# 1. Introduce Spring

## 1.1 About this book

## 1.2 What is Spring?

-Spring: a lightweight framework for building Java applications.

+You can use Spring to build any application in Java

+lightweight isn’t related to number of classes or the size of distribution, it’s the principle of Spring philosophy: minimal impact: you have to make few changes to app to gain benefits

### 1.2.1 Evolution of Spring Framework

(See in book)

### 1.2.2 Spring Projects

spring.io/projects

(See in book)

## 1.3 Inverting Control or Injecting Dependencies

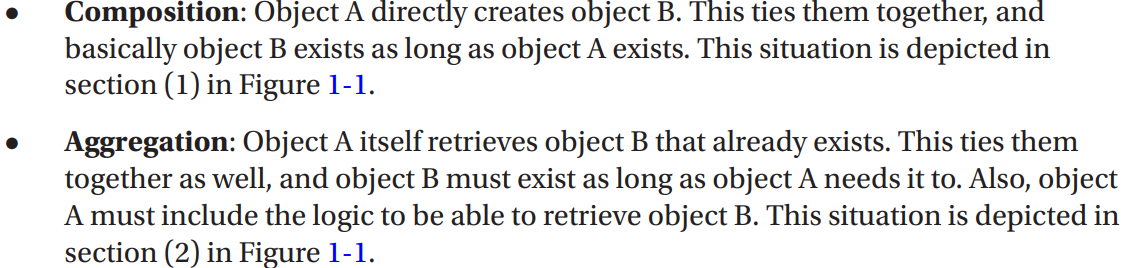
-The core of Spring Framework is based on the principle of **inversion of control (IoC)**: a technique that externalizes the creation and management of component dependencies. The action performed by any program is the result of interaction between its interdependent components.

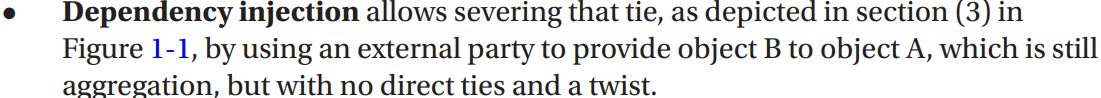
-**Dependency injection (DI)**: a concept that describes how dependent objects are connected at runtime by an external party.

-Object A needs an object of type B to perform its functions -> A depends on B. The 3 ways these 2 objects get to interact:

A screenshot of a computer

AI-generated content may be incorrect.





-**Inversion of control**: a design principle in which generic components are used to control the execution of problem-specific code, as in retrieving dependencies. Spring is a **dependency handler** used to perform DI, it was designed following IoC principle

-Spring’s DI implementation based on 2 core Java concepts: JavaBeans (POJOs-plan old Java objects) and interfaces.

-You can define dependency configuration in different ways: XML, annotation, Java configuration classes, Groovy-based configuration

-JavaBeans (POJOs) provides a standard mechanism for creating Java resources that are configurable in a number of was (constructors, setter methods). Any Spring-managed resource is bean.

-Interfaces and DI are technologies that are mutually beneficial. Designing and coding application to interfaces makes for a flexible application, but the complexity of wiring together is high. Using DI, you reduce the code you need to use an interface-based design in app.

+The use of interfaces allows dynamic proxies (Proxy pattern) to provide AOP.

-In context of DI, Spring acts like a container: provide instances of your application classes with all dependencies they need. Using Spring for DI follows JavaBeans naming conventions within your classes

### 1.3.1 Evolution of Dependency Injection

(See in books)

### 1.3.2 Beyond Dependency Injection

(See in books)

## 1.4 The Spring Community

# 2. Getting Started

## 2.1 Conventions

## 2.2 Who this book is for

## 2.3 What you need for this book

## 2.4 Prepare development environment

## 2.5 Understand Spring Packaging

-Spring packaging is modular: it allows to pick and choose which components you want to use in your application and to include only those components when you are distributing your app.

-Spring modules are JAR files that package the required code for that module.

-Spring Framework 6.0 comes with 22 modules. These name like: spring-<>-6.0.0.jar

A screenshot of a computer

AI-generated content may be incorrect.

A close up of a text

AI-generated content may be incorrect.

A close up of a text

AI-generated content may be incorrect.

A close up of a text

AI-generated content may be incorrect.

A close-up of a text

AI-generated content may be incorrect.

A close-up of a text

AI-generated content may be incorrect.

-If you sing Spring Boot, the appropriate set of Spring dependencies are configured depending on Spring Boot starter dependencies used. There are more than 30 of them

A close up of a text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## 2.6 Choose Modules for your application

-Without dependency management tool (Maven, Gradle), choosing which modules to use in application may be tricky.

### 2.6.1 Access Spring Modules on Maven Repository

### 2.6.2 Access Spring Modules Using Gradle

-Gradle is a powerful build tool use Groovy. Starting with version 4.x, Spring team has switched to using Gradle for configuration.

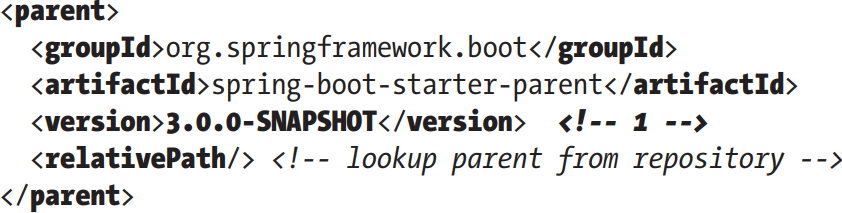
-The default name of Gradle configuration file for project is build.gradle.

A computer code with black text

AI-generated content may be incorrect.

### 2.6.3 Use Spring Boot Dependency Management

-You can use Spring Boot starter project as a parent project to provide your project with a minimal set of dependencies and default configuration.



-<dependencyManagement>

A screen shot of a computer program

AI-generated content may be incorrect.

-Gradle Spring Boot

A screen shot of a computer code

AI-generated content may be incorrect.

-In bigger projects, organized in multiple modules, configuration can create configuration templates that can be reused in modules.

## 2.7 Use Spring Documentation

docs.spring.io/spring-framework/docs/current/javadoc-api/

docs.spring.io/spring-framework/reference/

## 2.8 Putting a Spring into Hello World

### 2.8.1 Build sample

### 2.8.2 Refactor with Spring