

**Vježba: 10 – Zadaća 3. Web sustav "Informacije o aerodromima, lokacijama i meteorološkim podacima"**

1. Kreiranje direktorija roditeljskog projekta zadaće (direktorij {korisnik}/{LDAP\_korisnik}-zadaca\_3).  
U nastavku se direktorij za vježbu simbolički označava kao {zadaca}.

```
mkdir {korisnik}/{LDAP_korisnik}-zadaca_3
```

2. Postavljenje dozvola za rad na poslužitelju Glassfish EE i kopiranje upravljačkog programa za HSQLDB.

```
cd /opt/glassfish6/bin/  
sudo chmod go+rw *
```

```
cd /opt/glassfish6/glassfish/  
sudo chmod -R go+rw domains/
```

```
cd /opt/glassfish6/glassfish/domains/domain1/lib/  
cp /opt/hsqldb-2.6.1/hsqldb/lib/hsqldb.jar .
```

3. Otvoriti u Text Editor datoteku

```
/opt/glassfish6/glassfish/domains/domain1/config/domain.xml
```

4. Postaviti portove za poslužitelj Glassfish EE na 9090, 9191, i 4949. Spremiti datoteku.

```
<network-listeners>  
<network-listener port="9090" protocol="http-listener-1" transport="tcp" name="http-listener-1" thread-pool="http-thread-pool"></network-listener>  
<network-listener port="9191" protocol="http-listener-2" transport="tcp" name="http-listener-2" thread-pool="http-thread-pool"></network-listener>  
<network-listener port="4949" protocol="admin-listener" transport="tcp" name="admin-listener" thread-pool="admin-thread-pool"></network-listener>  
</network-listeners>
```

Pokretanje programa Eclipse IDE.

Priprema Glassfish EE poslužitelja za rad na vježbama:

5. Počinje se s dodavanjem Window/Preferences/Server/Runtime Environment/Add . Odaberi Payara/Payara i za Payara location: se upisuje /opt/glassfish6/, a za Java location: /lib/jvm/jdk-17.0.2. Staviti kao naziv Glassfish EE Server 6 (6.2.3). U inicijalnoj varijanti administratorski korisnički račun ima korisničko ime admin i nema lozinku, Debug port: 8011.
6. Pokrenuti poslužitelj Glassfish EE (Servers/Payara Server \*/Start ili Debug). Ako je sve u redu na konzolu Payara poslužitelja dobije se ispis:

```
Grizzly Framework 3.0.1 started in: 49ms - bound to [/0.0.0.0:9090]  
Grizzly Framework 3.0.1 started in: 1ms - bound to [/0.0.0.0:9191]  
Grizzly Framework 3.0.1 started in: 1ms - bound to [/0.0.0.0:4949]
```

Podaci pokazuju da se poslužitelj javlja na portu 9090 (HTTP), njegova sigurna veza je na portu 9191 (HTTPS), a administratorska konzola je na portu 4949.

7. Može se pogledati administratorska konzola na adresi <http://localhost:4949/>

## Vježba\_10\_1: Korijenski projekt

1. Create a sample project (skip archetype selection) – označiti

Location: {zadaca}

Group Id: **org.foi.nwtis.{LDAP\_korisnik}**

Artifact Id: **{LDAP\_korisnik}-zadaca\_3**

Version: **1.0.0**

Packaging: **pom**

Name: **{LDAP\_korisnik}-zadaca\_3**

2. Otvoriti **pom.xml** i dodati ispred `</project>` za module

```
        <properties>
            <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
            <maven.compiler.source>17</maven.compiler.source>
            <maven.compiler.target>17</maven.compiler.target>
        </properties>
    <modules>
<!--
        <module>{LDAP_korisnik}-zadaca_3_lib_03_1</module>
        <module>{LDAP_korisnik}-zadaca_3_lib_06_1</module>
        <module>{LDAP_korisnik}-zadaca_3_wa_1</module>
        <module>{LDAP_korisnik}-zadaca_3_wa_2</module>
-->
    </modules>
```

3. Kopirati {LDAP\_korisnik}-vjezba\_03\_1 i podesiti tako da postane {LDAP\_korisnik}-zadaca\_3\_lib\_03\_1. Postupak je upisan kod vježbe 4/zadaca 1. Isto Kopirati {LDAP\_korisnik}-vjezba\_06\_1 i podesiti tako da postane {LDAP\_korisnik}-zadaca\_3\_lib\_06\_1. Srediti da postanu aktivni moduli korijenskog projekta zadace.
4. Korijenski projekt zadace pripremiti za korištenje/izvršavanje (Run As/Maven install)

