

## Vježba 13. / Zadaća 4. - Opis

Naziv: enterprise aplikacija za rad s bazom podataka putem ORM uz korištenje JPA uz korisničku stranu s JSF na bazi predloška i korištenje Ajax-a.

Enterprise aplikacija služi za rad s bazom podataka putem ORM uz korištenje JPA (bez SQL, koristi se Criteria API), uz korisničku stranu s JSF na bazi predloška, dodavanja i ažuriranje adresa, preuzimanje prognoza, pretraživanje i filtriranje dnevnika, sve operacije provode se Ajax-om. Enterprise aplikacija pod nazivom **{LDAP\_korisničko\_ime}\_zadaca\_4** sastoji se od tri modula:

1. **{LDAP\_korisničko\_ime}\_zadaca\_4\_1**: EJB modul za podatkovni sloj s JPA
2. **{LDAP\_korisničko\_ime}\_zadaca\_4\_2**: EJB modul za poslovnu logiku i korištenje EJB podatkovnog sloja
3. **{LDAP\_korisničko\_ime}\_zadaca\_4\_3**: Web modul za korisničko sučelje

Svaka akcija treba biti zapisana u tablicu dnevnik. Svi prikazi trebaju imati abecedni popis elemenata. Postupak počinje unosom nove adrese i pregledom postojećih adresa (o obliku padajućeg izbornika) iz tablice adrese. Dodavanje nove adrese (putem gumba) Ajaxom ažurira tablicu adresa i pregled postojećih adresa. Iz postojećih adresa može se odabrati pojedina adresa (ili više njih) i putem gumba uz Ajax dodati u popis kandidiranih adresa za prognoze. Adresa(e) koja se odabire više se ne pojavljuje u pregledu adresa. Moguće je odabrati adresu i pokrenuti njeno ažuriranje (putem gumba) u istom pogledu (daj dio je prethodno bio sakriven). Nakon ažuriranja podataka (putem gumba) vraća se na odabir adresa (i sakriva se dio za ažuriranje). Kod kandidiranih adresa za prognoze može se maknuti pojedina adresa (ili više njih) putem gumba i Ajaxa (i tada se ona(e) ponovno pojavljuje u pregledu adresa). Nakon pokretanja prognoza putem gumba i Ajaxa preuzimaju se prognoze vremena za odabranu adresu (samo jednu) između kandidiranih adresa za prognozu. Zatim se prikazuju podaci prognoza (datum, meteo podaci...) u bloku s tablicom (daj dio je prethodno bio sakriven) i nakon završetka pregleda prognoza (putem gumba) cijeli blok se sakriva. Pregled dnevnika obavlja se u posebnom pogledu. Mogu se unijeti podaci za filtriranje na bazi; vremena od-do, ipadrese, trajanja i statusa. Može se unijeti od ni jednog pa do svih podataka. Nakon pokretanja pregleda (putem gumba) prikazuju se filtrirani podaci iz dnevnika.

Klase i metode trebaju biti komentirane u javadoc formatu. Prije predavanja projekta potrebno je napraviti Clean na projektu. Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom **{LDAP\_korisničko\_ime}\_zadaca\_4.zip** i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka i popunjeni obrazac za zadaću pod nazivom **{LDAP\_korisničko\_ime}\_zadaca\_4.doc** (u korijenskom direktoriju projekta).

Naziv projekta: **{LDAP\_korisničko\_ime}\_zadaca\_4**

1. Kreiranje direktorija **{LDAP\_korisničko\_ime}\_zadaca\_4**. U nastavku se

direktorij za vježbu simbolički označava kao **{vježba}**.

2. startati Java DB

3. kreirati Java DB bazu podataka (Services/Java DB/Create Database...) nwtis\_zadaca\_4, korisničko ime nwtis, lozinka foi2016 (Slika 1)

4. otvoriti vezu prema novoj bazi podataka

5. preuzeti datoteku adrese\_JavaDB.sql, kopirati sadržaj i izvršiti (u otvorenoj vezi na bazu podataka odabrati Tables/Execute Command...)

6. preuzeti datoteku dnevnik\_JavaDB.sql, kopirati sadržaj i izvršiti (u otvorenoj vezi na bazu podataka odabrati Tables/Execute Command...)

7. provjeriti sadržaj sheme (kao na Slici 2.)

