Spring Introduction

Table of Contents

[Framework 4](#_Toc53177636)

[Why Framework? 4](#_Toc53177637)

[What is a Software Framework? 4](#_Toc53177638)

[Advantages of using a software framework 4](#_Toc53177639)

[Java Frameworks 4](#_Toc53177640)

[Spring 5](#_Toc53177641)

[What is Spring? 5](#_Toc53177642)

[Why to learn Spring Framework? 5](#_Toc53177643)

[Features of Spring Framework 5](#_Toc53177644)

[Lightweight 5](#_Toc53177645)

[Dependency Injection (DI) 5](#_Toc53177646)

[Inversion of Control (IOC) 6](#_Toc53177647)

[So, what Spring does? 6](#_Toc53177648)

[What makes Spring Framework special? 7](#_Toc53177649)

[Spring is Open Source 7](#_Toc53177650)

[Framework of Frameworks 7](#_Toc53177651)

[Spring has evolved over time 7](#_Toc53177652)

[Easy Development & Testing Cycles 7](#_Toc53177653)

[History of Spring Framework 8](#_Toc53177654)

[Spring Framework Architecture 8](#_Toc53177655)

[Spring Core Container 9](#_Toc53177656)

[Spring Data Integration and Data Access 9](#_Toc53177657)

[Spring Web 10](#_Toc53177658)

[Web Services and Rest API 10](#_Toc53177659)

[Aspect-Oriented Programming (AOP) 10](#_Toc53177660)

[Instrumentation 11](#_Toc53177661)

[Messaging 11](#_Toc53177662)

[Test 11](#_Toc53177663)

[What are Spring Framework, Spring JDBC, Spring MVC and Spring Boot? 11](#_Toc53177664)

[Spring Framework 11](#_Toc53177665)

[Spring JDBC 11](#_Toc53177666)

[Spring MVC 11](#_Toc53177667)

[Spring Boot 11](#_Toc53177668)

[Prerequisites 12](#_Toc53177669)

# Framework

## Why Framework?

Developing software is a complex process. It contains a lot of tasks such as – architecture, coding, designing, testing, deployment, maintenance, scaling etc. For only the coding part, programmers had to take a lot of care in syntax, declarations, garbage collection, statements, error handling, space complexity, time complexity, development cycles and more.

It could be really nice if some readymade library or system was there to simplify these tasks. This is where frameworks come into picture. These frameworks help developers to minimize these headaches.

## What is a Software Framework?

Software frameworks make life easier for developers by allowing them to take control of the entire software development process, or most of it, from a single platform.

## Advantages of using a software framework

1. Assists in establishing better programming practices and fitting use of design patterns
2. Code is more secure
3. Duplicate and redundant code can be avoided
4. Helps consistent development of code with fewer bugs
5. Makes it easier to work on sophisticated technologies
6. Several code segments and functionalities are pre-built and pre-tested. This makes applications more reliable
7. Testing and debugging the code is a lot easier and can be done even by developers who do not own the code
8. The time required to develop an application is reduced significantly
9. With the reduction in time and labor, softwares are cost efficient

## Java Frameworks

Java is one of the oldest languages with a huge community of developers. There are many frameworks of Java that are currently being used for various tasks, such as:

1. Spring
2. Apache Struts
3. Grails
4. Hibernate
5. JSF (Java Server Faces)
6. GWT (Google Web Toolkit)
7. Blade
8. Play
9. Vaadin
10. DropWizard
11. And many more…

# Spring

# What is Spring?

Spring is a **Java framework** used to develop Enterprise Applications.

Or, in other words:

Spring is a Framework used to develop Enterprise applications using the best practices of J2EE.

So, it is clear from above facts that:

1. Spring is a **Software Development framework**
2. Spring is a **Java based** software development framework
3. Spring is used to develop **Enterprise grade applications**

## More on Spring

Spring is **not just one framework** – it is a framework of frameworks.

What it means is:

1. It consists of 20+ modules – all targeted to solve some particular problem of an enterprise grade application
2. It can work well with other Java Frameworks as well with easy integration techniques.

## What is Spring?

* Spring is a Dependency Injection Framework used to make Java Applications loosely coupled
* Spring is a powerful, lightweight, open source application development framework used for developing Java Enterprise Edition (JEE) Applications.

## Why should we learn Spring Framework?

* Spring is a big framework, it can be used to create one full application
* It is the most popular application development framework for enterprise Java.
* Millions of developers around the world use Spring Framework to create high performing, easily testable, and reusable code.

