
Vježba: 9 – Zadaća 2. Web sustav "Informacije o aerodromima, letovima i problemima"

1. Kreiranje direktorija roditeljskog projekta zadaće (direktorij **{korisnik}/{LDAP_korisničko_ime}-zadaca_2**). U nastavku se direktorij za vježbu simbolički označava kao **{zadaća}**.

```
mkdir  {korisnik}/{LDAP_korisnik}-zadaca_2
```

2. Postavljenje dozvola za rad na poslužitelju Payara.

```
cd /opt/payara5-web/glassfish/
sudo chmod -R go+rw domains/
```

Pokretanje programa **Eclipse IDE**.

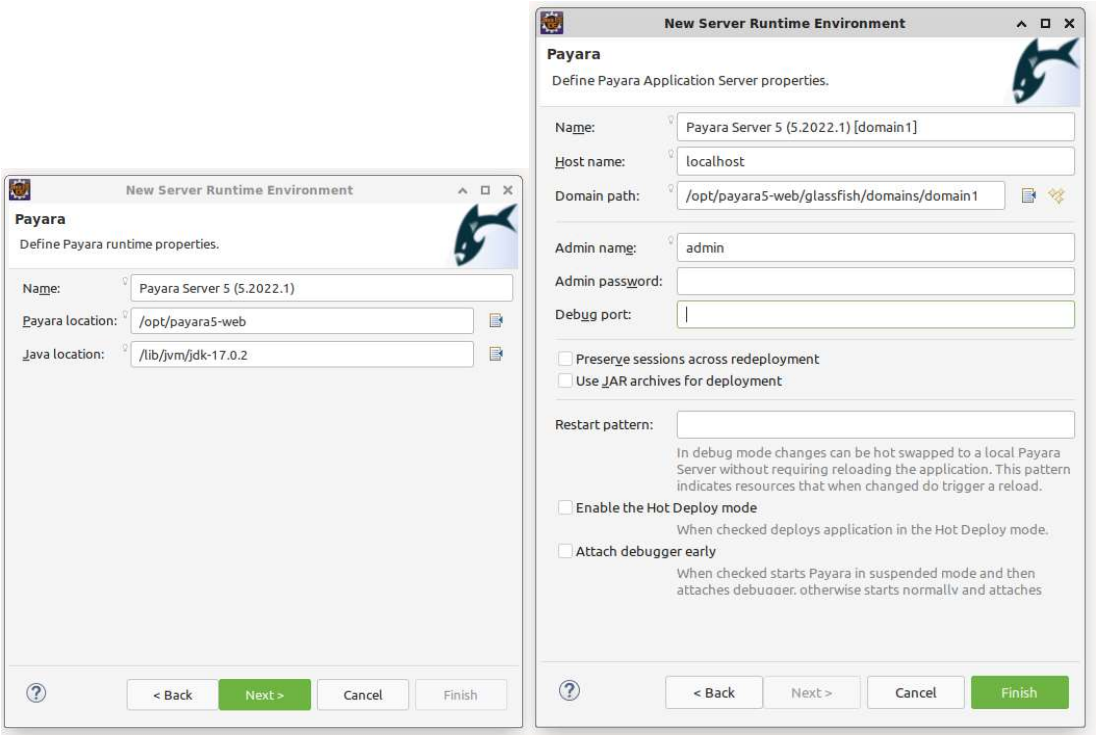
Instalirati dodatake za Eclipse IDE pod nazivom:

- Payara Tools

Help/Eclipse Marketplace/ Find: Web odabrati {naziv}. Nakon provedenog instaliranja dodatka potrebno je ponovno pokrenuti Eclipse IDE.

Priprema Payara Web poslužitelja za rad na vježbama:

3. Počinje se s dodavanje Window/Preferences/Server/Runtime Environment/Add . Odabrati Payara/Payara i za Payara location: se upisuje /opt/payara5-web, a za Java location: /lib/jvm/jdk-17.0.2. U inicijalnoj varijanti administratorski korisnički račun ima korisničko ime admin i nema lozinku.



4. Pokrenuti poslužitelj Payara Web Server (Servers/Payara Server */Start ili Debug). Ako je sve u redu na konzolu Payara poslužitelja dobije je ispis:

   Grizzly 2.4.4 started in: 2269ms - bound to [http-listener-1:8080, http-listener-2:8181, admin-listener:4848]

   Podaci pokazuju da se poslužitelj javlja na portu 8080 (HTTP), njegova sigurna veza je na portu 8181 (HTTPS), a administratorska konzola je na portu 4848.

