

Vježba: 13. Upoznavanje s ORM, Entity Bean, razvoj Web aplikacije koja koristi web servise, JSF, Ajax i WebSocket

Kreiranje 13. vježbe (direktorij **{LDAP_korisničko_ime}_zadaca_3**). U nastavku se direktorij za vježbu simbolički označava kao **{vježba}**.

Pokretanje programa NetBeans.

Pokretanje poslužitelja Glassfish 5.1.0 (Services/Servers/Glassfish Server 5.1.0/Start)

Vježba_13 – Zadaća 3: Web aplikacije s ORM, Entity Bean, web servisima uz korištenje JSF, Ajax i WebSocket

Naziv projekta: {LDAP_korisničko_ime}_zadaca_3

Korijenski direktorij treba biti {LDAP_korisničko_ime}_zadaca_3

Za rad s postavkama treba koristiti Java biblioteke iz vježba_03_2 i vježba_06_1. Vlastite klase i metode trebaju biti komentirane u Javadoc formatu. **Projekt isključivo treba predati u formatu NetBeans projekta.** Prije predavanja projekta potrebno je napraviti Clean na svim projektima. Zatim cijelu zadaću (uključujući tri projekta iz zadaće te projekte vježba_03_2 i vježba_06_1) sažeti u .zip (NE .rar) format s nazivom {LDAP_korisničko_ime}_zadaca_3.zip i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka (.xml) i popunjeni obrazac za zadaću pod nazivom {LDAP_korisničko_ime}_zadaca_3.[doc | pdf] (u direktoriju {LDAP_korisničko_ime}_zadaca_3).

Sustav je sastoji od jedne Enterprise Application (projekt {LDAP_korisničko_ime}_zadaca_3) koja sadrži tri projekta:

{LDAP_korisničko_ime}_zadaca_3_1 – podatkovni dio

{LDAP_korisničko_ime}_zadaca_3_2 – poslovni dio

{LDAP_korisničko_ime}_zadaca_3_3 – korisnički dio.

Kako su za izvršavanje projekata iz zadaće potrebni vaši projekti vježba_03_2 i vježba_06_1 potrebno je kod njih odraditi malu promjenu u nazivu (vidljivi dio u NetBeans). Otvori se Properties za projekt (npr. vježba_03_2) te se u dijelu Name podesi da izgleda u obliku ({LDAP_korisničko_ime}_ vježba_03_2. Kako je autor Matija onda je naziv matnovak_vježba_03_2. Tako napraviti i za vježba_06_1 pa će kod Matije biti matnovak_vježba_06_1. Oba projekta kopirajte u direktorij {LDAP_korisničko_ime}_zadaca_3) tako da budu zajedno s 3 projekta zadaće.

Napomena: za testiranje, izvršavanje i bodovanje ove zadaće očekuje se da će se putem prethodne zadaće prikupiti sljedeći minimalni brojevi zapisa po tablicama:

1. MYAIRPORTS – 20 jedinstvenih aerodroma
2. MYAIRPORTSLOG – 30 dana za svaki aerodrom iz MYAIRPORTS
3. AIRPLANES – 100.000 letova aviona

Priprema JDBC bazena veza (Connection pool) i izvora podataka (Data source) za rad:

Priprema Glassfish poslužitelja za rad s JDBC. Mogu se koristiti dva načina:

- Domain Admin Console:
 1. Neće biti ovdje opisan zbog mnogo podataka koje treba unositi u više dijaloških okvira
- asadmin: ('Command Prompt'):
 1. preporučuje se postaviti putanju izvršnih direktorija na Java 8.* (npr.)

```
set path="C:\Program Files\Java\jdk1.8.0_201\bin"
```
 2. postaviti se na bin direktorij Glassfish poslužitelja (npr.)

```
cd /d "C:\Program Files\Glassfish5.1.0\bin"
```
 3. preuzeti datoteku glassfish_DZ3_komande.txt. Datoteka sadrži veći broj komandi osim onih koje se spominju u nastavku.
 4. ući u komandni mod rada s poslužiteljem:

```
asadmin --host localhost --port 4848
```
 5. izvršiti komandu za kreiranje JDBC bazena veza (jdbc connection pool) kao jednu liniju. g(i) treba prilagoditi za grupu i=1,..,4 na vježbama:

```
create-jdbc-connection-pool  
--datasourceclassname  
org.apache.derby.jdbc.ClientDataSource40  
--restype javax.sql.DataSource  
--property  
portNumber=1527:password=g1_nwtis:user=nwtis_g1:serverName=lo  
calhost:databaseName=nwtis_g1:url="jdbc\:derby\://localhost\  
1527/nwtis_g1" NWTiS_DZ3_Pool
```
 6. provjeriti prethodno kreirani JDBC bazen veza:

```
ping-connection-pool NWTiS_DZ3_Pool
```
 7. izvršiti komandu za kreiranje JDBC resursa (jdbc resource) kao jednu liniju:

```
create-jdbc-resource  
--connectionpoolid NWTiS_DZ3_Pool jdbc/NWTiS_DZ3
```

Dopuna baze podataka za rad:

1. Preuzeti datoteku SQL izmjene i dopuneDatoteka. Datoteka sadrži veći broj SQL komandi za kreiranje tablice, punjenje podataka, kreiranje indeks i vanjskih ključeva. To je za Java DB.
2. Otvorite vezu u Java DB na bazu podataka za vježbe nwtis_g{1-4}
3. Izvršite komande

Projekt {LDAP_korisničko_ime}_zadaca_3_1 se sastoji od:

1. EntityBean klasa kao veze prema tablicama iz baze podataka
2. Facade klasa za rad s EntityBean klasama. Kod određenih klasa potrebno je dodati metode koje će obavljati potrebne aktivnosti na povezanom EB.

Projekt {LDAP_korisničko_ime}_zadaca_3_2 se sastoji od:

1. SessionBean klasa kao dio provođenja poslovnih aktivnosti kao što su dohvat meteo podataka za aerodrom na bazi njegove GPS lokacije uz korištenje klase OWMKlijent za web servis OpenWeatherMap. Druga aktivnost je komunikacijski dio na bazi WebSocket-a. Na poslužiteljevoj krajnjoj točki za WebSocket se po prijemu poruke s podacima o korisniku i aerodromu dodaje zapis u tablicu MYAIRPORTS. Nakon toga se svima povezanim na krajnju točku šalje poruka o ukupnom broju aerodroma za koje se preuzimaju podaci o letovima.

Projekt {LDAP_korisničko_ime}_zadaca_3_3 sastoji se od korisničkog dijela u kojem se mogu obaviti aktivnosti:

1. aktivnost (A1) se temelji na 3 hijerarhijski povezana prikaza: korisnici (P1), korisnikovi aerodromi (P2), dani preuzetih podataka za aerodrom (P3). Rad se isključivo temelji na Ajax-u. Na početku samo je vidljiv P1. Izvršavanjem odabira na P1 otvara se P2 (ako je sakriven), a ako je vidljiv P3 on se sakriva. Izvršavanjem odabira na P2 otvara se P3 (ako je sakriven). Prikazivanje i sakrivanje temelje se na primjeni CSS klasa koja sadrže potrebne osobine. P1 se temelji na odabiru korisnika iz popisa korisnika iz tablice KORISNICI. Može se koristiti padajući izbornik ili tablični prikaz. Kod izbornika nakon imena i prezimena treba biti u zagradi podatak o broju aerodroma koje prati taj korisnik. Kod tablice to je poseban stupac. Gumbom (posebnim kod izbornika ili u retku za korisnika) „Prikaži korisnikove aerodrome“ otvara se P2 tablični prikaz korisnikovih aerodroma na temelju filtriranih podataka iz tablice MYAIRPORTS. U svakom retku uz podatke o aerodromu postoje podaci o broju korisnika koji ga prate (MYAIRPORTS), broju dana za koje su preuzeti podaci o letovima s tog aerodroma (MYAIRPORTSLOG), broj preuzetih letova s tog aerodroma (AIRPLANES) i gumb „Prikaži datume“ za odabir aerodroma. Odabirom aerodroma otvara se P3 tablični prikaz s podacima o danima preuzimanja podataka (prethodno je bio sakriven). U svakom retku prikazan je podatak o danu preuzimanja u HR formatu dd.mm.gggg, vremenu preuzimanja u HR formatu dd.mm.gggg hh:mm:ss.mmm te gumb „Brisanje podataka“. Gumbom se briše zapis aerodroma s odabranim datumom (MYAIRPORTSLOG). Ispod tablice nalazi se gumb „Vrati se na početak“ kojim se sakrivaju prikazi P2 i P3.
2. aktivnost (A2) se temelji na 2 hijerarhijski povezana prikaza: korisnici (P1), filtrirani aerodromi (P2). Rad se temelji na Ajax-u i WebSocket-u. Na početku samo je vidljiv P1. Izvršavanjem odabira na P1 otvara se P2 (ako je sakriven). Prikazivanje i sakrivanje temelje se na primjeni CSS klasa koja sadrže potrebne osobine. P1 na vrhu stranice sadrži podatak o ukupnom broju aerodroma za koje se preuzimaju podaci o letovima i vrijeme zadnjeg osvježavanja podatka o

