa stringom koji predstavlja administratorski token ili token korisnika čiji se komentar briše.

Zahtev

Metoda	URL	Opis
GET	zavrzni_radovi/webapi/theses/files	Vraća informacije o upload-ovanim fajlovima svih radova

Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra

Odgovor

Status	Response
200	<pre>[   {     "id": 1,     "fileName": "ElasticSearch.txt ",     "thesisName": "Elastic Search",     "downloadCount": 10   },   {     "id": 2,     "fileName": "SpringBoot.txt ",     "thesisName": "Spring Boot",     "downloadCount": 25   } ]</pre>

#### Zahtev

Metoda	URL	Opis
GET	zavrsni_radovi/webapi/theses/files/{fileID}download	Skida fajl(datoteku) sa prosledjenim id-em

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
PATH PARAM	FileID	Long

#### Zahtev

Metoda	URL	Opis
DELETE	zavrsni_radovi/webapi/theses/files/{fileID}	Briše rekord fajla sa prosledjenim id-em, kao i datoteku vezanu sa fajl sa sistema

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
HEADER	ticket	String
PATH PARAM	fileID	Long

#### Odgovor

Status	Response
200	
401	
403	

#### Preduslovi

##### Administratorski token

`ticket` header mora biti prisutan sa stringom koji predstavlja administratorski token

#### Zahtev

Metoda	URL	Opis
POST	zavrsni_radovi/webapi/theses/{thesisId}/upload	Upload-uje fajl za rad čiji je id prosledjen kao de url-a

#### Parametri

Tip parametra	Naziv parametra	Tip vrednosti parametra
---------------	-----------------	-------------------------

HEADER	ticket	String
PATH PARAM	fileID	Long
multipart/form-data	file	.zip,.pdf.,.txt,.word,...

Odgovor

Status	Response
200	<pre>{   "id": 1,   "fileName": "ElasticSearch.txt ",   "thesisName": "Elastic Search",   "downloadCount": 0 }</pre>
401	
403	

Preduslovi

#### Administratorski token

**ticket** header mora biti prisutan sa stringom koji predstavlja administratorski token

#### HTTP status kodovi

U tabeli ispod navedeni su svi HTTP status kodovi korišćeni u web servisu za pretragu završnih radova.

**2XX** – Uspešan zahtev

**4XX** – Greška na strani klijenta

**5XX** – Greška na strani servera

Status Code	Description
200(OK)	Uspešan zahtev
400(Bad request)	Greška u prosleđenim parametrima
404(Not found)	Resurs nije pronađen
401(Authentication failure)	Token nije prosledjen u hederu prilikom pristupanja zaštićenom resursu
403(Forbidden)	Prosleđen korisnički token prilikom pristupanja resursu koji zahteva administratorski token
500(Internal Server Error)	Greška na serveru
503(Service Unavailable)	Traženi servis trenutno nedostupan

