

Spring

- Spring is an application development framework for enterprise Java
- Benefits of spring framework
 - POJO based development. Eliminates the need for EJB application server. Cost effective.
 - Spring makes use of existing frameworks. Jell well with the standard java stack.
 - Easily testable through dependency injection.
 - Lightweight.
- Dependency Injection
 - Let's look at these two words separately. Here the dependency part translates into an association between two classes. For example, class A is dependent on class B. Now, let's look at the second part, injection. All this means is that class B will get injected into class A by the IoC.
 - Dependency injection can happen in the way of passing parameters to the constructor or by post-construction using setter methods.
 - Types – Mandatory dependencies via constructor and optional dependencies via setter.
 - Constructor based
 - Setter based.
 - Primitive type, collection, another object shall be injected.
- Key components of Spring.
 - Core – Dependency Injection.
 - Bean – Implementation of factory pattern.
 - Expression Language – Manipulating objects graph at run time.
 - Spring MVC.
 - Struts integration
 - JDBC – Same transaction interface for local and global transactions.
 - ORM – Integration with existing frameworks like hibernate.
 - AOP – Aspect Oriented Programming
 - Test – Works with JUnit.
- Spring container is at the core of spring framework.
 - Creates objects.
 - Wire them together.
 - Configure them. – through xml or annotations or Java based.
 - Annotation injection is performed before XML injection, thus the latter configuration will override the former for properties wired through both approaches.
 - Manage their lifecycle.
 - Types of container
 - Bean Factory
 - Application context. – More powerful – Supports JNDI, EJB, Remoting, i18n, listeners, more than 1 config files, life cycle events.
 - Events specific to this context life also shall be handled.
- Bean

- Bean scope – Could be session, singleton, prototype.
- Singleton beans are not thread safe. Default scope of the bean is singleton.
- Lifecycle – Init, destroy.
- It is possible to hook in custom methods before & after initialization of a bean using Bean post processors.
- A bean configuration can inherit another bean configuration.
- Inner bean - A <bean/> element inside the <property/> or <constructor-arg/> elements defines a so-called inner bean. Inner beans are always anonymous with no id / name and they are always scoped as prototypes.
- AOP – Aspect Oriented Programming / Cross cutting concerns
 - Aspect – Cross cutting concern. (e.g. Logging)
 - PointCut – Method where we introduce advice
 - Advice – Code that needs to be run on point cut.
 - Before method execution
 - After method execution regardless of outcome.
 - After returning – After successful returning of the method.
 - After throwing – Exception
- Transaction management – 2 types supported.
 - Programmatic.
 - Declarative.

<https://spring.io/>

<https://spring.io/projects/spring-boot>