## Vježba\_10\_2: Web aplikacija br. 1

1. Kreiranje korijenskog direktorija projekta `{LDAP_korisničko_ime}-zadaca_3_wa_1`  
`mkdir {zadaca}/{LDAP_korisnik}-zadaca_3_wa_1`
2. New/Other.../Dynamic Web Project. Naziv projekta `{LDAP_korisničko_ime}-zadaca_3_wa_1`
3. MišD na projektu/Configure/Convert to Maven Project. Podaci su sljedeći:  
Group Id: `org.foi.nwtis.{LDAP_korisnik}`  
Artifact Id: `{LDAP_korisničko_ime}-zadaca_3_wa_1`  
Version: `1.0.0`  
Packaging: `war`  
Name: `{LDAP_korisničko_ime}-zadaca_3_wa_1`
4. Preuzeti datoteku `zadaca_3_1.dio.zip` i u `pom.xml` dodati sadržaj datoteke `pom_Cargo_za_Glassfish_EE_DZ3_1.txt` koja je priložena uz zadaću
5. Dodati dependency za group id: `jakarta.platform` artifact: `jakarta.jakartaee-api` i version: `9.1.0`
6. Dodati dependency za group id: `org.projectlombok` artifact: `lombok` i version `1.18.22`
7. Dodati dependency za group id: `org.hsqldb` artifact: `hsqldb` i version: `2.6.1`
8. Dodati dependency za group id: `mysql` artifact: `mysql-connector-java` i version: `8.0.28`
9. Dodati dependency za group id: `com.google.code.gson` artifact: `gson` i version `2.9.0`
10. Dodati dependency za group id: `org.foi.nwtis` artifact: `NWTIS_REST_lib` i version: `2.3.0`
11. Dodati dependency za projekte `{LDAP_korisnik}-zadaca_3_lib_03_1` i `{LDAP_korisnik}-zadaca_3_lib_06_1`
12. Kreirati pakete
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.dretve`
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.podaci`
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.wsock`
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.ws`
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.slusaci`
13. Kreirati klasu `Info` za `WebSocket`. Dodati anotaciju `@ServerEndpoint("/info")`. Odabrati kolekciju u koju će se spremati sjednice korisnika i inicijalizirati je. Varijabla je na razini klase. Dodati metodu `informiraj(String)` na razini klase koja šalje poruku svim sjednicama iz kolekcije. Dodati 4 metode s anotacijama. Kao na primjeru s predavanja, klasa `NadzorKorisnika`.
14. Kreirati klasu `Osvjezivač` kao `dretvu`. U metodi `start()` preuzeti iz postavki potrebne podatke i pridružiti ih varijablama. U `run()` metodi postaviti petlju koja se izvodi dok je ispunjen uvjet. Uzima se trenutno vrijeme na poslužitelj u traženom formatu. Dohvati se broj aerodroma za koje se preuzimaju podaci. Poziva se metoda `informiraj(String)` iz klase `Info`.
15. Pripremiti za korištenje/izvršavanje (Run As/Maven install)
16. Preporučuje se da je poslužitelj `Glassfish EE Server` pokrene u debug modu
17. Pripremiti za isporuku (Run As/Maven build...). Kod Goals: staviti `cargo:redeploy`, kod Profiles: `ServerEE-local`. Preporučuje se dodati kod naziva – `redeploy`

18. Analizirati rad web aplikacije

19. Kreirati klasu za slušača aplikacije, dodati učitavanje konfiguracijskih podataka kod kreiranja aplikacije itd. Može se kopirati iz prethodne vježbe 9. / zadatak 2.

20. Upisati u web.xml inicijalni parametar konteksta konfiguracija i dodati jednu od datoteka NWTiS.db.config\_1.xml ili NWTiS.db.config\_2.xml

21. Kreirati klasu WsAerodromi. Dodati anotaciju @WebService(serviceName = "aerodromi"). Dodati u klasu resurs za pristup do konteksta aplikacije:

```
@Resource  
private WebServiceContext wsContext;
```

22. Do postavki se dolazi putem (bolje je da se refaktorira u metodu):

```
ServletContext context = (ServletContext)  
    wsContext.getMessageContext().get(MessageContext.SERVLET_CONTEXT);  
pbp = (PostavkeBazaPodataka) context.getAttribute("Postavke");
```

23. Kreirati metode:

- public List<Aerodrom> dajSveAerodrome()
- public List<Aerodrom> dajAerodromeZaPratiti()
- public List<AvionLeti> dajPolaskeAerodroma(String icao)
- public List<AvionLeti> dajDolaskeAerodroma(String icao)
- public boolean dodajAerodromZaPratiti(String icao)
- public Aerodrom dajNajbliziAerodrom(Lokacija lokacija, String icao)

24. Sve metode treba anotirati s @WebMethod

25. Kreirati klasu WsMeteo. Dodati anotaciju @WebService(serviceName = "meteo"). Dodati u klasu resurs za pristup do postavki kao i kod WsAerodromi.

