

# Spring Injection och Wiring - Övningar

## Mål

I denna övning kommer vi att:

- Använda dependency injection och Spring Container.
- Skapa ett mer realistiskt projekt.
- Implementera en mock (prototyp) av en service.
- Lära oss hur vi kan växla mellan olika implementationer utan att ändra klientkoden.

## Projekt: BookStore

Ladda ner projektet [BookStore](#) och lägg till det i din IDE.

Granska klasserna Book och BookService för att se vilka egenskaper och metoder som finns.

## Steg 1: Implementera en mock av BookService

- Eftersom vi kanske inte har en fullt fungerande databas ännu, skapar vi en mock-implementation för att simulera backend:
  - Titta på klassen *BookServiceMockImpl* som implementerar *BookService*.
  - Implementera metoderna så att de hämtar och hanterar böcker från den lokala datastrukturen.
  - För metoder som inte är klara, låt dem returnera null eller en tom lista.

## Varför mock-implementation?

- Vi kan testa applikationen utan att ha en databas.
- Vi kan enkelt byta till en riktig databas senare genom att bara ändra konfigurationen.

## Steg 2: Lägg till en bean i application.xml

- Skapa en Spring-konfigurationsfil application.xml.
- Lägg till en bean för BookServiceMockImpl.
- Definiera ett id (bookService) för beanen.

## Steg 3: Skapa klientklassen

- Skapa en klientklass som använder Spring Container för att hämta och anropa BookService.
- Använd ClassPathXmlApplicationContext för att ladda application.xml.
- Hämta BookService från containern.
- Anropa metoden för att hämta hela bokenkatalogen och skriv ut resultatet.

## Vad du ska testa

- Kontrollera att alla tre böckerna som lades till i mock-implementationen skrivs ut korrekt.
- Om böckerna visas i terminalen betyder det att dependency injection fungerar.