

1. Laboratorijska vježba iz "Web aplikacija u Javi"

1.1. Jednostavna web aplikacija

Svrha laboratorijske vježbe je upoznavanje s razvojnim okruženjem IntelliJ IDEA Ultimate, autokonfiguracijom Spring Boot frameworka te osnovnom konfiguracijom Spring Web aplikacije koja se sastoji od jednog REST *controllera*, servisa i repozitorija.

1.2. Zadatak

Potrebno je kreirati Java web aplikaciju koja će omogućavati dohvat podataka o studentima. Dohvat podataka treba biti omogućen za sve studente ili za pojedinačnog studenta prema JMBAG-u. Podaci prikazani korisniku trebaju biti bez osobnih podataka studenta, tj. prikazani trebaju biti samo JMBAG i ECTS bodovi.

1. Unutar razvojnog okruženja IntelliJ IDEA Ultimate potrebno je kreirati Spring Boot projekt baziran na Maven strukturi. Odabrati "File -> New -> Project... -> Spring Initializr" te pritisnuti "Next". Na novom ekranu odabrati Java verziju 11. Pod "Group" unijeti "hr.tvz.prezime" gdje prezime treba zamijeniti s vlastitim prezimenom. Pod "artifact" unijeti "studapp". Pritisnuti "Next". Na sljedećem ekranu potrebno je odabrati samo "Spring Web" *dependency* te pritisnuti "Next" i "Finish". Nakon što se projekt otvori u novom ekranu, u donjem desnom kutu potrebno je odabrati "Enable Auto-Import".
2. Napisati klasu Student s *fieldovima* ime, prezime, datum rođenja, JMBAG i broj ECTS bodova. Nazive *fieldova* pisati na engleskom.
3. Napisati klasu StudentDTO s *fieldovima* JMBAG, broj ECTS bodova. Također, potrebno je dodati boolean *field* koji predstavlja treba li student plaćati studij. Uvjet za plaćanje studija je navršenih 26 godina. Nazive *fieldova* pisati na engleskom. Ova klasa predstavljat će podatke koji se prikazuju korisniku.

4. Napisati repozitorij klasu koja implementira sljedeći *interface*:

```
interface StudentRepository {  
  
    List<Student> findAll();  
  
    Optional<Student> findStudentByJMBAG(String JMBAG);  
  
}
```

Klasa treba funkcionirati kao odgovarajući Spring *bean* te se unutar nje treba kreirati lista s barem dva Student objekta iz koje će se vraćati povratne vrijednosti.

5. Napisati servisnu klasu koja implementira sljedeći *interface*:

```
public interface StudentService {  
  
    List<StudentDTO> findAll();  
  
    StudentDTO findStudentByJMBAG(String JMBAG);  
  
}
```

Klasa treba funkcionirati kao odgovarajući Spring *bean* te se u nju korištenjem *dependency injection* mehanizma treba ubaciti repozitorij klasa iz 4. koraka.

6. Napisati REST *controller* klasu s dvije GET *request handler* metode. Jednu za dohvat svih studentskih podataka i drugu za dohvat podataka jednog studenta prema JMBAG-u. Klasa treba funkcionirati kao odgovarajući Spring *bean* te se u nju korištenjem *dependency injection* mehanizma treba ubaciti servisna klasa iz 5. koraka. Podaci koje REST *controller* vrati web pregledniku trebaju biti u JSON reprezentaciji.
7. Kroz Spring Boot postavku uključiti logiranje *requestova* i *responsova* za REST *controller* u konzoli.
8. Demonstrirati rad aplikacije kroz *web* preglednik ili Postman aplikaciju.