**{LDAP\_korisničko\_ime}\_zadaca\_4\_1**: Kreiranje modula za podatkovni sloj s EJB: EntityBean - JPA

1. Kreiranje projekta **{LDAP\_korisničko\_ime}\_zadaca\_4\_1** kao Java EE/EJB Module , server Glassfish, Java EE verzija: Java EE 7

2. kreirati pakete **org.foi.nwtis.{LDAP\_korisnik}.ejb.eb**, **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**

3. kreirati perzistentnu jedinicu (New/Other/Persistence/Persistence Unit) . Slike 3, 4 i 5 prikazuju aktivnosti koje se provode: određivanje naziva: zadaca\_4\_1PU, kreiranje izvora podataka (Data source) pod nazivom jdbc/NWTiS\_2016 za ranije kreranu vezu prema bazi podataka nwtis\_zadaca\_4. Slika 6 prikazuje podatke kreirane perzistente jedinice. Slike 7 i 8 pokazuju kreirane datoteke opisnika (persistence.xml i glassfish-resources.xml)

4. kreirati shemu baze podataka (New/Other/Persistence/Database Schema). Naziv: NWTiS\_2016 (Slika 7) na **src/conf** i odabrati ranije kreranu vezu prema bazi podataka nwtis\_zadaca\_4 (Slika 8). Slijedi odabir svih tablica (Slika 9). Struktura sheme baze podataka prikazana je na slici 10. Služi nam kao pomoć i nije nužno to obavljati.

5. kreirati klase na entitete (New/Other/Persistence/Entity Classes from Database) na **org.foi.nwtis.{LDAP\_korisnik}.ejb.eb**. Slijedi odabir svih tablica s time da se može odabrati kao osnova izvor podataka jdbc/NWTiS\_2016 (Slika 11) ili shema baze podataka NWTiS\_2016 ako smo ju kreirali u projektu (Slika 12). Preslikavanja tablica baze podataka u klase prikazana je na slici 13. I na kraju se određuju elementi preslikavanja (Slika 14)

6. kreirati session bean-nove (Facade) za klase entiteta (New/Other/Persistence/Session Beans for Entity Classes) na **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**. Slijedi odabir klasa (Slika 15) za koje će se generirati fasade (Slika 16)

**{LDAP\_korisničko\_ime}\_zadaca\_4\_2**: Kreiranje modula za poslovnu logiku i korištenje EJB: SessionBean (Stateful, Stateless i Singleton)

1. Kreiranje projekta **{LDAP\_korisničko\_ime}\_zadaca\_4\_2** kao Java EE/EJB Module , server Glassfish, Java EE verzija: Java EE 7
2. kreirati paket **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**
3. kopirati pakete **org.foi.nwtis.{LDAP\_korisnik}.web.podaci**, **org.foi.nwtis.{LDAP\_korisnik}.rest.klijenti** iz **{LDAP\_korisnik}\_zadaca\_3\_1**
4. kreirati Stateless SessionBean **MeteoAdresniKlijent** (New/Other/Enterprise JavaBeans/Session Bean/Session Type: Stateless/Create Interface (označiti ni jedan)) u **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**
5. dodati privatnu varijablu sa svojim podacima (apiKey), koja preuzima vrijednost iz datoteke postavki.
6. kreirati poslovnu operaciju void postaviKorisnickePodatke(String apiKey) (Insert Code/Add Business Method...) koja pridružuje korisničke podatke privatnim varijablama
7. kreirati poslovnu operaciju public MeteoPodaci dajVazeceMeteoPodatke(String adresa) (Insert Code/Add Business Method...)
8. operacija kreira objekt klase GMKlijent, poziva metodu getLocation(adresa), kreira objekt klase OWMKlijent i s dobivenom lokacijom poziva metodu getRealTimeWeather(...). Kopirati kod funkcije iz klase GeoMeteoWS iz **{LDAP\_korisničko\_ime}\_zadaca\_3\_1** i prilagoditi za pridruživanje apiKey.
9. preuzeti klasu MeteoPrognoza.java i prilagoditi paket
10. u **OWMKlijent** dodati metodu  
public MeteoPrognoza[] getWeatherForecast(String latitude, String longitude).  
Može se kreirati posebna klasa za podatke meteo prognoze. Na temelju metode getRealTimeWeather(...) napraviti poziv REST servisa openweathermap.org za prognozu vremena za 5 dana po 3 sata (<http://openweathermap.org/forecast>) i prilagoditi podatke za traženi tip podatka u odgovoru.
11. u **MeteoAdresniKlijent** kreirati poslovnu operaciju public Lokacija dajLokaciju(String adresa) (Insert Code/Add Business Method...)
12. operacija kreira objekt klase GMKlijent, poziva metodu getLocation(adresa) i vraća lokaciju
13. kreirati poslovnu operaciju public MeteoPrognoza[] dajMeteoPrognoze(String adresa) (Insert Code/Call Web Service Operation...) .
14. operacija kreira objekt klase klase OWMKlijent , poziva metodu dajLokaciju(String adresa) i s dobivenom lokacijom poziva metodu getWeatherForecast(...)