## Features of Spring Framework

### Lightweight

The Spring Framework is very lightweight with respect to its size and functionality.

1. This is due to its POJO implementation, which doesn’t force it to inherit any class or implement any interfaces.
2. We don't need all of Spring to use, only part of it. For example, we can use Spring JDBC without Spring MVC.
3. It follows modular approach: Spring provides various modules for different purposes; we can just use certain according to our required module.

Dependency Injection (DI)

#### What is DI?

Dependency Injection is a design pattern. It is not just specific to Java, it is used in planning software architecture and therefore it can be applied in any language

1. Suppose there are 2 classes: class A and class B.
2. Now class B is a member of class A, i.e. class A has “has-a” relationship with class B.
3. Class A will be known as dependent class and,
4. Class B will be known as a dependency
5. When we inject (supply/assign) a value to B’s reference in class A, it is known as injecting a dependency.

**Let us see an example: DIExample**

#### Why worry about DI?

##### Tight Coupling

As we saw in the example, a change in class B forced us to change the coding of class A. This interdependency of classes is known as coupling and when one change forces us to change other classes’ code, it is known as tight coupling. In tight coupling, we have to change the code in both the classes and recompile the source

##### Loose Coupling

When we build large applications, scalability and maintainability are important aspects to keep in mind. This is where design patterns come into picture. These design patterns help us to develop an application that is easy to maintain and scale. With the help of Spring Framework, we can develop loosely coupled applications where the change in one class does not affect the other class. In loose coupling, we only need to recompile the changed code

Inversion of Control (IOC)

Spring Framework takes control of creating objects and injecting dependencies itself dynamically on run time, i.e. it uses the Dependency Injection design pattern to loosely couple the application code. This is known as Inversion of Control (control is shifted from programmer to Spring Framework).

#### Spring/IOC/DI Container

Spring Framework provides us with Spring/IOC/DI container:

* To create and manage the life cycle of Java Beans
* To manage configuration of application objects.
* To achieve Dependency Injection
* Spring Container is also known as **IOC Container** or **DI container**
* With the help of either of the following ways, we can specify dependencies in a project:
  + Configuration through XML
  + Configuration through Java Class
  + Configuration through Annotations

## So, what Spring does?

Suppose we have following layers & classes in our JEE application:

|  |  |
| --- | --- |
| Layer | Classes |
| UI Layer | HTML/JSP Page |
| Business Layer | ProductController class |
| Service Layer | ProductService class |
| Data Access Layer | ProductDao, Product Model class |
| DataBase | Data |

* All the above dependencies will be managed by Spring
* Spring will inject the object of:
  + Product object in ProductDao class
  + ProductDao object in ProductService class
  + ProductService object in ProductController class
* We will have to just maintain config data in xml or annotation

## What makes Spring Framework special?

### Spring is Open Source

* Spring is freely available. We can learn it, develop in it with absolutely no cost up front
* Strong community to support the developers
* Continuous updates
* Wide adaptation in the industry

### Framework of Frameworks

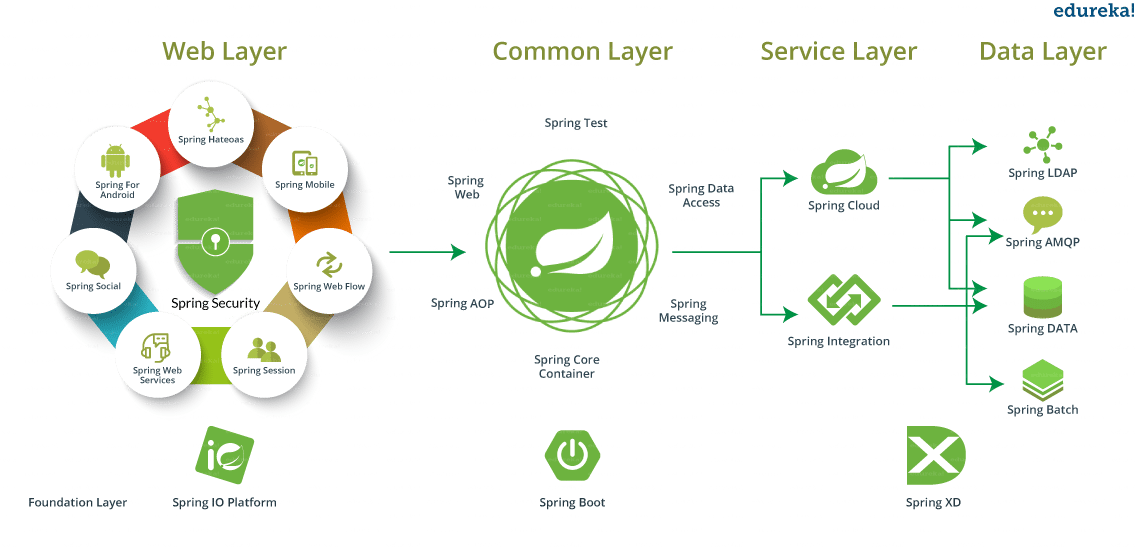
Spring is not just another java framework. It is a framework of frameworks. It provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF, etc.