5. Može se pogledati administratorska konzola na adresi http://localhost:4848/

Vježba_09_1: NWTiS_REST_demo

1. Preuzeti NWTiS_REST_demo i raspakiranje na direktorij {korisnik}
2. Otvoriti projekt (New/Open Projects from File System…)
3. Pripremiti za korištenje/izvršavanje (Run As/Maven install)
4. Otvoriti terminal i postaviti se na direktorij projekta i izvršiti projekt uz vlastite korisničke podatke za OpenSky Network (korisničko ime umjesto xxxx, a lozinka umjesto yyyy). Podaci se preuzimaju za aerodrom u Beču za razdoblje od 01.04.2022 00:00:00 do 02.04.2022 00:00:00 (lokalno vrijeme). Za pretvaranje vremenskih podataka iz jednog u drugi format (dd.mm.gggg u broj sekundi ili obratno) može se koristiti https://www.epochconverter.com/

```
cd  {korisnik}/NWTiS_REST_demo

java -jar target/NWTiS_REST_demo-2.3.0.jar xxxx yyyy LOWW 1648764000 1648850400
```

Vježba_09_2: Korijenski projekt

1. Create a sample project (skip archetype selection) – označiti
   Location: **{zadaća}**
   Group Id: **org.foi.nwtis.{LDAP_korisnik}**
   Artifact Id: **{LDAP_korisnik}-zadaca_2**
   Version: **1.0.0**
   Packaging: **pom**
   Name: **{LDAP_korisnik}-zadaca_2**

