

# Projekt: NWTiS\_2013\_2014

## Sustav se treba sastojati od sljedećih aplikacija:

1. web aplikacija ({korisnicko\_ime}\_aplikacija\_1) u pozadinskom modu (tj. servletu s automatskim startanjem ili putem slušača), pokreće dretvu (konfiguracijom se određuje pravilni vremenski interval (jedinica je sekunda) preuzimanja podataka, npr. 30 sec, 100 sec, 2 min, 10 min, 30 min, 60 min, ...) koja preuzima važeće meteorološke podatke od WeatherBug REST web servisa (u prilogu se nalazi opis postupka) za izabrani skup adresa iz svijeta. U tablici {korisnicko\_ime}\_adrese nalaze se podaci o adresama (id, adresa, koordinate, korisnik tko je zatražio i sl) koje su tražene. Potrebno je spremati minimalno 5 meteoroloških podataka (temp, tlak, vlaga, vjetar, kiša, snijeg i sl) u bazu podataka. Upravljanje pozadinskom dretvom provodi se putem primitivnog poslužitelja koji ima ulogu socket servera na određenom portu (postavkom se određuje). Kada poslužitelj primi zahtjev od klijenta zapisuje podatke u dnevnik rada u bazu podataka. Na svaki zahtjev odgovara se u obliku

OK nn [dodatni dio koji ovisi o vrsti zahtjeva]

nn je status odgovora za primljeni zahtjev. Za administratorski zahtjev prvo treba obaviti autentikaciju korisnika prema bazi podataka te se ako je u redu, šalje email poruka (adresa primatelja i predmet poruke određuju se postavkama) u MIME tipu „text/plain“ s informacijama o komandi (vrijeme izvršavanja, trajanje prethodnog stanja, broj primljenih, neispravnih i izvršenih korisničkih komandi). Provođenje spomenutih operacija ne smije utjecati na sposobnost poslužitelja da primi nove zahtjeve. Zahtjev se temelji na komandama (isključivo u jednom retku), koje mogu biti:

administratorske:

```
USER korisnik; PASSWD lozinka; {PAUSE; | START; | STOP;  
| ADD „adresa“; | TEST „adresa“; | GET „adresa“; | ADD  
korisnik1; NEWPASSWD lozinka1;}
```

korisničke:

```
USER korisnik; GET „adresa“;
```

Objašnjenje komandi:

- USER korisnik; PASSWD lozinka; – autentikacija korisnika s administratorskim ovlastima. Prelazi na obradu komande ako postoji korisnik i odgovara lozinka, odnosno vraća ERR 30 ako ne postoji korisnik ili ne odgovara lozinka.
- PAUSE; – privremeno prekida preuzimanje meteoroloških podataka od sljedećeg ciklusa (i dalje može preuzimati korisničke komande). Vraća OK 10 ako nije bio u pauzi, odnosno ERR 40 ako je bio u pauzi.
- START; – nastavlja s preuzimanjem meteoroloških podataka od sljedećeg ciklusa. Vraća OK 10 ako je bio u pauzi, odnosno ERR 41 ako nije bio u pauzi.
- STOP; – potpuno prekida preuzimanje meteoroloških podataka i preuzimanje korisničkih komandi. Vraća OK 10 ako nije bio u postupku prekida, odnosno ERR 42 ako je bio u postupku prekida.

- ADD „adresa“; – dodaje adresu za preuzimanje meteoroloških podataka od sljedećeg ciklusa. Vraća OK 10 ako ne postoji u evidenciji za koje se preuzimaju meteorološki podaci, odnosno ERR 50 ako postoji i već se preuzimaju podaci.
- TEST „adresa“; – vraća status za „adresa“ tj. da li se za nju preuzimaju podaci. Vraća OK 10 ako postoji u evidenciji za koje se preuzimaju meteorološki podaci, odnosno ERR 51 ako ne postoji.
- GET „adresa“; – vraća zadnje meteorološke podatke za „adresa“. Vraća OK 10 i

```
TEMP {-}nn.nn VLAGA nn.nn TLAK nnnn.nn GEOSIR {-}
nnn.nnnnnn GEODUZ {-}nnn.nnnnnn
```

ako postoji u evidenciji za koje se preuzimaju meteorološki podaci, odnosno, ERR 52 ako ne postoji.

