

Uputstvo za predaju zadatka:

Kreirati *Spring Boot* web projekat upotrebom MVC šablona sa *Thymeleaf*-om, odnosno REST arhitekture sa AJAX pozivima.

Eclipse projekat nazvati **JWDXXModul2TestYY**, pri čemu je XX Vaš broj kursa, a YY Vaš identifikacioni broj u kursu (broj ugovora).

Za *Group Id* unesite **com.ftninformatika.jwd.modul2.test**

Za *Artifact Id* unesite **JWDXXModul2TestYY**

Projekat treba da bude napisan za Javu 1.8.

Projekat treba da se predaje tako da sadrži sledeće:

1. Direktorijum *src* (Izvorni kod u odgovarajućem *source* poddirektorijumu organizovan u odgovarajuću hijerarhiju paketa, HTML stranice u odgovarajućem poddirektorijumu)
2. Direktorijum *target*
3. *pom.xml*

Po završetku projekat zapakovati u arhivu pod nazivom **JWDXXModul2TestYY.zip**, pri čemu je XX Vaš broj kursa, a YY Vaš identifikacioni broj u kursu (broj ugovora).

Arhivu postaviti na <https://enastava.ftninformatika.com> u sekciju *Assignment*.

Tekst zadatka:

NAPOMENA: sledeće stavke zadatka su manje ili više nezavisne jedna od druge. Osim tamo gde je eksplicitno navedeno (npr. redirekcija), nije potrebno praviti nikakve veze (npr. link-ove) između stranica realizovanih u pojedinačnim stavkama. Pri ocenjivanju zadatka, stranicama će se pristupati kroz URL *web browser*-a (navedeni su za svaki zadatak pojedinačno).

Podesiti da osnovni URL nastavak na nivou aplikacije bude /Test!

1. Definirati klase koje predstavljaju liniju vožnju. Postaviti u memoriju na nivou aplikacije kolekciju od bar 3 linije i kolekciju od bar 5 unapred kreiranih vožnji (npr. neke od onih iz datoteke *sablon.html*). Za svaki kreirani objekat klase vožnja, potrebno je vezati po jedan objekat njemu odgovarajuće linije.
2. Potrebno je kopirati formu iz datoteke *sablon.html* u datoteku **zadatak2.html**, a nju smestiti u odgovarajući direktorijum tako da joj se može pristupiti kroz URL **localhost:8080/Test/zadatak2.html**. Napisati kontroler **VoznjaController** i *handler* metodu koja obrađuje HTTP *post* zahteve, a poziva se pri kreiranju novih vožnji. Metoda preuzima podatke o vožnji sa forme iz datoteke *zadatak2.html* a zatim na osnovu njih kreira novi objekat klase koji predstavlja vožnju i čuva je u postojećoj kolekciji u memoriji na nivou aplikacije. Pre kreiranja nove vožnje pronaći iz postojeće kolekcije u memoriji na nivou aplikacije liniju - odgovarajući objekat klase koji predstavlja odabranu liniju i njega povezati sa vožnjom pri njenom kreiranju. HTML kod u datoteci *zadatak2.html* je potrebno dopuniti tako da podrži slanje zahteva sa odgovarajućim parametrima implementirano u *handler* metodi kontrolera. Svaki put nakon dodavanja nove vožnje, u *for* petlji u **Eclipse konzoli ispisati** sve vožnje (premostiti *toString()* metodu klase koja predstavlja vožnju). Sprečiti unos i izvršiti redirekciju na stranicu *zadatak2.html* ukoliko je odabrana prigradska linija, a vreme polaska je između 00:00 i 04:00.

ALTERNATIVNO: Ukoliko neko ne zna da pošalje parametre iz input-a tipa *time*, može da input prepravi u input tipa *text*, a vreme unosi u odgovarajućem standardnom ISO formatu HH:mm.

ALTERNATIVNO: Ukoliko neko ne zna da uradi validaciju parametara, može da je izostavi ali će rešenje nositi manje bodova.

NAPOMENA: Ne koristiti servisni sloj i ne raditi sa bazom!

NAPOMENA: Zadatak neće biti pregledan ukoliko se stranici *zadatak2.html* ne može prisupiti iz *web browser*-a, kao i ukoliko slanje zahteva iz forme ne uspe bar da pozove *handler* metodu!

3. Potrebno je kopirati tabelu iz datoteke *sablon.html* u *Thymeleaf* šablon *zadatak3.html*, a njega smestiti u odgovarajući direktorijum. Za kontroler *VoznjaController* napisati *handler* metodu koja odgovara na zahteve sa URL-om *localhost:8080/Test/Voznje/Zadatak3* i uz oslonac na *Thymeleaf* šablon *zadatak3.html* obezbeđuje tabelarni prikaz svih vožnji koje se nalaze u kolekciji u memoriji na nivou aplikacije. Obratiti pažnju da 2. i 4. kolona prikazuju podatke povezanog entiteta, pa da je za pun broj bodova u ovom zadatku potrebno napisati izraze koji ispis formatiraju na način koji je prikazan u primeru. 3. kolonu je potrebno formatirati uz pomoć odgovarajućeg izraza. Ispod tabele prikazati brojače vožnji u gradskom i prigradskom saobraćaju. Ove vrednosti izračunati u *handler* metodi kontrolera i proslediti u pogled.

Dopuniti u zadatku 2 *handler* metodu tako da se nakon svega (ostaviti usputni ispis u *Eclipse* konzoli) radi i redirekcija na prikaz svih vožnji, a koji je realizovan u okviru ovog zadatka.

ALTERNATIVNO: Izračunavanje i prikaz broja vožnji za gradski i prigradski saobraćaj se mogu izostaviti ali će rešenje nositi manje bodova.