#### **{LDAP\_korisničko\_ime}\_zadaca\_4: Kreiranje Enterprise aplikacije**

1. Kreiranje projekta **{LDAP\_korisničko\_ime}\_zadaca\_4** kao Enterprise Application, server Glassfish, Java EE verzija: Java EE 7

2. otvoriti projekt

3. na Java EE Modules, Add Java EE Module... Označiti:

{LDAP\_korisničko\_ime}\_zadaca\_4\_1, {LDAP\_korisničko\_ime}\_zadaca\_4\_2 i OK

4. kopirati datoteku **glassfish-resources.xml** iz projekta {LDAP\_korisničko\_ime}\_zadaca\_4\_1 (Files/ {LDAP\_korisničko\_ime}\_zadaca\_4\_1/src/conf/META-INF) u projekt {LDAP\_korisničko\_ime}\_zadaca\_4 (Files/ {LDAP\_korisničko\_ime}\_zadaca\_4/src/conf/)

**{LDAP\_korisničko\_ime}\_zadaca\_4\_3**: Kreiranje web modula za korisničko sučelje

1. Kreiranje **{LDAP\_korisničko\_ime}\_zadaca\_4\_3** kao Java Web aplikaciju, za Enterprise aplikaciju: **{LDAP\_korisničko\_ime}\_zadaca\_4**, server Glassfish, Java EE verzija: Java EE 7 (Frameworks: JSF), Libraries/Server Library: JSF 2.0, Configuration/JSF Servlet URL pattern: /faces/\* , Preferred Page Language: Facelets
2. kreirati paket **org.foi.nwtis.{LDAP\_korisnik}.web.zrna**
3. kopirati paket **org.foi.nwtis.{LDAP\_korisnik}.web.podaci** iz **{LDAP\_korisnik}\_zadaca\_4\_2**
4. kreirati JSF konfiguracijsku datoteku (New/Other/JavaServer Faces/ JSF Faces Configuration)
5. kreirati predložak pod nazivom **predlozak.xhtml** (New/Other/JavaServer Faces/Facelets Template) Layout Style: CSS, izbor izgleda sa zaglavljem, podnožjem i lijevim izbornikom. U <title>...</title> umetnuti <ui:insert name="naslov">Naslov</ui:insert> tako da se naslov može postaviti u pojedinom pogledu. Za bottom ime prezime - NWTiS &copy; 2016. U <h:outputStylesheet name="./css..."> obrišite u name ./ tako da bude name="css/..."
6. kreirati JSF stranicu za korištenje predloška pod nazivom **pregledAdresa.xhtml** (New/Other/JavaServer Faces/Facelets Template Client), odabrati predložak predlozak.xhtml, Generated Root Tag: <html>. Za naslov NWTiS - Pregled adresa i prognoza vremena. Sadržaj za top staviti isto kao naslov, za left poveznicu na index.xhtml i na pregledDnevnika.xhtml, Pogled služi za pregled i pretraživanje podataka o adresama i prognozama
7. selektirati sadržaj **pregledAdresa.xhtml** i kopirati u **index.xhtml** . Promijeniti potrebne elemente (naslov, poveznicu,...)
8. kreirati JSF stranicu za korištenje predloška pod nazivom **pregledDnevnika.xhtml** (New/Other/JavaServer Faces/Facelets Template Client), odabrati predložak predlozak.xhtml, Generated Root Tag: <html> .... Pogled služi za pregled dnevnika.
9. dodati navigaciju u faces-config.xml