- ADD korisnik1; NEWPASSWD lozinka1; - dodaje korisnika s lozinkom u tablicu korisnika uz administratorske ovlasti.

Drugi zadatak web aplikacije je pružanje SOAP web servisa za prikupljene meteorološke podatke kao što su operacije: popis svih adresa za koje se prikupljaju meteorološki podaci, trenutni meteorološki podaci za izabranu adresu, rang lista (prvih n) adresa za koje je prikupljeno najviše meteoroloških podataka, posljednjih n meteoroloških podataka za izabranu adresu, meteorološki podaci za adresu u nekom vremenskom intervalu (od Date, do Date) i sl. Potrebno je pripremiti u NetBeans-ima za testiranje vlastitog web servisa (u Web services kreirati novu grupu NWTiS i dodati vlastiti servis). Treći zadatak je vidljivi dio web aplikacije odnosno korisnički dio koji treba sadržavati unos i pregled adresa za koje se prikupljaju meteo podaci (na bazi adrese putem Google Maps API preuzimaju se podaci o geolokaciji, opis u prilogu), pregled prikupljenih meteoroloških podataka, pregled zahtjeva za server koji upravlja dretvom i pregled dnevnika korisničkih zahtjeva (kasnije opisano) putem JSP (bez skripleta) i displaytag-a uz filtriranje podataka (elementi za filtriranje: **izabrana adresa**, vremenski interval (od-do, u formatu dd.dd.MM.yyyy hh.mm.ss) u kojem su prikupljeni podaci i sl.), status i straničenje (izbor od 5, 10, 20, 50, 100, svi). Filtriranje treba omogućiti korištenje od ni jednog pa do svih elemenata u istom pogledu. Korisnički dio web aplikacije treba biti zaštićen putem web/servlet kontejnera na bazi obrasca/forme za prijavljivanje uz pomoć JDBC pristupa do baze podataka te osiguranja sigurnog kanala (SSL uz vlastiti certifikat s imenom i prezimenom studenta). Četvrti zadatak je vođenje dnevnika korisničkih zahtjeva vidljivog dijela aplikacije na bazi aplikacijskog filtera uz zapis trajanje obrade zahtjeva u bazu podataka. U tablici prikupljenih meteoroloških podataka trebaju biti podaci za minimalno 20 adresa i za svaki od njih minimalno 100 preuzetih podataka u vremenskom intervalu većem od zadnjih 6 sati.

2. enterprise aplikacija ({korisnicko\_ime}\_aplikacija\_2) koja ima EJB i Web module. Aplikacija u pozadinskom modu (tj. servletu s automatskim startanjem ili putem slušača) pokreće dretvu (konfiguracijom se određuje pravilni vremenski interval rada (jedinica je sekunda), npr. 5 sec, 20 sec, 100 sec, ...) koja provjerava u poštanskom sandučiću (adresa poslužitelja, korisničko ime i lozinka definiraju se u konfiguracijskoj datoteci) pristiglu poštu. Od pristiglih email poruka one koje imaju predmet poruke prema postavkama i MIME tip „text/plain“ nazivamo NWTiS porukama. Obradene NWTiS poruke treba prebaciti u mapu/direktorij prema postavkama za NWTiS poruke. Ostale ne-NWTiS poruke treba prebaciti u mapu/direktorij prema postavkama za ne-NWTiS poruke. Na kraju svake iteracije obrade email poruka treba poslati JMS poruku (naziv reda čekanja NWTiS\_{korisnicko\_ime}\_1) s podacima o vremenu početka i završetka rada, broju pročitanih poruka, broju NWTiS poruka i sl. Poruka treba biti u obliku ObjectMessage, pri

čemu je naziv klase proizvoljan, a njena struktura treba sadržavati potrebne podatke koji su prethodno spomenuti.