### Spring has evolved over time

Since its origin till date, Spring has spread its popularity across various domains. Spring Framework now is the foundation for various other Spring Projects that have come up in the offerings in the last two to three years.

### Easy Development & Testing Cycles

Spring Framework follows best practices for coding and testing thereby making the development of Java EE application easy.



## History of Spring Framework

Spring was initially developed by Rod Johnson in 2002

1. The first version of the Spring framework was written by **Rod Johnson** along with a book – “*Expert One-on-One J2EE Design and Development*” in October 2002.
2. The framework was first released in **June 2003** under the **Apache license version 2.0.**
3. The first milestone release of **Spring framework (1.0)** was released in March **2004.**
4. Spring 2.0, which came in 2006, simplified the XML config files.
5. Spring 2.5, which came in 2007, introduced annotation configurations.
6. Spring 3.2, which came in 2012, introduced Java configuration, had support for Java 7, Hibernate 4, Servlet 3.0, and also required a minimum of Java 1.5.
7. Spring 4.0, which came in 2014, had support for Java 8.
8. Spring Boot also was introduced in 2014.
9. Spring 5.0 came out in 2017. Spring Boot 2.x has support for Spring 5.

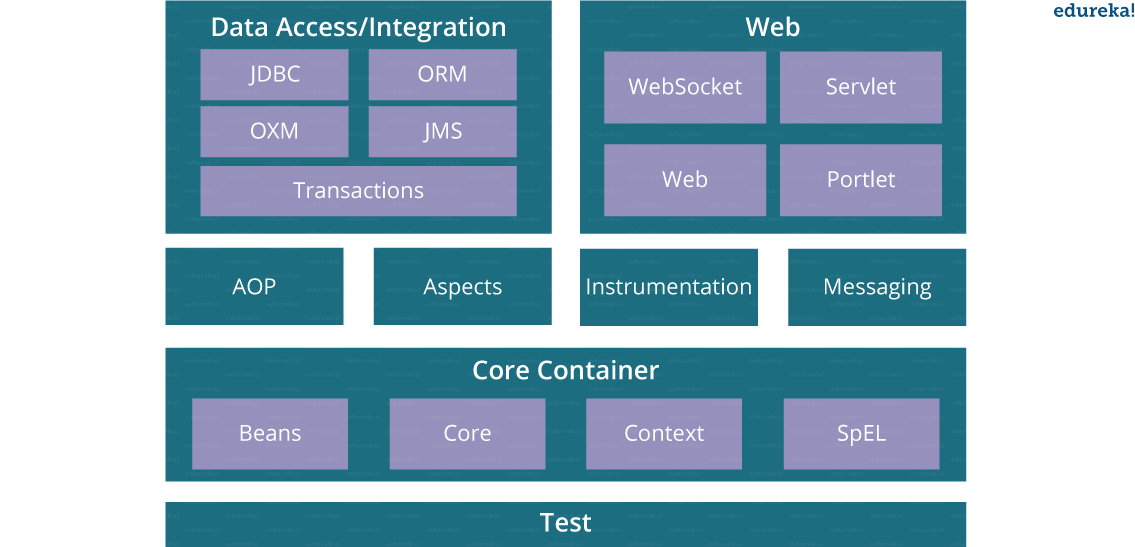
**Official Website for Spring Framework:** [**https://spring.io/**](https://spring.io/)

## Spring Framework Architecture

Spring Framework architecture is an arranged layered architecture that consists of different modules. All the modules have their own functionalities that are utilized to build an application.

There are around **20 modules** that are generalized into **Core Container**, **Data Access/ Integration**, **Web**, **AOP** (Aspect Oriented Programming), **Instrumentation**, and **Test**.

Here, the developer is free to choose the required module. Its modular architecture enables integration with other frameworks without much hassle.



