

Prirodno – matematički fakultet

Univerzitet u Kragujevcu



Tema projekta:

- Studentska služba -

Team Leader: Dragana Anđelković

Kragujevac, 2020. godina

PROJEKTNA DOKUMENTACIJA

Spring Framework

Spring je najpopularniji okvir (eng. Framework) za razvoj Java enterprise veb aplikacija.

Spring je po dizajnu modularan. Nudi više od dvadeset projekata, koji obezbeđuju različite infrastrukture:

- Osnovni Spring okvir (engl. Spring framework)
- Okvir koji postavlja i podiže aplikaciju (engl. Spring Boot)
- Okvir za rad sa bazama podataka (engl. Spring Data)
- Okviri za sigurnost aplikacije (engl. Spring Security, Spring LDAP)
- Okviri za upravljanje porukama (engl. Spring Integration, Spring AMQP) i drugi.

Osnovni Spring okvir omogućava kreiranje skladišta objekata, korišćenjem dizajn uzorka Inverzija kontrole (engl. Inversion of control container).

Inversion of control (loc) – programerska tehnika koja obezbeđuje da svaka klasa u toku izvršenja aplikacije dobije sve objekte koji su joj neophodni.

Prednosti loc:

- Uklanjanje zavisnosti (dependency)
- Čistiji kod
- Ponovno korišćenje delova koda (Code reuse)

Dependency injection – prosleđivanje objekata drugim objektima kojima su potrebni, umesto da ih oni sami kreiraju. To znači da ćemo klasi proslediti objekte koji su joj neophodni I time smanjiti njenu odgovornost da ih sama kreira. Ovaj patern implementira loc.

Spring minimizuje svoje zavisnosti ka drugim bibliotekama i okvirima. Integracija Springa u aplikaciju je jednostavna uz korišćenje alata, kao što je Maven, Gradle, itd.

Konkretno, u projektu je korišćen **Maven**, alat koji se koristi za izgradnju I upravljanje bilo kojim projektima temeljenim na Javi, kao i integraciju sa već postojećim bibliotekama klasa.

Prednosti Maven-a:

- Jednostavno projektovanje
- Mogućnost dodavanja dependency
- Maven pruža informacije o projektu (log dokumenta, list dependency, izveštaje ...)
- Pomoći Maven-a možemo izgraditi bilo koji broj projekata u izlazne tipove poput JAR, WAR itd.

Spring Boot

Spring Boot je okvir koji omogućava kreiranje infrastrukture za samostalne aplikacije, spremne za produkciju. Spring Boot olakšava konfiguraciju Spring projekata sa dosta funkcionalnosti koje su ugrađene, a koje bi se inače zasebno konfigurisale.

Pogodnosti:

- Ugrađeni Tomcat ili Jetty serveri bez potrebe za zasebnim WAR fajlom, pa je aplikaciju dovoljno pokrenuti kao Java aplikaciju.
- Mogućnost korišćenja "starter" biblioteka za olakšanu inicijalnu konfiguraciju
- Automatska konfiguracija Spring i stranih biblioteka
- Ugrađene su biblioteke za različite metrike, statistike, procesa, niti ...
- Postavljanje inicijalnog *pom.xml* fajla, za konfiguraciju sa *Maven* alatom
- Security
- Obezbeđuje pristupačne startovanje razvoja aplikacije

Starter projekti se mogu kreirati na <https://start.spring.io>

Spring Boot starter omogućava da se smanji broj dodatnih zavisnosti u Maven ***pom.xml*** fajlu.
Primeri:

spring-boot-starter-parent:

- Java version UTF-8 source encoding
- Application.properties
- Dependency Management section

spring-boot-starter-web

spring-boot-devtools

spring-boot-starter-test

spring-boot-starter-thymeleaf

itd.

Thymeleaf template engine

Thymeleaf je serverski orjentisan Java generator forme, koji se koristi kako za web okruženja, tako i za ostala okruženja. Prednost je što se uklapa u HTML document ne menjajući njegovu spoljašnju formu. Lako se integriše sa modulima Spring framework-a. Thymeleaf je open-source Software, pod licencom Apache Licence 2.0.

Za konfiguraciju projekta se koriste zabeleške (engl. annotation). Na primer, zabeleška `@RestController` označava da je klasa REST kontroler; zabeleška `@RequestMapping` vezuje kontroler za URL preko koga je dostupan; zabeleška `@Controller` označava da je klasa kontroler koji radi sa template-ovima.

Ista zabeleška sa odgovarajućim atributima se koristi i na nivou metoda kontrolera da označi URL do metode, kao i tip zahteva: DOHVATI (engl. **GET**), POSTAVI (engl. **POST**), OBRISI (engl. **DELETE**) itd.