Drugi zadatak je korisnički dio odnosno web modul, koji treba realizirati putem JSF (facelets) ili PrimeFaces uz minimalno dvojezičnu varijantu (hrvatski i engleski jezik). To znači da svi statički tekstovi u pogledima trebaju biti označeni kao „labele“ i dobiti jezične prijevode. Jezik se odabire na početnoj stranici aplikacije. U javnom dijelu aplikacije može se dobiti pregled adresa za koje se u {korisnicko\_ime}\_aplikacija\_1 prikupljaju meteorološki podaci, zadnji meteorološki podaci za izabranu adresu iz skupa adresa i sl. Kod pregleda treba postojati straničenje i filtriranje adresa. Pri tome se koristi Ajax za preuzimanje podataka. Privatni dio treba se štititi putem aplikacijskog filtra koji propušta samo prijavljenog korisnika, a ostale prebacuje na prijavljivanje. Prvi korak za novog korisnika je registracija. Korisnik nakon prijavljivanja mora kreirati svoj meteorološki portfelj ukoliko ga još nema, s time da ih može imati više. Minimalni i maksimalan broj adresa za portfelj određuje se postavkama. Meteorološki portfelji ima svoj identifikator (tj. naziv kao jednu riječ). Korisnik kod definiranja pojedinog portfelja određuje listu adresa koje ulaze u portfelj, na bazi filtriranja postojećih adresa ili unosi nove adrese. Za svaku novu adresu provjeravaju se podaci u odnosu na definiranu operaciju web servisa iz {korisnicko\_ime}\_aplikacija\_1. Za adresu za koju se ne prikupljaju podaci potrebno je poslati JMS poruku (naziv reda čekanja NWTiS\_{korisnicko\_ime}\_2) u kojoj se nalazi адреса za koji treba početi prikupljati meteorološke podatke. Poruka treba biti u obliku ObjectMessage, pri čemu je naziv klase proizvoljan a njena struktura treba sadržavati potrebne podatke koji su prethodno spomenuti. Dodatni bodovi mogu se dobiti ako se za prikaz izabranih adresa za portfelj i njihovih podataka (npr. koordinate, trenutna temp i sl.) koristi Google Maps JavaScript API.

Prijavljeni korisnik može pregledavati trenutne, povijesne (od-do) i sl. meteorološke podatke za adrese izabranog portfelja, uz straničenje, na bazi web servisa koje pruža {korisnicko\_ime}\_aplikacija\_1.

Administrator sustava može pregledavati poruke u poštanskom sandučiću (adresa poslužitelja, korisničko ime i lozinka definiraju se u postavkama). Pri tome može izabrati mapu/direktorij u kojem pregledava poruke. Nazivi mapa preuzimaju se od mape poštanskog sandučića korisnika email poslužitelja. Izabrane poruke može obrisati. Administrator sustava može pregledavati dnevnik rada uz filtriranje (vremenski interval od-do, korisnik i sl) i straničenje.

Četvrti zadatak je pružanje popisa aktivnih (trenutno prijavljenih) Web korisnika, popisa portfelja izabranog korisnika (na bazi putanje), popis adresa iz izabranog portfelja izabranog korisnika (na bazi putanje) putem RESTful web servisa (XML ili JSON).

Pristup do podataka u bazi podataka treba biti realiziran putem ORM-a tj. putem session, entity bean-ova i criteria API.

