

## Početak

U okviru **Eclipse IDE for Enterprise Java and Web Developers 2023-09** razvojnog okruženja, prvenstveno treba odabrati *Workspace* u kom ćete raditi. Zatim, ukoliko nije otvorena podrazumevana Java EE perspektiva, potrebno ju je odabrati klikom na *Window -> Perspective -> Open Perspective -> Other -> Java EE*.

## Kreiranje konekcije

U okviru *Data Source Explorer* taba je potrebno napraviti konekciju sa bazom desnim klikom na folder *Database Connections -> New*. U okviru prozora je od ponuđenih *Connection Profile Types* potrebno izabrati **MySQL** i promeniti ime konekcije (opciono). Klikom na dugme *Next>* prelazi se na sledeći prozor u kom je potrebno izabrati Driver, koji je prethodno neophodno definisati klikom na malo dugme sa plusićem pored padajućeg menija koje označava opciju *New Driver Definition*. Klikom na navedenu opciju, otvara se novi prozor u okviru kog je potrebno odabrati MySQL JDBC Driver verziju 5.1 i opciono promeniti ime koje je automatski generisano odabirom (kao i u slučaju naziva konekcije). Zatim, potrebno je preći u tab *JAR List*, označiti ponuđeni .jar fajl u okviru *Driver files* i zatim kliknuti na dugme *Edit JAR/Zip* čime će se otvoriti fajl sistem računara u kom je potrebno odabrati prethodno preuzeti **mysql-connector-java-8.0.30.jar** fajl, kliknuti *Open* i zatim *OK*. Ukoliko je definicija ovog drivera nekada ranije već napravljena nije potrebno praviti iznova nove, iako potencijalno kreirate konekciju više puta. Na kraju, ostalo je samo da popunimo parametre za konekciju na bazu sledećim vrednostima:

Database:	database (opciono promeniti, nije obavezno)
URL:	jdbc:mysql://nastava.is.pmf.uns.ac.rs:3306/ris
Username:	ris
Password:	ris2Nastava!

Dodatno, označite kvadratić *Save Password* kako ne bi svaki put kad želite da se povežete na fakultetsku bazu iz Eclipse-a morali da unosite lozinku za pristup. Opciono, kliknite na dugme *Test Connection* kako bi pre klika na dugme *Finish* bili uvereni da ste ispravno uneli sve parametre za povezivanje na bazu. Klikom na dugme *Finish*, u okviru *Data Source Explorer* se prikazuje kreirana instance konekcije.

**Napomena:** Da biste mogli da se povežete na fakultetsku bazu, podrazumeva se da Vam je aktivan VPN, koji prethodno treba da konfigurišete -> <https://www.pmf.uns.ac.rs/servisi/korisnicka-uputstva/>

## Kreiranje JPA projekta

Klikom na *File -> New -> JPA Project*, otvara se prozor u kom je potrebno samo zadati ime projekta i klikom na prvo i sledeće dugme *Next>*, na poslednjem prozoru proveriti da li je u okviru *JPA Implementations* za *Type* odabrano *Disable Library Configuration*, ukoliko to nije slučaj, promeniti na navedeno. Nakon što se klikne *Finish*, prikazuje se u okviru *Project Explorer* taba kreiran JPA projekat.

Kako bi mogli nastaviti dalje sa radom, potrebno je kreirani projekat konvertovati u Maven projekat, prilikom čega će biti generisan neophodan *pom.xml* fajl za dodavanje zavisnosti. Desnim klikom na JPA Projekat, pa zatim *Configure-> Convert to Maven Project* i klikom na dugme *Finish*, otvara nam se generisani *pom.xml* fajl u okviru kog je neophodno navesti potrebne zavisnosti.

