

Starting to use Spring Framework

Managing Beans and Dependency Injection

Spring Beans

- ▶ A Spring bean is a Java object that is managed by the Spring Container
- ▶ Spring is responsible for creating, configuring and managing its lifecycle
- ▶ When a bean is created it is typically stored in a bean container

Spring Beans - Dependency Injection

- ▶ One of the main benefits of using Spring Beans is the support for DI
- ▶ Spring can automatically wire up dependencies between objects with DI
- ▶ This allows more modular and reusable code since objects can be swapped out or reused in different contexts

Spring Beans - Configuration

- ▶ Spring Beans are typically defined in a configuration file or a Java configuration class
- ▶ This allows developers to configure and customize the properties and behavior of the bean

Spring Beans - Scoping

- ▶ Spring Beans can be scoped to control their lifecycle and visibility
- ▶ A bean can be defined as a singleton - only one instance is created and shared across the application
- ▶ A bean can be defined as a prototype - a new instance is created each time the bean is requested
- ▶ Other scopes, such as request and session can be used in web apps

Configuration classes

- ▶ Configuration classes are used to define and configure Spring Beans
- ▶ Can be used instead of XML configuration that was used in the past
- ▶ Are annotated with `@Configuration` which tells Spring their purpose

Configuration classes - features

- ▶ Bean Definition - each @Bean annotated method defines a Spring bean and its dependencies.
- ▶ Takes advantage of the dependency injection capabilities in Spring to wire up dependencies between beans
- ▶ Dependencies can be specified as constructor parameters or using @Autowired annotations

Demo 1 - Java without Spring

- ▶ Introducing a Java application without using Spring for dependencies

Demo 2 - Java with Spring

- ▶ Starting to use Spring with the Spring Application context
- ▶ Spring context dependency in pom.xml
- ▶ Spring annotations, like @Component
- ▶ Using the Application context for dependency to the Spring Bean

Exercise 1 - Java with Spring

- ▶ Use the starter code in the starter package of the demo project for this exercise
- ▶ Start using Spring to handle the BookRepository
- ▶ Steps:
 - ▶ Create a @Configuration class and configure component scan
 - ▶ Annotate the BookRepository to become a Spring bean
 - ▶ In the main method, create an ApplicationContext and use it to get a reference to the BookRepository Spring bean

Demo 3 - Spring with more beans

- ▶ Beans with dependency to another bean
- ▶ @Service annotation
- ▶ @Autowired and Constructor Dependency Injection

Exercise 2 - Create a @Service

- ▶ Create a BookService class that uses the BookRepository
- ▶ Inject the dependency to the BookRepository in BookService with @Autowired or constructor injection
- ▶ Use the BookService instead of the BookRepository from the main method

Demo 4 - Creating beans with @Bean

- ▶ Using @Bean methods in the configuration class to create Spring beans