3. enterprise aplikacija ({korisnicko\_ime}\_aplikacija\_3) koja ima EJB i Web module. Aplikacija preuzima dvije vrste JMS poruka (kao MessageDriven Bean) za stanje obrade email poruka i adresa za koji treba početi preuzimati meteorološke podatke. Kod primitka JMS poruke za adresu potrebno je provjeriti status prikupljanja meteoroloških podataka za traženu adresu putem slanja komande USER korisnik; PASSWD lozinka; TEST „adresa“; socket serveru iz {korisnicko\_ime}\_aplikacija\_1 (postavkom se određuje adresa i port). Podaci za administratorsko korisničko ime i lozinku definiraju se u postavkama. Ukoliko se još ne prikupljaju podaci, šalje se zahtjev USER korisnik; PASSWD lozinka; ADD „adresa“; da se počne s preuzimanjem meteoroloških podataka za traženu adresu.

Primljene JMS poruke spremaju se u memoriji aplikacije. Ako aplikacija prestaje s radom, potrebno je poruke serijalizirati na vanjski spremnik (naziv datoteke u postavkama, smještena u WEB-INF direktoriju). Kada se aplikacija podiže potrebno je učitati serijalizirane poruke (ako postoji datoteka) u memoriju aplikacije.

Drugi zadatak je korisnički dio odnosno web modul koji treba realizirati putem JSF (facelets) ili PrimeFaces. On služi za pregled spremljenih JMS poruka. Pomoću websocketa treba obavijestiti pregled poruka da je stigla nova JMS poruka te treba osvježiti pregled. Korisnik može brisati izabrane poruke ili sve poruke. Korisnik može pozivom RESTful web servisa dohvatiti i ispisati podatke za aktivne korisnike {korisnicko\_ime}\_aplikacija\_2, za izabranog aktivnog korisnika ispisati nazive njegovih portfelja ili ispisati adrese za izabrani portfelj. Za jedan od prethodno ispisanih adresa može tražiti trenutne meteorološke podatke putem slanja komande USER korisnik; PASSWD lozinka; GET „adresa“; socket serveru iz {korisnicko\_ime}\_aplikacija\_1 te ih ispisati. Preporučuje se upravljački dio putem kojeg se mogu slati komande na socket server iz {korisnicko\_ime}\_aplikacija\_1, posebno za upravljanje preuzimanja meteoroloških podataka.

### **Projektna dokumentacija:**

dokumentacija.html koja je dostupna iz svih web aplikacija, sadrži opis projektnog zadatka i glavne odrednice projektnog rješenja. Potrebna je Java doc dokumentacija. Nije potrebna korisnička dokumentacija

## Instalacijska, programska i komunikacijska arhitektura sustava:

### `{korisnicko_ime}_aplikacija_1:`

- Web poslužitelj: Tomcat
- EE osobine: EE6 Web, uz obavezno korištenje web.xml i bez korištena EE6 osobina za filtere, slušače, servlete i sl.)
- korisničko sučelje: JSP
- baza podataka: MySQL - naziv nwtis\_{korisnickoime}\_bp\_1, treba sadržavati tablice: {korisnicko\_ime}\_adrese i ostale koje su potrebne za rad
- rad s bazom podataka: JDBC, SQL
- šalje email poruku
- daje socker server
- daje SOAP web servis za meteorološke podatke izabranih adresa
- koristi WeatherBug REST web servis za preuzimanje meteoroloških podataka
- koristi Google Maps API REST web servis za preuzimanje geolokacijskih podataka za adresu

### `{korisnicko_ime}_aplikacija_2:`

- Web poslužitelj: Glassfish
- EE osobine: EE7 (anotacije za filtere, slušače, servlete i sl.)
- korisničko sučelje: JSF (facelets) ili PrimeFaces
- baza podataka: JavaDB – naziv nwtis\_{korisnickoime}\_bp\_2, treba sadržavati podatke i tablice koje su potrebne za rad.
- rad s bazom podataka: ORM (EclipseLink), Criteria API
- James email poslužitelj
- obrađuje i pregledava email poruke
- šalje JMS poruku u red poruka: NWTiS\_{korisnicko\_ime}\_1 za statistiku obrade email poruka
- šalje JMS poruku u red poruka: NWTiS\_{korisnicko\_ime}\_2 za novu adresu
- koristi SOAP web servis za adrese i meteorološke podatke za izabrane adrese
- daje REST web servis za aktivne korisnike, njihove portfelje, adrese u portfelju