26. Kreirati metodu:

- public MeteoPodaci dajMeteo(String icao)

27. Metodu treba anotirati s @WebMethod

28. Metoda koristi klasu OWMKlijent za dohvat meteo podataka.

29. U pom.xml treba podesiti pakete kod generiranja klasa. Potražiti <!-- 1. dio početak --> i za <sei> postaviti vlastiti paket tj. korisničko ime umjesto {LDAP\_korisnik}

```
<sei>org.foi.nwtis.{LDAP_korisnik}.zadaca_3.ws.WsAerodromi</sei>  
<sei>org.foi.nwtis.{LDAP_korisnik}.zadaca_3.ws.WsMeteo</sei>
```

30. Pripremiti za korištenje/izvršavanje (Run As/Maven install)

31. Pogledati sadržaj direktorija target/generated-sources/wsdl i target/generated-sources/wsgen

32. Pripremiti za isporuku (Run As/Maven build...). Kod Goals: staviti cargo:redeploy, kod Profiles: ServerEE-local. Preporučuje se dodati kod naziva – redeploy

33. Otvoriti Admin konzolu i pogledati aplikacije

34. Za testiranje JAXWS servisa preporučuje se instalirati program SoapUI na računalo. Adresa <https://www.soapui.org/downloads/soapui/>

35. Nakon preuzimanje (pretpostavimo da se spremi na Downloads direktorij i današnji naziv datoteke je SoapUI-x64-5.7.0.sh) izvrši se komande za instaliranje SoapUI programa

```
sudo sh /home/NWTiS_{n}/Downloads/ SoapUI-x64-5.7.0.sh
```

36. U Admin konzoli otvori se dio s aplikacijama te se za aplikaciju {LDAP\_korisnik}-zadaca\_3\_wa\_1 preuzme adresa WSDL za krajnju točku aerodromi. U SoapUI kreira se novi SOAP projekt za koji se upisuje naziv „Zadaća 3.“ i stavlja se preuzeti WSDL. Sada se mogu slati zahtjevi prema JAXWS servisu „aerodromi“.
37. U Admin konzoli otvori se dio s aplikacijama te se za aplikaciju {LDAP\_korisnik}-zadaca\_3\_wa\_1 preuzme adresa WSDL za krajnju točku meteo. U SoapUI dodaje se novi WSDL za meteo. Sada se mogu slati zahtjevi prema JAXWS servisu „meteo“.

## Vježba\_09\_4: Web aplikacija br. 2

1. Kreiranje korijenskog direktorija projekta `{LDAP_korisničko_ime}-zadaca_3_wa_2`  
`mkdir {zadaca}/{LDAP_korisnik}-zadaca_3_wa_2`
2. New/Other.../Dynamic Web Project. Naziv projekta `{LDAP_korisničko_ime}-zadaca_3_wa_2`
3. MišD na projektu/Configure/Convert to Maven Project. Podaci su sljedeći:  
Group Id: `org.foi.nwtis.{LDAP_korisnik}`  
Artifact Id: `{LDAP_korisničko_ime}-zadaca_3_wa_2`  
Version: `1.0.0`  
Packaging: `war`  
Name: `{LDAP_korisničko_ime}-zadaca_3_wa_2`
4. Preuzeti datoteku `zadaca_3_2.dio.zip` i u `pom.xml` dodati sadržaj datoteke `pom_Cargo_z_Glassfish_EE_DZ3_2.txt` koja je priložena uz zadaću
5. Dodati dependency za group id: `jakarta.platform` artifact: `jakarta.jakartaee-ee-api` i version: `9.1.0`
6. Dodati dependency za group id: `org.glassfish.web` artifact: `jakarta.servlet.jsp.jstl` i version: `2.0.0`
7. Dodati dependency za group id: `com.google.code.gson` artifact: `gson` i version `2.9.0`
8. Dodati dependency za group id: `org.projectlombok` artifact: `lombok` i version `1.18.22`
9. Dodati dependency za group id: `org.foi.nwtis` artifact: `NWTiS_REST_lib` i version: `2.3.0`
10. Dodati dependency za projekte `{LDAP_korisnik}-zadaca_3_lib_03_1` i `{LDAP_korisnik}-zadaca_3_lib_06_1`
11. Kreirati pakete
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.zrna`
  - `org.foi.nwtis.{LDAP_korisnik}.zadaca_3.slusaci`
12. Kreirati klasu za slušača aplikacije, dodati učitavanje konfiguracijskih podataka kod kreiranja aplikacije itd. Može se kopirati iz prethodne vježbe 9. / zadace 2.
13. Upisati u `web.xml` inicijalni parametar konteksta konfiguracija i dodati jednu od datoteka `NWTiS.db.config_1.xml` ili `NWTiS.db.config_2.xml`
14. U `pom.xml` treba podesiti pakete kod generiranja klasa. Potražiti `<!-- 2. dio početak -->` i za postaviti vlastitu aplikaciju i paket tj. korisničko ime umjesto `{LDAP_korisnik}`

```
<wsdlDirectory>${basedir}/../{LDAP_korisnik}-zadaca_3_wa_1/target/generated-sources/wsdl</wsdlDirectory>
```

```
<wsdlLocation>http://localhost:9090/{LDAP_korisnik}-zadaca_3_wa_1/aerodromi?wsdl</wsdlLocation>
```

```
<packageName>org.foi.nwtis.{LDAP_korisnik}.ws.aerodromi</packageName>
```

```
<wsdlDirectory>${basedir}/../{LDAP_korisnik}-zadaca_3_wa_1/target/generated-sources/wsdl</wsdlDirectory>
```

```
<wsdlLocation>http://localhost:9090/{LDAP_korisnik}-zadaca_3_wa_1/meteo?wsdl</wsdlLocation>
```

```
<packageName>org.foi.nwtis.{LDAP_korisnik}.ws.meteo</packageName>
```

