

Java Persistence API Övningar

Översikt

I denna övning kommer vi att:

- Lägga till och konfigurera persistence.xml.
- Konfigurera EntityManager i huvudmetoden.
- Skapa objekt och lägga till dem i databasen.
- Utföra några SQL-frågor.
- Ta bort data från databasen.

Använda JPA i projektet

Denna övning bygger på en **One-To-Many-relation mellan Tutor och Student**. Om du inte har gjort One-To-Many övningarna tidigare, gör dem först.

Alternativt kan du ladda ner det senaste projektet vi har skapat [här](#).

Nu ska vi använda JPA istället för klassisk Hibernate, vilket innebär att vi måste göra vissa ändringar.

Steg 1: Lägga till och konfigurera persistence.xml

- Skapa en ny mapp under src/main/resources och döp den till META-INF.
- Lägg till filen [persistence.xml](#) i META-INF.
- Kopiera databasdrivrutinen från hibernate.cfg.xml och uppdatera persistence.xml genom att ersätta TODO med rätt värden:
 - Ange hibernate.connection.driver_class.
 - Ange hibernate.connection.url.
- Ställ in följande egenskaper:
 - Inaktivera SQL-utskrift.
 - Använd create som hbm2ddl.auto-inställning.
- Ta bort hibernate.cfg.xml från projektet eftersom den inte längre behövs.

Steg 2: Konfigurera JPA i huvudmetoden

- Ta bort all kod relaterad till SessionFactory, Session och Transaction.
- Skapa och konfigurera en EntityManager och EntityTransaction.
- Säkerställ att transaktionen startas och avslutas korrekt.
- Importera nödvändiga bibliotek:
 - jakarta.persistence.EntityManager
 - jakarta.persistence.EntityManagerFactory
 - jakarta.persistence.EntityTransaction
 - jakarta.persistence.Persistence
- Ta bort eventuella gamla helper-metoder som skapar en SessionFactory.

Steg 3: Skapa och lagra objekt i databasen

- Skapa en Tutor-instans och lagra den i databasen.
- Skapa tre Student-objekt och koppla dem till tutorn.
- Lagra studenterna i databasen.
- Hämta alla studenter i tutors teachingGroup och skriv ut antalet.
- Kör programmet och kontrollera att datan lagras korrekt i databasen.

Steg 4: Uppdatera persistence.xml

- Ändra hbm2ddl.auto från create till update.

Förstå skillnaden mellan create och update:

- create tar bort och skapar om tabeller varje gång koden körs.
- update behåller data och uppdaterar endast befintliga tabeller.

Steg 5: Utföra SQL-frågor

- Kommentera bort kod som skapar Tutor och Student-objekt.
- Hämta en tutor från databasen genom att använda find-metoden.
- Hämta och skriv ut tutors namn.
- Hämta alla studenter kopplade till tutorn och skriv ut deras namn.

- Hämta en specifik student med ett givet ID och skriv ut informationen.
- Kör programmet och verifiera att datan hämtas korrekt.

Steg 6: Ta bort data

- Hämta en student med ett specifikt ID.
- Ta bort studenten från databasen.
- Kör programmet och verifiera att studenten har tagits bort genom att kontrollera databasen.