

Vježba 11. / Zadaća 3.

Naziv: sustav aplikacija s korištenjem web servisa openweathermap.org i Google Maps API

Sustav je sastoji od dva projekta ({LDAP_korisničko_ime}_zadaca_3_1 i {LDAP_korisničko_ime}_zadaca_3_2). Prvi projekt ima tri dijela:

1. u korisničkom dijelu potrebno je unositi pojedinačne adrese za koje će se preuzimati meteorološki podaci. Prvo se unese adresa, zatim se pokrene akcija koja preuzima njene geolokacijski podaci putem Google Maps API. Slijedi pokretanje akcije za prikaz geolokacijski podataka. Nakon toga je akcija za spremanje podataka o adresi u tablicu baze podataka (Java DB, adrese, SQL datoteka za kreiranje tablice priložena je uz zadaću). Zadnja akcija je preuzimanje važećih meteoroloških podataka za tu adresu na bazi njenih geolokacijskih podataka te njihov prikaz na ekranu korisnika.
2. u pozadinskoj dretvi preuzimaju se meteorološki podaci putem REST web servisa openweathermap.org za izabrani skup adresa i pohranjuju se u tablicu u bazi podataka (Java DB, meteo)
3. pruža SOAP web servis za meteorološke podatke spremljenih adresa. Dio operacija se temelji na podacima koje se nalaze u tablici u bazi podataka, a ostale na pozivu REST web servisa openweathermap.org
4. pruža dva REST web servis za podatke o spremljenim adresama:
 1. osnovna adresa - daje popis svih adresa u application/json formatu
 2. na bazi putanje {id} za izabranu adresu (id je identifikator adrese u tablici adrese u bazi podataka. Vraća podatke u application/json formatu.

Drugi projekt sastoji se od korisničkog dijela u kojem se preuzimaju adrese za koje se prikupljaju meteorološki podaci u prvom projektu te se prikazuju u obliku padajućeg izbornika s mogućim odabirom više elemenata. Ako se odabere samo jedna adresa tada na raspolaganju stoje dvije mogućnosti:

- aktivirati gumb kojim se pokreće akcija koja će preuzeti sve pohranjenje meteorološke podatke za tu adresu putem SOAP web servisa iz prvog projekta. Preuzeti meteorološki podaci prikazuju se u obliku tablice.
- aktivirati gumb kojim se pokreće akcija koja će preuzeti vremensku prognozu za sljedećih 5 dana u intervalu od 3 sata na temelju poziva REST web servisa openweathermap.org (<http://openweathermap.org/forecast5>, npr. <http://api.openweathermap.org/data/2.5/forecast?lat=46.3076267&lon=16.3382566&appid=nnn>)

APPID za openweathermap treba biti spremljen u konfiguracijskoj datoteci.

Ako korisnik odabere minimalno dvije adrese može aktivirati gumb kojim se pokreće akcija kojom se putem REST web servisa iz prvog projekta preuzimaju važaći (zadnje upisani) meteorološki podaci za

odabrane adrese te se prikazuju u obliku redova blok elemenata.

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_3.zip i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka i popunjeni obrazac za zadaću pod nazivom {LDAP_korisničko_ime}_zadaca_3.doc (u korijenskom direktoriju projekta).