15. Pripremiti za korištenje/izvršavanje (Run As/Maven install)

16. Pogledati sadržaj direktorija target/generated-sources/jax-wsimport

17. Kreirati klijentsku klasu s ulogom zrna pod nazivom AerodromiWs i anotirati ju s

```
@RequestScoped  
@Named("AerodromiWs")
```

18. Dodati u klasu referencu za JAXWS „aerodromi“:

```
@WebServiceRef(wsdlLocation =  
    "http://localhost:9090/{LDAP_korisnik}-zadaca_3_wa_1/korisnici?wsdl")  
private Aerodromi service;
```

19. Dodati potrebne metode za pristup do podataka putem JAXRS „aerodromi“ pozivom varijable service. Prethodno se kreira objekt novi objekt Aerodromi za varijablu servise. Npr. za dajSveAerodrome():

```
public List<Aerodrom> dajSveAerodrome() {  
    service = new Aerodromi();  
  
    WsAerodromi wsAerodromi = service.getWsAerodromiPort();  
    List<Aerodrom> lAerodromi = wsAerodromi.dajSveAerodrome();  
  
    return lAerodromi;  
}
```

20. Dodati potrebne varijable i njihove gettere/settere za pristup do podataka iz pogleda.

21. Kreirati klijentsku klasu s ulogom zrna pod nazivom pod nazivom MeteoWs i anotirati ju s

```
@RequestScoped  
@Named("MeteoWs")
```

22. Dodati u klasu referencu za JAXWS „meteo“:

```
@WebServiceRef(wsdlLocation =  
    "http://localhost:9090/{LDAP_korisnik}-zadaca_3_wa_1/meteo?wsdl")  
private Meteo service;
```

23. Dodati potrebne metode za pristup do podataka putem JAXRS „meteo“ pozivom varijable service. Prethodno se kreira objekt novi objekt Meteo za varijablu servise.

24. Upisati u **web.xml** dodati nakon </context-param> za uključivanje FacesServlet

```
<servlet>  
    <servlet-name>NWTiS_FacesServlet</servlet-name>  
    <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>  
</servlet>  
<servlet-mapping>  
    <servlet-name>NWTiS_FacesServlet</servlet-name>  
    <url-pattern>*.xhtml</url-pattern>  
</servlet-mapping>  
  
<welcome-file-list>  
    <welcome-file>index.xhtml</welcome-file>  
</welcome-file-list>
```

25. Kreirati potrebne poglede index.xhtml, sviAerodromi.xhtml, rasporedAerodroma.xhtml, ... Kao predložak mogu se uzeti primjeri s predavanja za JSF.

26. Preuzeti datoteku zadaca\_3\_3.dio.zip i izvršiti komande iz datoteke komandeCertifikati.txt. Time će se dodati (importirati) certifikati za web mjesta OpenSkyNetwork, OpenWeatherMap i LocationIQ u spremište certifikata/certifikata na poslužitelju Glassfish EE. Standardna lozinka za spremište ključeva je: „changeit“.



27. Trajanje certifikata za web mjesta:

- OpenSky Network – 30.07.2022.
- OpenWeatherMap – 19.06.2022.
- LocationIQ – 18.06.2022.