NAPOMENA: Ne koristiti servisni sloj i ne raditi sa bazom!

NAPOMENA: Zadatak neće biti pregledan ukoliko *handler* metoda nije povezana sa *Thymeleaf* šablonom, kao i ukoliko se u prozoru pretraživača prikazuje *Thymeleaf* greška. Pri predaji rešenja izostaviti sve elemente *Thymeleaf* sintakse koji su nedovršeni i proizvode greške!

4. Inicijalizovati u memoriji na nivou korisnika kolekciju svih linija za koje je korisnik dodavao vožnje. **Dopuniti *handler* metodu iz zadatka 2** tako da se u kolekciju jednom doda svaka linija kada korisnik prvi put za nju doda vožnju. **Dopuniti *handler* metodu zadatka 3** tako da u *Eclipse* konzoli u *for* petlji ispiše sadržaj ove kolekcije. **Dopuniti i *Thymeleaf* šablon *zadatak3.html*** tako da ispod tabele ispiše i sadržaj ove kolekcije.

5. Za kontroler *VoznjaController* napisati *handler* metodu koja odgovara na zahteve sa URL-om *localhost:8080/Test/Voznje/Zadatak5* čijim pozivom se korisniku obezbeđuje tabelarni prikaz svih vožnji sa pretragom. Uz oslonac na **servisni sloj i sloj za pristup podacima (DAL)**, implementirati funkcionalnost pretrage vožnji po odabranoj liniji. Prikaz treba da bude realizovan pomoću *Thymeleaf* šablona *zadatak5.html* nastalog iz kopije sadržaja šablona *zadatak3.html*, a dopunjenog formom za pretragu sa padajućim menijem za odabir linije (omogućiti i da se odabere opcija za prikaz svih linija). Pre prosleđivanja *Thymeleaf* šablonu, rezultat pretrage u *for* petlji **ispisati u *Eclipse* konzoli** (premostiti *toString()* metodu klase koja predstavlja vožnju).

localhost:8080/Test/Voznje/Zadatak5?linija=0

localhost:8080/Test/Voznje/Zadatak5?linija=1

localhost:8080/Test/Voznje/Zadatak5?linija=2

ALTERNATIVNO: ukoliko neko ne zna da radi sa bazom, rešenje može biti urađeno oslanjajući se na *in-memory* servise, ali će rešenje nosti manje bodova.

ALTERNATIVNO: ukoliko neko u okviru zadatka ne zna implementira pretragu, može da implementira prikaz svih vožnji, bez pretrage a sa čitanjem iz servisnog sloja, ali će rešenje nosti manje bodova.

ALTERNATIVNO: ukoliko neko ne zna da radi sa *Thymeleaf*-om, može ostaviti ispis rezultata pretrage u konzoli, ali će rešenje nositi manje bodova.

NAPOMENA: Zadatak neće biti pregledan ukoliko ne bude radio bar ispis kolekcije objekata preuzete iz servisnog sloja (sa ili bez pretrage) u konzoli!

6. Potrebno je kopirati formu iz datoteke *zadatak7.html* u datoteku ***zadatak6.html***, a nju smestiti u odgovarajući direktorijum tako da joj se može pristupiti kroz URL ***localhost:8080/Test/zadatak6.html***. Upotrebom *JavaScript*-a onemogućiti da se pošalju vrednosti ukoliko:

- I. nisu popunjena sva polja
- II. je redni broj manji ili jednak 0
- III. su polazište i odredište jednake vrednosti

Za svaki od prethodnih slučajeva prikazati obaveštenje o greški u *alert* prozoru. Nije potrebno farbati polja sa neispravnim unosom.

NAPOMENA: Nije potrebno povezivati ovu stranicu sa *controller*-ima!

NAPOMENA: Zadatak neće biti pregledan ukoliko se stranici *zadatak6.html* ne može prisupiti iz *web browser*-a, kao i ako *JavaScript* proizvodi greške. Pri predaji rešenja izostaviti fragmente *JavaScript* koda koji proizvode greške!

7. Iskopirati datoteku ***zadatak7.html*** u odgovarajući direktorijum tako da joj se može pristupiti kroz URL ***localhost:8080/Test/zadatak7.html***. Upotrebom *JavaScript*-a za slanje AJAX zahteva i obradu odgovora uz razmenu JSON objekata, u *handler* metodi/metodama novog ***LinijaController***-a implementirati dodavanje nove linije i prikaz svih (i dodatih) linija na istoj stranici (bez prelaska na novu stranicu). Inicijalno pri otvaranju stranice prikazati postojeće linije tj. popuniti tabelu za pregled linija. Dodavanje novih i prikaz postojećih linija implementirati uz oslonac na **servisni sloj i sloj za pristup podacima (DAL)**.

Vrednost kolone *Red vožnje* je potrebno izvesti iz vrednosti atributa *gradski*. Ne određivati ovaj podatak na strani servera!

ALTERNATIVNO: ukoliko neko neko ne zna da implementira AJAX pozive, može implementirati *offline client-side* stranicu sa čitanjem iz *in-memory* niza (po ugledu na termin 10), ali će rešenje nositi manje bodova.

ALTERNATIVNO: ukoliko neko ne zna da radi sa bazom rešenje može biti urađeno oslanjajući se na *in-memory* servise, ali će rešenje nositi manje bodova.

NAPOMENA: Zadatak neće biti pregledan ukoliko se stranici *zadatak7.html* ne može prisupiti iz *web browser*-a, kao i ako *JavaScript* proizvodi greške. Pri predaji rešenja izostaviti fragmente *JavaScript* koda koji proizvode greške!