Nakon zatvorenog taga `</build>`, navesti sledeći skup zavisnosti:

```
<dependencies>
  <dependency>
    <groupId>jakarta.persistence</groupId>
    <artifactId>jakarta.persistence-api</artifactId>
    <version>3.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.0.0.Final</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.30</version>
  </dependency>
</dependencies>
```

Zatim, potrebno je i instalirati ove gorenavedene zavisnosti u projekat tako što ćemo ići desni klik na projekat *Run As -> Maven install*. Indikator da je uspešno realizovana instalacija jeste poruka u konzoli **BUILD SUCCESS**.

Ostalo je još da u okviru projekta klikom na *JPA Content* otvorimo *persistence.xml* fajl. Odlaskom u *Connection* tab, za *Transaction Type* biramo *Resource Local*, i zatim nam je omogućeno da kliknemo na plavi link *Populate from connection* i odaberemo kreiranu konekciju na bazu. Da bismo sačuvali promene, dovoljno je samo iskoristiti prečicu **Ctrl + s**.

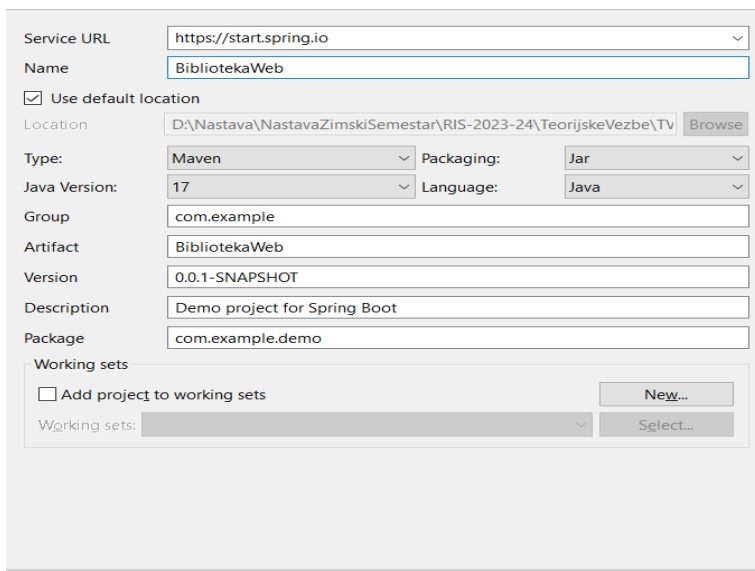
Sada je potrebno generisati entitete na osnovu tabela u bazi podataka. Desnim klikom na projekat, *New -> JPA Entities from Tables*, odabirom kreirane konekcije i šeme *ris* iz padajućeg menija, pojavljuju se sve tabele koje imamo u bazi. Ukoliko radimo sa modelom Biblioteka, potrebno je štiklirati adekvatne tabele, u suportonom tabele iz modela Pozorište. Nakon što označimo željene tabele, klikom na dugme *Next->* prelazimo na prozor u kom vidimo koje sve tipove veza imamo između prethodno označenih tabela. Zatim, kliknemo na *Next>* i dolazimo do momenta u kom treba da izaberemo na koji način se kreiraju vrednosti primarnih ključeva tabela. Za to je u okviru *Key generator* potrebno odabrati *identity*. Zatim do kraja idemo na *Next>* i kliknemo na *Finish*.

U okviru JPA projekta sada imamo paket *model* u kom se nalaze generisani entiteti. Ono što je još potrebno promeniti u svakom od tih entiteta (Java klasa) jeste biblioteka koja se koristi od JPA verzije 3.0 (koje nažalost iz nekog razloga u novom Eclipsu još nema, a Spring Boot 3.1.5 upravo radi sa tom verzijom). Dakle, na početku svake klase gde se nalaze importi, neophodno je promeniti import **javax.persistence.\*** na **jakarta.persistence.\***. Za to koristimo opciju *Search -> Search*, potrebno je biti u *File Search* tabu, ukucamo u *Containing text* **javax**, kliknemo na dugme *Replace* i u okviru *With* navedemo **jakarta**. Za kraj, kako bi promena biblioteke bila registrovana globalno na nivou čitavog projekta, ići desni klik na *persistence.xml* fajl i opciju *Synchronize Class List* i zatim desni klik na *JPA projekat-> Maven -> Update project* i **OK**.