### `{korisnicko_ime}_aplikacija_3:`

- Web poslužitelj: Glassfish
- EE osobine: EE7 (anotacije za filtere, slušače, servlete i sl.)
- korisničko sučelje: JSF (facelets) ili PrimeFaces
- baza podataka: ne koristi bazu podataka.
- koristi JMS redove poruka NWTiS\_{korisnicko\_ime}\_1 i NWTiS\_{korisnicko\_ime}\_2 za preuzimanje, spremanje i pregled JMS poruka
- koristi websocket za osvježavanje pregleda poruka nakon prijema nove poruke
- koristi socker server provjeru i dodavanje adrese, preuzimanje meteoroloških podataka za adresu iz portfelja
- koristi REST web servis za pregled aktivnih korisnika, njihovih portfelja, adresa u portfelju

## Komunikacija između aplikacija:

- putem SOAP i REST web servisa, email poruka, JMS poruka i vlastitog jezika na bazi socket servera (NE putem baza podataka).

## Postupak za aktiviranje i korištenje web servisa WeatherBug:

1. Dokumentacija: <http://api-portal.earthnetworks.com/api-documentation>
2. Obaviti registraciju na: <http://api-portal.earthnetworks.com/user/login>
3. Prijaviti se i otići na myapps i tamo "kupiti" **Pro package** / free
4. Popuniti podatke za tvrtku
5. Aktivirati/dodati ključ
6. Zapisati podatke za Consumer Key: i Secret Key:
7. Servisi:
  - a. OAuth Security – opisana je u svakom dokumentu

Autentikacija se provodi slanjem korisničkog ključa (Consumer Key) i tajnog ključa (Secret Key). Ukoliko je u redu dobije se žeton (token) s vremenskim trajanjem koji se koristi u ostalim pozivima REST web servisa. Osnovni format vraćenih podataka je JSON, a može se podesiti da se vraća u XML format.

[https://thepulseapi.earthnetworks.com/oauth20/token?grant\\_type=client\\_credentials&client\\_id=xxx&client\\_secret=yyyy](https://thepulseapi.earthnetworks.com/oauth20/token?grant_type=client_credentials&client_id=xxx&client_secret=yyyy)

- b. Real-time Weather - <http://api-portal.earthnetworks.com/sites/default/files/docs/RealTimeWeather%20-%20UM48.pdf>

Dobiveni žeton (token) koristi se u pozivu REST web servisa za važeće meteorološke podatke. Osnovni format vraćenih podataka je JSON, a može se podesiti da se vraća u XML format.

[https://thepulseapi.earthnetworks.com/data/observations/v1/current?location=aaa.aaa,bbb.bab&locationtype=latitudelongitude&units=metric&cultureinfo=en-en&verbose=true&access\\_token=zzzz](https://thepulseapi.earthnetworks.com/data/observations/v1/current?location=aaa.aaa,bbb.bab&locationtype=latitudelongitude&units=metric&cultureinfo=en-en&verbose=true&access_token=zzzz)

## Google Maps API – za dobivanje geolokacijskih podataka za adresu

### JSON

<http://maps.google.com/maps/api/geocode/json?address=varazdin&sensor=false>

### XML

<http://maps.google.com/maps/api/geocode/xml?address=varazdin&sensor=false>

Prije slanja adrese važno je obaviti njeno pretvaranje u HTTP URL format pomoću funkcije `URLEncoder.encode(adresa, "UTF-8")`