Naziv projekta: {LDAP_korisničko_ime}_zadaca_3

1. Kreiranje 11. vježbe/zadace 3 (direktorij {LDAP_korisničko_ime}_zadaca_3). U nastavku se direktorij za vježbu simbolički označava kao {vježba}).
2. Kreiranje projekta {LDAP_korisničko_ime}_zadaca_3_1 kao Java Web aplikaciju, server Tomcat 8.*, Java EE verzija: Java EE 7, kontekst {LDAP_korisničko_ime}_zadaca_3_1, (Frameworks: Ne)
3. kreirati pakete org.foi.nwtis.{LDAP_korisnik}.ws.klijenti, org.foi.nwtis.{LDAP_korisnik}.ws.serveri, org.foi.nwtis.{LDAP_korisnik}.rest.klijenti, org.foi.nwtis.{LDAP_korisnik}.rest.serveri, org.foi.nwtis.{LDAP_korisnik}.web.podaci, org.foi.nwtis.{LDAP_korisnik}.web.slusaci
4. Kreirati direktorij lib u projektu
5. Kopiranje biblioteke dist\vjezba_03_2.jar iz projekta **vjezba_03_2** u direktorij lib
6. Kopiranje biblioteke dist\vjezba_06_1.jar iz projekta **vjezba_06_1** u direktorij lib
7. Dodavanje biblioteke (jar/folder) lib\vjezba_03_2.jar u projekt
8. Dodavanje biblioteke (jar/folder) lib\vjezba_06_1.jar u projekt
9. Dodavanje biblioteka Java DB Driver, Java EE7 API Library, JAX-RS, JAX-WS, Jersey
10. Kopirati glassfish-4.1.1\glassfish\modules\javax.json.jar u direktorij lib
11. Dodavanje biblioteke (jar/folder) lib\javax.json.jar u projekt
12. Preuzeti primjere datoteke konfiguracija za JavaDB (NWTiS.db.config_2.xml) iz vjezba_06_2 te kopirati u direktorij WEB-INF
13. kreirati **web.xml** (New file/Web/Standard Deployment Descriptor) i upisati u web.xml inicijalni parametar konteksta **konfiguracija**
14. kreirati slušaca konteksta **SlusacAplikacije** (anotirani bez uključivanja u web.xml) i kopirati kod funkcije iz vjezba_08_1 ili {LDAP_korisničko_ime}_zadaca_2 Vrijednost ServletContext spremi u varijablu klase kako bi se jednostavno pristupilo podacima iz konteksta (aplikacije). Dodati getter za varijablu.
15. izgraditi, isporučiti, izvršiti i testirati aplikaciju
16. izvršiti poziv REST web servisa Google Maps

API [http://maps.google.com/maps/api/geocode/json?](http://maps.google.com/maps/api/geocode/json?address=xxx&sensor=false)
[address=xxx&sensor=false](http://maps.google.com/maps/api/geocode/json?address=xxx&sensor=false) (umjesto xxx upusati adresu po želji)
npr. [http://maps.google.com/maps/api/geocode/json?address=Croatia,Vara](http://maps.google.com/maps/api/geocode/json?address=Croatia,Vara%C5%BEdin,Pavlinska%20&sensor=false)
[%C5%BEdin,Pavlinska%20&sensor=false](http://maps.google.com/maps/api/geocode/json?address=Croatia,Vara%C5%BEdin,Pavlinska%20&sensor=false)

17. preuzeti geolokacijske podatke (latitude i longitude)
18. izvršiti poziv REST servisa [openweathermap.org](http://api.openweathermap.org/data/2.5/weather?lat=aaa.aaa&lon=bbb.bbb&units=metric&lang=hr&APIKEY=nnn) za važeće meteorološke podatke [http://api.openweathermap.org/data/2.5/weather?](http://api.openweathermap.org/data/2.5/weather?lat=aaa.aaa&lon=bbb.bbb&units=metric&lang=hr&APIKEY=nnn)
[lat=aaa.aaa&lon=bbb.bbb&units=metric&lang=hr&APIKEY=nnn](http://api.openweathermap.org/data/2.5/weather?lat=aaa.aaa&lon=bbb.bbb&units=metric&lang=hr&APIKEY=nnn) (umjesto aaa.aaa upisati latitude, umjesto bbb.bbb upisati longitude, umjesto nnn upisati APPID)
19. preuzeti priložene klase za servis (Klase: GMRESTHelper, OWMRESTHelper, MeteoPodaci, Lokacija, Adresa)
20. kreirati novi SOAP web servis **GeoMeteoWS** (New/Other/Web Services/Web Service) u paketu `org.foi.nwtis.{LDAP_korisnik}.ws.serveri`
21. slijedi prikaz dijaloškog okvira (Slika 1.) u vezi uključivanja GEOMETEO web servisa u projekt. Odabrati Yes.
22. otvoriti klasu **GeoMeteoWS** i odabrati Design
23. kliknuti na operaciju hello i obrisati operaciju hello
24. promijeniti u Properties za projekt Source/Binary/Format u 1.8 (ako nije)
25. preuzeti sa moodla adrese.sql i kreirati tablicu u JavaDB baza `nwtis_g1`
26. dodati operaciju `dajSveAdrese()`, a vraća `java.util.List<Adresa>`
27. izgraditi i isporučiti aplikaciju
28. testirati web servis, desni klik na web servis i Test Web Service, otići na adresu http://localhost:xxxx/{LDAP_korisnicko_ime}_zadaca_3_1/GeoMeteoWS?WSDL i upamtiti adresu
29. testirati operaciju web servisa. Kartica Services/Web Services pa desna tipka na mišu Create Group/NWTiS, desna tipka na mišu pa Add Web Service... i kopirati adresu WSDL-a. Otvoriti web servis i odabrati operaciju, desna tipka na mišu i Test Method i Submit
30. dodati `index.jsp`
31. u `web.xml` promijeniti da je `index.jsp` početna stranica (Pages/Welcome Files)
32. obrisati `index.html`
33. dodati u `index.jsp` obrazac za unos adrese i tri gumba za akcije:
 1. dohvat geo podataka
 2. spremanje podataka o adresi u tablicu baze podataka
 3. dohvat važećih meteopodataka upisane adrese
34. kreirati servlet `DodajAdresu` (paket `org.foi.nwtis.{LDAP_korisnik}.web`) koji će primiti podatke i obaviti potrebne akcije
35. Kreirati dretvu `PreuzmiMeteoPodatke` koja će za spremljene adrese preuzimati meteorološke podatke i spremati u odgovarajuću tablicu pod nazivom `meteo`.