`@RequestMapping("/home")` – prihvata url (ako ne navedemo HTTP metod, po default-u postaje GET)

Ekvivalentno ovome `@RequestMapping("/home" method=RequestMethod.GET)` je `@GetMapping (value = "/home")`

`@GetMapping` – prikaz (prikaz stranice ...)

`@PostMapping` – kreiranje nečega

Embedded Server name: **TomCat**

Olakšica:

- TomCat je deo aplikacije u spring boot-u, pa nema potrebe za dodatnim podešavanjima i preuzimanjima

Resources folder

Framework automatski učitava fajlove iz foldera resources prilikom startovanja aplikacije. U folderu se nalaze dva zasebna foldera static (u kome su smešteni css fajlovi ...) i templates (u kome su smeštene html stranice).

Fajl koji se tu takođe nalazi je application.properties, u kome možemo da vršimo konfiguraciju:

- konfiguracija port na kome se startuje aplikacija
- tip baze
- konekciju sa bazom
- itd.

MariaDB

MariaDB je sistem za upravljanje bazama podataka. Sistem MariaDB je sistem otvorenog koda, licenciran GNU javnom licencom.

Sistem MariaDB i sistem MySQL su dva binarno kompatibilna sistema, odnosno oba sistema mogu izvršavati isti izvršni kod. U većini slučajeva, MariaDB će raditi potpuno isto kao i MySQL: sve komande, interfejsi, biblioteke i API-ji koji postoje u MySQL-u takođe egzistiraju i u MariaDB-u.

Oba sistema koriste iste nazive fajlova, iste putanje, protokole, interfejse, biblioteke za povezivanje sa različitim programskim jezicima, itd. Stoga, prelazak sa jednog sistema na drugi ne iziskuje puno vremena, dovoljno je samo instalirati servis MariaDB i deinstalirati servis MySQL.

Sistem MariaDB proširuje spektar funkcionalnosti sistema MySQL omogućavajući podršku za statističku obradu podataka, rad sa geografskim podacima, kao i različite funkcije za obradu teksta. MariaDB je skalabilno rešenje koje može biti dobar izbor i u slučaju da softversko rešenje obrađuje veliku količinu podataka, ali isto tako i kada to nije slučaj. Ovo je opravdano time što MariaDB podržava i relacione i nerelacione koncepte sistema za upravljanje bazama podataka. Dobra osobina ovog sistema je sigurnost i bezbednost.

Kod sistema MariaDB se koristi enkripcija u podrazumevanom režimu rada. S obzirom da je rešenje otvorenog koda ova činjenica dosta doprinosi bezbednosti rešenja.

MariaDB ima nekoliko integrisanih podsistema koji podržavaju rad sa bazom podataka. Podsistemi su odgovorni za izvršavanje upita, tj. oni preuzimaju upit od korisnika, izvršavaju isti nad bazom i vraćaju dobijeni rezultat.

Čitanje i pisanje podataka u sistemu MariaDB se izvršava tako što se te naredbe delegiraju podisteminima. Pored čitanja i pisanja podataka, podsistemi mogu pružiti i podršku za transakcije, keširanje podataka i indeksa i referencijalni integritet.

Transakcije

Transakcije su jedno od najbitnijih svojstava relacionih sistema za upravljanje bazama podataka. Sistem MariaDB ne podržava transakcije sama po sebi, ali ima omogućenu podršku preko podsistema. Podsistemi koji se mogu integrisati sa sistemom MariaDB i koji podržavaju transakcije su InnoDB, TokuDB i XtraDB. Podrška koju ćemo dobiti u smislu rada sa transakcijama i njihovih specifičnosti zavise od konkretnog podsistema koji koristimo.

Projekat: STUDENTSKA BAZA

Sistem za upravljanje bazom podataka: **MariaDB**

Klasa koja sadrži funkcije za rad sa bazom, izvršavanje upita i njenim povezivanjem je *Data.java*

Naziv baze: studentskabaza

password: ""

username: "root"

Konekcija sa bazom:

```
Connection conn = null;
Statement stmt = null;
String sql = null;

/**
 * Povezivanje sa bazom
 */

public Data(){

    try{

        conn = DriverManager.getConnection(
            "jdbc:mariadb://localhost/studentskabaza",
            "root",
            ""
        );
        stmt = conn.createStatement();

    } catch (SQLException throwables) {

        throwables.printStackTrace();
    }

}
```

STRUKTURA BAZE PODATAKA

baza: **studentskabaza**

Tabela users – čuva podatke o definisanim tipovima i dodeljenim ulogama korisnika koji se prijavljuju

USERS (id,username,password,email,role,ime,prezime)

Tabela predavaci – čuva podatke o profesorima

PREDAVACI (id,id_predavaca,ime,prezime)

