

# Spring JDBC Template Övningar

## Mål

I denna övning kommer vi att:

- Implementera ren JDBC för att hantera databasen i BookStore-projektet.
- Använda Spring JDBC Template för att förenkla JDBC-koden.
- Skapa och konfigurera en databasanslutning med HSQLDB.

## Steg 1: Implementera ren JDBC i projektet

- Skapa ett nytt paket *se.yrgo.spring.data* och lägg till två filer:
  - [BookDao.java](#) (ett interface som definierar databasoperationer).
  - [BookDaoJdbcImpl.java](#) (en implementation av interfacet som använder ren JDBC).
- Säkerställ att Book-klassen innehåller följande metoder:
  - getTitle()
  - getAuthor()
  - getPrice()
- Lägg till följande beroende i din pom.xml: (Vi kommer att använda *HSQLDB* som databas)

```
<dependency>
  <groupId>hsqldb</groupId>
  <artifactId>hsqldb</artifactId>
  <version>1.8.0.10</version>
</dependency>
```

- I klientklassen:
  - Skapa en ClassPathXmlApplicationContext för att ladda application.xml.
  - Hämta BookService från containern och registrera en ny bok.
  - Hämta hela katalogen och skriv ut böckerna.

## Steg 2: Ersätt mock-implementationen med en riktig databas

- Skapa en ny klass *BookServiceProductionImpl* som implementerar *BookService*.
- Implementera alla metoder i *BookService* genom att delegera anropen till *BookDao*.
- Ändra *application.xml*:
  - Uppdatera *bookService* så att den använder den nya implementeringen.
  - Injicera en DAO i *bookService* med constructor injection.
  - Kör koden och verifiera att böckerna sparas och hämtas från databasen.

## Steg 3: Använda Spring JDBC Template

- Skapa en ny klass *BookDaoSpringJdbcImpl* som implementerar *BookDao*.
- Skapa en privat instansvariabel av *JdbcTemplate* och injicera den via konstruktorn.
  - Lägg till följande beroende i *pom.xml*:

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>5.3.14</version>
</dependency>
```

- Implementera metoden *create(Book newBook)* genom att använda *update()*-metoden i *JdbcTemplate*.
- Deklarera och använd en SQL-variabel för att förenkla hanteringen av SQL-frågor.

## Steg 4: Implementera databasoperationer med Spring JDBC Template

- Implementera metoden *allBooks()* genom att använda *query()* i *JdbcTemplate*.
- Skapa en intern klass *BookMapper* som implementerar *RowMapper<Book>* och mappar *ResultSet* till *Book*-objekt.
- Implementera metoden *findBooksByAuthor(String author)* genom att använda *query()* med en parameter.
- Implementera metoden *findByIsbn(String isbn)* genom att använda *queryForObject()*.

- Implementera metoden *delete(Book redundantBook)* genom att använda `update()` för att ta bort en bok.

Nästa steg blir att konfigurera databasanslutningarna.