- 10.izgraditi i izvršiti aplikaciju **{LDAP\_korisničko\_ime}\_zadaca\_4**. Provjeriti izgled za sve pogleda.
- 11.kreirati JSF ManageBean **OdabirAdresaProгноza** (Session scope) koji implementira sučelje Serializable
- 12.dodati varijable za novu adresu, popis aktivnih adresa, lista odabranih adresa za dodavanje, popis kandidiranih adresa,lista odabranih adresa za brisanje i pregled prognoza, ažurirana Adresa, lista prognoza, prikaz ažuriranja adresa, prikaz pregleda prognoza. Popisi imaju tip podatka Map<String, Object>, a liste imaju tip podatka List<String>
- 13.dodati settere i gettere za varijable
- 14.u **pregledAdresa.xhtml** dodati obrazac (slika 19) s elementima za unos nove adrese, izbor adresa, pregled izabranih adresa, ažuriranje adrese i pregled prognoza. Predvidjeti element za poruke o pogreškama. Elementi koji trebaju biti sakriveni koriste atribut rendered koji putem UEL ispituje stanje svoje pridružene varijable (true, false). Potrebne su kontrole (gumb) za pojedine akcije. Za svaki gumb potrebno je dodati podršku za ajax (<f:ajax ..> pri čemu su važna dva atributa:
  1. execute - kojim se određuje koje elemente (na bazi id) je potrebno osvježiti, učitati prije nego što se pozove akcija
  2. render - kojim se određuju koje elemente je potrebno osvježiti nakon akcije
- 15.Dodati projekt {LDAP\_korisničko\_ime}\_zadaca\_4\_3 u Enterprise aplikaciju ukoliko to nije napravljeno do sad
- 16.u **OdabirAdresaProгноza** dodati poziv EJB AdreseFacade (Insert Code .../Call Enterprise Bean i odabrati {LDAP\_korisničko\_ime}\_zadaca\_4\_1/AdreseFacade)
- 17.u **OdabirAdresaProгноza** dodati poziv EJB AdreseFacade (Insert Code .../Call Enterprise Bean i odabrati {LDAP\_korisničko\_ime}\_zadaca\_4\_2/MeteoAdresniKlijent)
- 18.dodati metode koje će izvoditi akcije dodavanja nove adrese, ažuriranja odabrane adrese, dodavanje odabrane adrese(a) u popis kandidiranih adrese, izbacivanje odabrane adrese(a) iz popisa kandidiranih adresa, pregled prognoza za odabranu adresu itd
- 19.dodati kod kojim se preuzimaju podaci o adresama putem AdreseFacade
- 20.dodati kod kojim se dodaje adresa putem AdreseFacade
- 21.dodati kod kojim se ažurira adresa putem AdreseFacade. U AdreseFacade treba dodati metodu za traženje adrese na temelju adrese jer nije poznat id adrese.
- 22.dodati akcije u pregledAdresa.xhtml
- 23.da bi se vidjele SQL komande koje izvršava EclipseLink može se dodati u persistence.xml (<http://eclipse.org/eclipselink/> <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>
```

24. u **pregledDnevnika.xhtml** dodati obrazac s elementima za filtriranje podataka: korisnik, ipadresa, trajanje, status. Potrebna je kontrola (gumb) za pokretanje pregleda uz filtriranje. Za gumb potrebno je dodati podršku za ajax (<f:ajax ..>
25. kreirati JSF ManageBean **PregledDnevnika** (Session scope) i koji implementira sučelje Serializable
26. dodati potrebne varijable za podatke iz dnevnika
27. dodati gettere i settere
28. dodati poziv EJB AdreseFacade (Insert Code .../Call Enterprise Bean i odabrati {LDAP\_korisničko\_ime}\_zadaca\_4\_2/DnevnikFacade)
29. dodati metodu koja će izvoditi akciju preuzimanja podataka iz dnevnika
30. dodati kod kojim se preuzimaju podaci iz dnevnika putem dnevnikFacade
31. Preporučuju se:
  1. za JSF [http://elf.foi.hr/pluginfile.php/8314/mod\\_resource/content/0/DZone/rc058-010d-jsf2.pdf](http://elf.foi.hr/pluginfile.php/8314/mod_resource/content/0/DZone/rc058-010d-jsf2.pdf)
  2. za Ajax <http://www.coreservlets.com/JSF-Tutorial/jsf2/#Ajax>
  3. za Criteria API <http://docs.oracle.com/cd/E19798-01/821-1841/gjitr/index.html>, [http://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic\\_JPA\\_Development/Querying/Criteria](http://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic_JPA_Development/Querying/Criteria)