## **Kreiranje Spring Boot projekta**

Kako bismo mogli raditi bilo šta sa Spring-om, potrebno je instalirati jedan plugin. Klikom na *Help -> Eclipse Marketplace* otvara se prozor je u okviru kog u *Search* polju potrebno ukucati *spring*. Ono što treba instalirati je **Spring Tools 4 (aka Spring Tool Suite 4) 4.20.0 RELEASE** (potrajaće neko vreme). Kada instalacija uspešno prođe, možemo preći na kreiranje Spring Boot projekta.

Klikom na opciju *File->New->Other->Spring Starter Project* otvara nam se prozor koji, pre prelaska na sledeći prikaz, treba da izgleda ovako (naziv projekta staviti po želi):



Zatim, kada se pređe na sledeći prozor, za verziju Spring Boot-a odabrati **3.1.5** i označiti sledeće zavisnosti koje će se automatski dodati u *pom.xml* fajla projekta:

- JDBC API, MySQL Driver, Spring Data JPA, Spring Web, Spring Boot DevTools i Validation

Klikom na dugme Finish, dobijamo kreiran *Spring Boot* projekat. Još je potrebno u okviru *pom.xml* fajla Spring Boot projekta dodati još tri zavisnosti kako bismo mogli raditi sa JSP stranicama i koristiti potrebne taglib direktive. Dodati sledeće:

```
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
<dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
</dependency>
<dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
</dependency>
```

Za kraj rada sa *pom.xml* fajlom, potrebno je navesti još jednu zavisnost koja će povezati prethodno kreirani JPA projekat sa Spring Boot projektom kako bi generisane entitete mogli koristiti u Spring Boot projektu. Potrebno je dodati sledeću zavisnost u okviru taga *<dependencies>* (obratiti pažnju da je BibliotekaJPA naziv JPA projekta, u skladu sa nazivom Vašeg JPA projekta, **potrebno** promeniti):

```
<dependency>
    <groupId>BibliotekaJPA</groupId>
    <artifactId>BibliotekaJPA</artifactId>
    <version>0.0.1-SNAPSHOT</version>
</dependency>
```

U Spring Boot projektu u folderu **src/main/java** u paketu **com.example.demo** se nalazi **BibliotekaWebApplication.java** fajl u okviru kog je potrebno iznad zaglavlja klase, a posle anotacije *@SpringBootApplication* dodati anotaciju *@EntityScan("model")*.

Još je potrebno u okviru **src/main/resources** foldera u fajlu **application.properties** dodati sledeće:

```
spring.mvc.view.prefix=/
spring.mvc.view.suffix=.jsp
spring.datasource.url=jdbc:mysql://nastava.is.pmf.uns.ac.rs:3306/ris
spring.datasource.username=ris
spring.datasource.password=ris2Nastava!
server.servlet.context-path=/Biblioteka
server.port=8080
spring.jpa.hibernate.naming.physical-
strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```

Za kraj je još samo potreban folder u kom će biti smeštene HTML i JSP stranice, a njega treba kreirati u **src/main** folderu (pored već postojećih *java* i *resource* foldera). Folder nazvati **webapp**.

Radi sigurnosti da je sve preuzeto i uvezano kako treba, preporuka je ići desni klik na Spring Boot projekat -> *Maven* -> *Update Project*, označiti i JPA i Spring Boot projekat i kliknuti na *OK*.

Pokretanje projekta: Desni klik na *Spring Boot projekat* -> *Run As* -> *Spring Boot App*.

U okviru pretraživača, navodimo putanju do *index.jsp* stranice aplikacije:

**localhost:8080/putanja koju ste naveli u server.servlet.context-path**