Dretvu treba startati u slušaču aplikacije kod kreiranje konteksta. **Interval dretve određen je konfiguracijskim podatkom.**

36.preuzeti datoteku SQL Tablica meteo i izvršiti u Java DB

37.dodati tri dodatne operacije web servisa GeoMeteoWS:

1. `dajVazeceMeteoPodatkeZaAdresu(String)` - spaja se direktno na REST servis i vraća trenutno važeće meteo podatke za danu adresu
2. `dajZadnjeMeteoPodatkeZaAdresu(String)` - vraća posljednje spremljene meteo podatke iz baze podataka, ukoliko nema podataka vraća null
3. `dajSveMeteoPodatkeZaAdresu(String)` - vraća sve spremljene podatke iz baze podataka za unesenu adresu, ukoliko nema podataka vraća null

38.testirati kreirane operacije

39.kreirati RESTful web servis u paketu `org.foi.nwtis.{LDAP_korisnik}.rest.serveri` (Other/Web Services/RESTful Web Services from Patterns/Container-Item/ (Slika 2) s nazivom resursa **MeteoREST**, klasa **MeteoRESTResource**, klasa za kontejner **MeteoRESTResourceContainer**, putanja `{id}` i putanja kontejnera `/meteoREST`, tip `application/json`. (Slika 3.) Odabrati Create default Jersey REST servlet adaptor in `web.xml` (Slika 4.)

40.u klasi `MeteoREST_Container` u metodi `getHtml()` ili `getJson()` staviti da vraća popis svih adresa u `application/json` formatu. Potrebno je napraviti upit prema bazi podataka.

41.u klasi `MeteoREST` u metodi `getJson()` staviti da vraća na bazi putanje `{id}` posljednje spremljene podatke izabrane adrese iz baze podataka. Vraća podatke u `application/json` formatu.

42.izgraditi i isporučiti aplikaciju

43.testirati RESTful web servis (RESTFul Test Services/Test RESTFul Test Services)

44.izvršiti RESTful `http://localhost:xxxx/{LDAP_korisnicko_ime}_zadaca_3_1/{web}resources/meteoREST/`

45.izvršiti RESTful `http://localhost:xxxx/{LDAP_korisnicko_ime}_zadaca_3_1/{web}resources/meteoREST/1` i druge id adresa

46.Kreiranje projekta `{LDAP_korisnicko_ime}_zadaca_3_2` kao Java Web aplikaciju, server Glassfish, Java EE verzija: Java EE 7, kontekst `{LDAP_korisnicko_ime}_zadaca_3_2`, (Frameworks: JSF), Libraries/Server Library: JSF 2.*, Configuration/JSF Servlet URL pattern: `/faces/*`, Preferred Page Language: Facelets

47.kreirati pakete `org.foi.nwtis.{LDAP_korisnik}.ws.klijenti`, `org.foi.nwtis.{LDAP_korisnik}.web.zrna`

48.kreirati novog klijenta web servisa (New/Other/Web Services/Web Service Client) za Web servis `MeteoWS` koji ima adresu WSDL na bazi projekta `{LDAP_korisnicko_ime}_zadaca_3_1`

49.kreirati klasu **MeteoWSKlijent** u paketu `org.foi.nwtis.{LDAP_korisnik}.ws.klijenti`

50.napraviti poziv operacije `dajSveAdrese` na web servisu `GeoMeteoWS` (Insert Code/Call Web Service Operation...)

51.promijeniti za metodu iz private u public

52.kreirati JSF MenagesBean **OdabirAdresa** (razmisliti primjeni session ili request)

53.dodati varijablu za preuzimanje adresa List<Adresa>, dodati gettere i settere

54.kreirati JSF odabirAdresa

55.dodati <h:dataTable... >za prikaz preuzetih adresa

56.....

57.izgraditi i isporučiti aplikaciju