broju aerodroma. Ti podaci se isključivo prikazuju temeljem WebSocket komunikacije. Slijedi odabir korisnika iz popisa korisnika iz tablice KORISNICI. Koristi se suprotna varijanta u odnosi na P1 u A1. Uz odabir korisnika nalazi se dio za filtriranje aerodroma na temelju naziva. On sadrži element za unos naziva i gumbi „Potraži JPQL” i „Potraži CApi”. Svi gumbi otvaraju P2 tablični prikaz aerodroma na temelju filtriranih podataka iz tablice AIRPORTS. Razlika je u pozivu metode iz pripadajuće facade. Svaka metoda koristi svoj način definiranja upita. Jedna koristi JPQL, a druga Criteria API. U svakom retku uz podatke o aerodromu postoje podaci o trenutnoj temperaturi i vlazi (na temelju web servisa OpenWeatherMap) te gumb za odabir aerodroma. Odabirom aerodroma šalje se poruka (korisnik, aerodrom) putem WebSocket veze na poslužitelj.

{LDAP_korisničko_ime}_zadaca_3_1: Razvoj EJB modula za podatkovni sloj logiku (Stateful, Stateless i Singleton Session Bean)

Preporučeni koraci:

1. Kreiranje projekta {LDAP_korisničko_ime}_zadaca_3_1 kao Java EE/EJB Module, server Glassfish 5.1.0, Java EE verzija: Java EE 8
2. Otvoriti Properties i promijeniti Packaging u ejb
3. Kreirati pakete org.foi.nwtis.{LDAP_korisnik}.ejb.eb, pakete org.foi.nwtis.{LDAP_korisnik}.ejb.sb
4. Kreirati opisnik isporuke za EJB (New/Other.../Enterprise JavaBeans/Standard Deployment Descriptor)
5. Obisati klase JavaEE8Resource i JAXRSConfiguration
6. U pom.xml dodati NWTiS riznicu (repository) kao što je opisano u prvom projektu
7. Dodati ovisnosti/dependency za (2.5.2 ili 2.7.6):

artifactId	groupId	version
NWTiS_2020_REST_lib	org.foi.nwtis	1.1
org.eclipse.persistence.core	org.eclipse.persistence	2.7.6
org.eclipse.persistence.asm	org.eclipse.persistence	2.7.6
org.eclipse.persistenceantlr	org.eclipse.persistence	2.7.6
org.eclipse.persistence.jpa	org.eclipse.persistence	2.7.6
org.eclipse.persistence.jpa.jpql	org.eclipse.persistence	2.7.6
org.eclipse.persistence.extension	org.eclipse.persistence	2.7.6
org.eclipse.persistence.jpa.modelgen.processor	org.eclipse.persistence	2.7.6
javax.persistence	org.eclipse.persistence	2.2.1