### Spring Core Container

This container has the following **four modules**:

#### Spring Core

This module is the core of the Spring Framework. It provides an implementation for features like IOC (Inversion of Control) and Dependency Injection with a singleton design pattern. The main concepts of Spring core are DI, IOC, Property Injection & Constructor Injection

#### Spring Bean

This module provides an implementation for the factory design pattern through BeanFactory. It is an important module and is used to inject objects in other class

#### Spring Context

Context module inherits features from bean module and adds to it: internationalization, event propagation, resource loading, and transparent creation of context.

This module is built on the solid base provided by the Core and the Bean modules and is a medium to access any object defined and configured.

#### Spring Expression Languages (SpEL)

This module is an extension to expression language supported by Java server pages. It provides a powerful expression language for querying and manipulating an object graph, at runtime.

### Spring Data Integration and Data Access

It consists of the following five modules:

#### Spring JDBC

This module provides JDBC abstraction layer which eliminates the need of repetitive/hideous and unnecessary exception handling overhead JDBC code.

#### Spring ORM

ORM stands for Object Relational Mapping. This module provides consistency/portability to our code regardless of data access technologies based on object oriented mapping concept.

It provides integration of any other ORM tool in our spring application such as integration of hibernate to our code

#### Spring OXM

OXM stands for Object XML Mappers. It is used to convert the objects into XML format and vice versa. The Spring OXM provides a uniform API to access any of these OXM frameworks. It provides an abstraction layer to integrate with object xml mapping tools such as Castor, Xstream etc.

#### Spring JMS

JMS stands for Java Messaging Service. This module contains features for producing and consuming messages among various clients/java applications.

#### Transaction

This module supports programmatic and declarative transaction management for classes that implement special interfaces and for all your POJOs. All the enterprise level transaction implementation concepts can be implemented in Spring by using this module.

### Spring Web

Web layer includes the following modules:

#### Web

This module uses servlet listeners and a web-oriented application context to provide basic web-oriented features to help us create web projects. We can use MVC, Rest APIs, Socket, Portlets, and Http Client etc. in the applications

#### Web-Servlet

This module contains Model-View-Controller (MVC) based implementation for web applications. It provides all other features of MVC, including UI tags and data validations.

#### Web-Socket

This module provides support for WebSocket based and two-way communication between the client and the server in web applications.

#### Web-Portlet

This module is also known as the Spring-MVC-Portlet module. It provides the support for Spring-based Portlets and mirrors the functionality of a Web-Servlet module.

### Web Services and Rest API

### Aspect-Oriented Programming (AOP)

AOP language is a powerful tool that allows developers to add enterprise functionality to the application such as transaction, security etc. It allows us to write less code and separate the code logic. AOP uses cross-cutting concerns, defines method interceptors, point cuts, code decoupling.

When we want to do something before or after function calls, this is known as method interceptor.

### Instrumentation

This module provides class instrumentation support and class loader implementations that are used in certain application servers.

### Messaging

It serves as a foundation to build a message based application. It uses annotations to do so. It could be used whenever we need messaging functionality in applications

### Test

This module supports the testing of Spring components with JUnit or TestNG. It provides consistent loading of Spring **ApplicationContexts** and caching of those contexts. It also provides mock objects that we can use to unit test our code in isolation.

## What are Spring Framework, Spring JDBC, Spring MVC and Spring Boot?

### Spring Framework

Spring Framework is a collection of modules. Spring JDBC and Spring MVC are modules of Spring Framework.

### Spring JDBC

Spring JDBC Framework takes care of all the low-level details starting from opening the connection, preparing and executing the SQL statement, processing exceptions, handling transactions, and finally closing the connection.

### Spring MVC

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

### Spring Boot

Over the past few years, due to added functionalities, the Spring framework has become increasingly **complex**. It requires going through a lengthy procedure in order to start a new Spring project. To avoid starting from scratch and save time, Spring Boot has been introduced. This uses the Spring framework as a foundation.

While the Spring framework focuses on providing flexibility to you, **Spring Boot aims to shorten the code length** and provide you with the easiest way to develop a web application. With annotation configuration and default codes, Spring Boot shortens the time involved in developing an application. It helps create a stand-alone application with less or almost zero-configuration.

# Prerequisites

To learn Spring Framework, one must have knowledge of following technologies:

* Core Java - OOP concepts: class, objects, constructor, method, overloading, overriding
* JDBC: Data integration layer/Access layer for Spring JDBC
* Hibernate to learn Spring ORM: Spring Hibernate Integration
* Servlet and JSP: Spring Web MVC
* Important Web and Database related terms: HTML, CSS, JS, Bootstrap, MySQL/PostGres/any database => working knowledge