Tabela admins – čuva podatke o administratorima

ADMINS (id_admina,ime,prezime,jmbg,mobilni,fiksni,username)

Tabela student – skladišti podatke o studentima

STUDENT (id_studenta,broj_indeksa,ime,prezime,jmbg,email,mobilni,fiksni,prosek,espb,username,password,finansije,role)

Tabela lozinka – čuva i prihvata prijave za promenu lozinke studenta ili predavača

LOZINKA (id,username)

Tabela ocene – skladišti sve podatke o prijavama studenata i dodeljenim ocenama

OCENE (id,broj_indeksa,sifra_predmeta,ocena)

Tabela prijave – čuva i prihvata podatke koji se odnose na prijave ispita

PRIJAVE (id_prijave,broj_indeksa,sifra_predmeta,naziv_predmeta,espb)

Tabela predmeti – čuva podatke o predmetima koje drže profesori i koje studenti mogu polagati

PREDMETI (id_predmeta,sifra_predmeta,naziv_predmeta,espb,id_predavaca)

Klasa projekta koja predstavlja kontroler ovog tipa ima naziv *Cont.java*

Endpoints:

Admin

[GET] <http://localhost:8080/admin> - pocetna strana za admina

[GET] <http://localhost:8080/indeksregister> prikaz forme za kreiranje novog studenta

[POST] <http://localhost:8080/registracijastudenta> - slanje podataka za kreiranje novog studenta kako bi se sacuvao u bazu

[GET] <http://localhost:8080/indeksregisterne> - prikaz forme za kreiranje novog studenta sa ispisom da je upisivanje studenta u bazu nemoguće iz nekog razloga (vec postoji u bazi itd.)

[GET] <http://localhost:8080/dodajprofesore> - prikaz forme za dodavanje novih profesora

[GET] <http://localhost:8080/dodajprofesorene> - prikaz forme za dodavanje novog profesora sa ispisom da je upisivanje profesora u bazu nemoguće iz nekog razloga

[POST] <http://localhost:8080/dodajprof> - slanje podataka u bazu za kreiranje novog profesora

[GET] <http://localhost:8080/dodajpredmete> - prikaz forme za dodavanje novih predmeta u bazu

[POST] <http://localhost:8080/dodelapredmeta> - slanje podataka u bazu za upis novih predmeta

[GET] <http://localhost:8080/updateprofesora> - prikaz forme za izmenu profesora

[GET] <http://localhost:8080/dodajprofesore> - prikaz forme za izmenu profesora sa ispisom da je nemoguće izvršiti izmenu iz nekog razloga (id profesora koji pokuša da se izmeni nije validan)

[POST] <http://localhost:8080/updatepredavaca> - aktivacija popunjene forme za izmenu profesora

[GET] <http://localhost:8080/brisanje> - prikaz forme za brisanje korisnika (profesora/studenta)

[POST] <http://localhost:8080/brisanjepodataka> - aktivacija funkcije za brisanje korisnika

[GET] <http://localhost:8080/brisanjee> - prikaz forme za brisanje korisnika sa ispisom da je zeljenog korisnika nemoguće obrisati (neispravni podaci/nepostojeci korisnik)

[GET] <http://localhost:8080/ispisiprofesore> - ispis profesora

Student

[GET] <http://localhost:8080/index> - studentska pocetna strana

[GET] <http://localhost:8080/ispiti> - ispis svih prijavljivanih ispita

[GET] <http://localhost:8080/predmeti> - ispis svih predmeta koje je moguće prijaviti I slusati u tekucoj godini

[GET] <http://localhost:8080/finansije> - prikaz finansija studenta

[GET] <http://localhost:8080/eprijava> - prikaz svih predmeta koje je moguće prijaviti

[POST] <http://localhost:8080/prijaviispit> - aktivacija prijave zeljenog predmeta za polaganje

Profesor

[GET] <http://localhost:8080/profesor> - pocetna profesorska strana

[GET] <http://localhost:8080/ispisiprijave> - ispis svih prijava studenata

[GET] <http://localhost:8080/dajocenu> - strana za dodeljivanje ocena studentima

[POST] <http://localhost:8080/potvrdiocenu> - aktivacija funkcije za dodelu ocene koja se upisuje u bazu

[POST] <http://localhost:8080/dodeliocenu> - uzimanje id prijave

[GET] <http://localhost:8080/ocena> - forma za dodeljivanje zeljene ocene

Forgotten password

[GET] <http://localhost:8080/forgottenpassword> - ispis forme za prosledjivanje zahteva za novom lozinkom

[POST] <http://localhost:8080/lozinka> - akcija za slanje zahteva za novom lozinkom

login

[GET] <http://localhost:8080/indexlogin> - login strana

[POST] <http://localhost:8080/checker> - akcija, proverava validnost prosledjenog username-a i passworda