7. Provjeriti ako postoji src/main/resource/persistence.xml i obrisati
8. Kreirati perzistantnu jedinicu (New/Other/Persistence/Persistence Unit) pod nazivom **NWTiS_DZ3_PU**, EclipseLink, Data Source: **jdbc/NWTiS_DZ3**, Table Generation Strategy: None
9. Kreirati klase na entitete (New/Other/Persistence/Entity Classes from Database) za sve tablice u paket org.foi.nwtis.{LDAP_korisnik}.ejb.eb. Bez Generate Named Query..., Bez Generate Mapped... Next>, Collection Type: List, označiti sve osim Fully Qualified Database Table Names
10. kreirati session bean-nove (Facade) za klase entiteta (New/Other/Persistence/Session Beans for Entity Classes) u paket org.foi.nwtis.{LDAP_korisnik}.ejb.sb, označiti Local
11. ako se tijekom rada javi problem sa Facade klasama najbolje je izaći iz NetBeans i obrisati NetBeans cache (npr. za Windows 10 C:\Users\{korisnik}\AppData\Local\NetBeans\Cache). Ako postoji problem s generiranje klasa meta modela, potrebno je obrisati biblioteku za generiranje metamodela i koristiti String nazive za atribute.
12. da bi se vidjele SQL komande koje izvršava EclipseLink može se dodati u persistence.xml (<http://wiki.eclipse.org/EclipseLink/Examples/JPA/Logging>)

```
<properties>
  <property name="eclipselink.logging.level" value="FINE"/>
  <property name="eclipselink.logging.level.sql" value="FINE"/>
  <property name="eclipselink.logging.parameters" value="true"/>
  <property name="eclipselink.logging.logger" value="DefaultLogger"/>
</properties>
```

13. izgraditi aplikaciju.

{LDAP_korisnicko_ime}_zadaca_3_2: Razvoj EJB modula za poslovnu logiku

Provesti korake iz vježba_12_2 od broja 1 do 9 te 19-21, uz potrebna prilagođavanja na potrebe 3. zadatke.

{LDAP_korisnicko_ime}_zadaca_3_3: Kreiranje web modula za korisnicko sučelja

Provesti korake iz vježba_12_3 od broja 1 do 10, 13-16, 24-29, te 51, uz potrebna prilagođavanja na potrebe 3. zadatke.

Preporučeni koraci:

1. Izgraditi, isporučiti, izvršiti u Debug mode (i testirati aplikaciju).
2. Pogledati u Files/{LDAP_korisnicko_ime}_zadaca_3_3/target/{LDAP_korisnicko_ime}_zadaca_3_3-1.0.war što je sve u sadržaju, posebno u WEB-INF/lib
3. Otvoriti **pom.xml** i promijeniti kod:
 - (a) <dependency> za {LDAP_korisnicko_ime}_zadaca_3_1, {LDAP_korisnicko_ime}_zadaca_3_2 dodati:

```
<scope>provided</scope>
```
4. Izgraditi, isporučiti, izvršiti u Debug mode (i testirati aplikaciju). Web aplikacija više ne radi jer ne može dohvatiti zrna koja su u ejb projektima. To je zbog provided tako da nisu više sastavni dio war datoteke.
5. Pogledati u Files/{LDAP_korisnicko_ime}_zadaca_3_3/target/{LDAP_korisnicko_ime}_zadaca_3_3-1.0.war i usporediti s prethodnim sadržajem
6. Aplikacija treba postati sastavni dio Enterprise Application i putem nje se izvršavati.