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_2_lib_03_1</module>
        <module>{LDAP_korisnik}-zadaca_2_lib_06_1</module>
        <module>{LDAP_korisnik}-zadaca_2_wa_1</module>
        <module>{LDAP_korisnik}-zadaca_2_wa_2</module>
-->
    </modules>
```

3. Kopirati {LDAP_korisnik}-vjezba_03_1 i podesiti tako da postane {LDAP_korisnik}-zadaca_2_lib_03_1. Postupak je upisan kod vježbe 4/zadaća 1. Isto Kopirati {LDAP_korisnik}-vjezba_06_1 i podesiti tako da postane {LDAP_korisnik}-zadaca_2_lib_06_1. Srediti da postanu aktivni moduli korijenskog projekta zadaće.
4. Korijenski projekt zadaće pripremiti za korištenje/izvršavanje (Run As/Maven install)

Vježba_09_4: Web aplikacija br. 1

1. Kreiranje korijenskog direktorija projekta **{LDAP_korisničko_ime}-zadaca_2_wa_1**

   ```
   mkdir  {zadaća}/{LDAP_korisnik}-zadaca_2_wa_1
   ```

2. New/Other…/Dynamic Web Project. Naziv projekta **{LDAP_korisničko_ime}-zadaca_2_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_2_wa_1**
   Version: **1.0.0**
   Packaging: **war**
   Name: **{LDAP_korisničko_ime}-zadaca_2_wa_1**
4. Preuzeti datoteku zadaca_2_1.dio.zip i u pom.xml dodati sadržaj datoteke pom_Cargo_za_Payara_Glassfish.txt koja je priložena uz zadaću
5. U pom.xml potražiti dio za uključivanje NWTiS repozitorija

   ```
   <repositories>
          <repository>
                 <id>NWTiS</id>
                 <name>NWTiS</name>
                 <url>http://nwtis.foi.hr:8088/repository/NWTiS_2022/</url>
          </repository>
   </repositories>
   ```

6. Dodati dependency za group id: **jakarta.platform** arthefact: **jakarta.jakartaee-web-api** i version: 9.1.0
7. Dodati dependency za group id: **org.glassfish.web** arthefact: **jakarta.servlet.jsp.jstl** i version: 2.0.0
8. Dodati dependency za group id: **org.projectlombok** artifact: **lombok** i version 1.18.22
9. Dodati dependency za group id: **org.hsqldb** artifact: **hsqldb** i version: 2.6.1
10. Dodati dependency za group id: **mysql** artifact: **mysql-connector-java** i version: 8.0.28
11. Dodati dependency za group id: **com.google.code.gson** artifact: **gson** i version 2.9.0
12. Dodati dependency za group id: **org.foi.nwtis** artifact: **NWTiS_REST_lib** i version: 2.3.0
13. Dodati dependency za projekte {LDAP_korisnik}-zadaca_2_lib_03_1 i {LDAP_korisnik}-zadaca_2_lib_06_1
14. Kreirati pakete
    - **org.foi.nwtis.{LDAP_korisnik}.zadaca_2.dretve**
    - **org.foi.nwtis.{LDAP_korisnik}.zadaca_2.podaci**
    - **org.foi.nwtis.{LDAP_korisnik}.zadaca_2.rest**
    - **org.foi.nwtis.{LDAP_korisnik}.zadaca_2.slusaci**
15. Kreirati klasu PreuzimanjeRasporedaAerodroma kao dretvu. U metodi start() preuzeti iz postavki potrebne podatke i pridružiti ih varijablama. Kreira se objekt klase OSKlijent. U run() metodi postaviti petlju koja se izvodi dok je ispunjen uvjet. U petlji se dohvaćaju aerodromi za koje se preuzimaju podaci. Za svaki aerodrom se pozivaju metode getDepartures(…) i getArrivals(…) na objektu klase OSKlijent. Može se iskoristiti kod iz primjena NWTiS_REST_demo. Na početku se testira radi li preuzimanje podataka od OpenSky Network servisa. Na više mjesta (početka funkcije start(), početak funkcije run() i kod petlje za aerodrome) postaviti kontrolne točke (Beakpoint)
16. Pripremiti za korištenje/izvršavanje (Run As/Maven install)

17. Preporučuje se da je poslužitelj Payara Web Server u debug modu
18. Pripremiti za isporuku (Run As/Maven build…). Kod Goals: staviti cargo:redeploy, kod Profiles: ServerEE-local. Prepoporučuje se dodati kod naziva – redeploy
19. Analizirati rad web aplikacije
20. PreuzimanjeRasporedaAerodroma: Tek kada su postavljeni svi potrebni elementi vezani uz rad dretve, potrebne pauze u radu, uvjeta za kraj rada dretve i sl. može se krenuti u realizaciju upisa podataka.
21. Kreirati klasu za slušača aplikacije, dodati učitavanje konfiguracijskih podataka kod kreiranja aplikacije itd.
22. Preuzeti datoteku zadaca_2_2.dio.zip i izvršiti skripte za HSQLDB i MySQL.
23. Kreirati DAO klase za tablice AERODROMI_PRACENI, AERODROMI_POLASCI, AERODROMI_DOLASCI, AERODROMI_PROBLEMI
24. PreuzimanjeRasporedaAerodroma: U metodi run() dobiveni podaci upisuju se u svoje tablice u bazi podataka pomoću objekata svojih DAO klasa. Ako se javlja problem kod preuzimanja podataka za pojedini aerodrom onda se upisuje u tablicu za probleme u bazi podataka.
25. Kreirati klasu RestServis da nasljeđuje klasu Application. Dodati anotaciju @ApplicationPath("api").
26. Kreirati klasu RestAerodromi. Dodati anotaciju @Path("aerodromi"). Kreirati metode:
    - public Response dajSveAerodrome()
    - public Response dajAerodromeZaPratiti()
    - public Response dodajAerodromZaPratiti(String icao)
    - public Response dajAerodrom(String icao)
    - public Response dajPolaskeAerodoma(String icao)
    - public Response dajDolaskeAerodoma(String icao)
27. Sve metode treba anotirati potrebnom metodom i formatom u kojem se vraćaju podaci
28. Pojedine metode treba anotirati putanjom i vrstom parametara kojom prima podatke
29. Pripremiti za korištenje/izvršavanje (Run As/Maven install)
30. Pripremiti za isporuku (Run As/Maven build…). Kod Goals: staviti cargo:redeploy, kod Profiles: ServerEE-local. Prepoporučuje se dodati kod naziva – redeploy
31. Otvoriti u pregledniku http://localhost:8080/{LDAP_korisničko_ime}-zadaca_2_wa_1/api/aerodromi
32. Kreirati klasu RestProblemi. Dodati anotaciju @Path("problemi"). Kreirati metode:
    - public Response dajSveProbleme()
    - public Response dajProbleme(String icao)
    - public Response obrisiProbleme (String icao)
33. Sve metode treba anotirati potrebnom metodom i formatom u kojem se vraćaju podaci
34. Pojedine metode treba anotirati putanjom i vrstom parametara kojom prima podatke
35. Pripremiti za korištenje/izvršavanje (Run As/Maven install)
36. Pripremiti za isporuku (Run As/Maven build…). Kod Goals: staviti cargo:redeploy, kod Profiles: ServerEE-local. Prepoporučuje se dodati kod naziva – redeploy
37. Otvoriti u pregledniku http://localhost:8080/{LDAP_korisničko_ime}-zadaca_2_wa_1/api/problemi

38. Preuzeti datoteku zadaca_2_3.dio.zip i ažurirati konfiguracijske datoteke prema uputama u dodatniKonfigPodaci.txt. Pripremljene su curl komande u datoteci komande.txt za različite metode servisa. Otvoriti terminal i izvršiti pojedinu komandu kako se završava pojedina metoda servisa.

Vježba_09_4: Web aplikacija br. 2

1. Kreiranje korijenskog direktorija projekta **{LDAP_korisničko_ime}-zadaca_2_wa_2**

   ```
   mkdir  {zadaća}/{LDAP_korisnik}-zadaca_2_wa_2
   ```

2. New/Other…/Dynamic Web Project. Naziv projekta **{LDAP_korisničko_ime}-zadaca_2_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_2_wa_2**
   Version: **1.0.0**
   Packaging: **war**
   Name: **{LDAP_korisničko_ime}-zadaca_2_wa_2**
4. U pom.xml dodati sadržaj datoteke „pom Cargo za Payara i Glassfish“ koja je priložena uz zadaću
5. Dodati dependency za group id: **jakarta.platform** arthefact: **jakarta.jakartaee-web-api** i version: 9.1.0
6. Dodati dependency za group id: **org.glassfish.web** arthefact: **jakarta.servlet.jsp.jstl** i version: 2.0.0
7. Dodati dependency za group id: **jakarta.mvc** arthefact: **jakarta.mvc-api** i version: 2.0.0
8. Dodati dependency za group id: **com.google.code.gson** artifact: **gson** i version 2.9.0
9. Dodati dependency za group id: **org.eclipse.krazo** arthefact: **krazo-core** i version: 2.0.0
10. Dodati dependency za group id: **org.eclipse.krazo** arthefact: **krazo-jersey** i version: 2.0.0
11. Dodati dependency za group id: **org.projectlombok** artifact: **lombok** i version 1.18.22
12. Dodati dependency za group id: **org.foi.nwtis** artifact: **NWTiS_REST_lib** i version: 2.3.0
13. Dodati dependency za projekte {LDAP_korisnik}-zadaca_2_lib_03_1 i {LDAP_korisnik}-zadaca_2_lib_06_1
14. Kreirati pakete
    - **org.foi.nwtis.{LDAP_korisnik}.zadaca_2.mvc**
    - **org.foi.nwtis.{LDAP_korisnik}.zadaca_2.slusaci**
15. Kreirati klasu za slušača aplikacije, dodati učitavanje konfiguracijskih podataka kod kreiranja aplikacije itd.
16. Kreirati klasu AerodromiKlijent.
17. Preuzeti datoteku zadaca_2_4.dio.zip i dodati metodu za sve aerodrome. Na temelju te metode rade se ostale metode.
18. Kreirati klasu MVC koja nasljeđuje Application i anotirati ju s @ApplicationPath("mvc")
19. Kreirati direktorij views unutar WEB-INF
20. Kreirati pogled index.jsp s poveznicama za ostale poglede. Voditi brigu o putanjama. Preporuka da se stavi naziv konteksta (kao u vježbi 08_1)
21. Kreirati pogled sviAerodromi.jsp. Za početak koristiti <c:forEach …> za ispis aerodroma.
22. Kreirati klasu PregledAerodroma. Anotirati ju

```
@Controller
@Path("aerodromi")
@RequestScoped
```

23. U klasu dodati

```
@Inject private Models model;
```

24. Za svaki pogled koji sami prikazuje podatke dodati metodu koja se anotira npr:

```
@GET
@Path("pocetak")
@View("index.jsp")
public void pocetak() {
}
```

25. U svakoj metodi koja se povezana na pogled koji dinamički prikazuje podatke potrebno je pripremljene podatke dodati u model putem metode model.put(ključ, podaci);