{LDAP_korisničko_ime}_zadaca_3: Integracija EJB i web modula u zajedničku Enterprise Application ear datoteku

Preporučeni koraci:

7. Kreiranje projekta **{LDAP_korisničko_ime}_zadaca_3** na bazi artefakta (New Project/Java with Maven/Project from Archetype) Search: ear, Known Archetypes: ear-javaee7.
8. Dodati ovisnost/dependency na {LDAP_korisničko_ime}_zadaca_3_1, {LDAP_korisničko_ime}_zadaca_3_2, {LDAP_korisničko_ime}_zadaca_3_3
9. Otvoriti **pom.xml** i promijeniti kod:
 - (b) <maven-compiler-plugin> <version> iz 3.1 u najnoviju (3.8.1)
 - (c) <configuration> <source> iz 1.7 u 1.8
 - (d) <maven-ear-plugin> <version> iz 2.8 u najnoviju (3.0.2)
 - (e) <configuration> <version> iz 6 u 8. To je Java EE 8.
 - (f) <configuration> dodati sve module od kojih se sastoji aplikacija

```
<modules>
  <ejbModule>
    <groupId>org.foi.nwtis.dkermek</groupId>
    <artifactId>{LDAP_korisničko_ime}_zadaca_3_1</artifactId>
    <unpack>true</unpack>
  </ejbModule>
  <ejbModule>
    <groupId>org.foi.nwtis.dkermek</groupId>
    <artifactId>{LDAP_korisničko_ime}_zadaca_3_2</artifactId>
    <unpack>true</unpack>
  </ejbModule>
  <webModule>
    <groupId>org.foi.nwtis.dkermek</groupId>
    <artifactId>{LDAP_korisničko_ime}_zadaca_3_3</artifactId>
    <unpack>true</unpack>
  </webModule>
</modules>
```
 - (g) <dependency> za {LDAP_korisničko_ime}_zadaca_3_1 i {LDAP_korisničko_ime}_zadaca_3_2 dodati

```
<type>ejb</type>
```
 - (h) <dependency> za {LDAP_korisničko_ime}_zadaca_3_3 dodati

```
<type>war</type>
```
10. Otvoriti Properties kod General staviti za Name: **{LDAP_korisničko_ime}_zadaca_3** i kod Run za Server: odabrati Glassfish Server 5.1.0.
11. Izgraditi, isporučiti, izvršiti u Debug mode (i testirati aplikaciju).
12. Pogledati u Files/{LDAP_korisničko_ime}_zadaca_3/target/{LDAP_korisničko_ime}_zadaca_3-1.0.ear što je sve u sadržaju, posebno META-INF/application.xml

{LDAP_korisnicko_ime}_zadaca_3-svi: Projekt koji je roditeljski za sve ostale projekte tako da se svi projekti mogu izgraditi s jednog mjesta

Preporučeni koraci:

1. Kreiranje projekta **{LDAP_korisnicko_ime}_zadaca_3-svi** na bazi artefakta (New Project/Java with Maven/Project from Archetype) Search: pom, Known Archetypes: pom-root. Kao mjesto se uzima privremeni direktorij, ne unutar korijenskog direktorija projekata 3. zadace
2. Ovo nije idealan način da se dobije ciljani oblik no jednostavan je i ne traži previše posla
3. Zatvoriti projekt {LDAP_korisnicko_ime}_zadaca_3-svi
4. Pomoću File Explorer kopirati datoteku pom.xml iz privremenog direktorija i ispustiti ju u korijenskom direktorija projekata 3. zadace
5. Otvoriti projekt u NetBeans iz korijenskog direktorija projekata 3. zadace
6. Otvoriti **pom.xml** iz projekta {LDAP_korisnicko_ime}_zadaca_3_3 te kopirati <properties>
7. Otvoriti **pom.xml** i ispustiti prije </project>
8. Obrisati ostalo tako da ostanu <maven.compiler.*> i <project.build.sourceEncoding>
9. Dodati module nakon </properties>

```
<modules>
  <module>{LDAP_korisnicko_ime}_zadaca_3_1</module>
  <module>{LDAP_korisnicko_ime}_zadaca_3_2</module>
  <module>{LDAP_korisnicko_ime}_zadaca_3_3</module>
  <module>{LDAP_korisnicko_ime}_zadaca_3</module>
</modules>
```

10. Sada se mogu svi projekti izgraditi jednom akcijom npr. Clean&Build