2022

Ruhan Potgieter

Belgium Campus

Monday, August 22, 2022

PRG381- Project



Contents

[Introduction about the spring boot framework and how it facilitates Aspect-Oriented Programming. 2](#_Toc112499615)

[1.1.1 Core Container 3](#_Toc112499616)

[1.1.2 Data Access/Integration 3](#_Toc112499617)

[1.1.3 Web 4](#_Toc112499618)

[1.1.4 AOP (Aspect Orientated Programming) and Instrumentation 4](#_Toc112499619)

[1.1.5 Test 4](#_Toc112499620)

[Spring Boot with Spring Framework 5](#_Toc112499621)

[1.2 AOP (Aspect Oriented Programming) with Spring Boot Framework 6](#_Toc112499622)

[Introduction 6](#_Toc112499623)

[Index 6](#_Toc112499624)

[Figures 7](#_Toc112499625)

[References 7](#_Toc112499626)

# Introduction about the spring boot framework and how it facilitates Aspect-Oriented Programming.

What is spring framework, it is a Java platform that supply’s a comprehensive infrastructure support for developing Java applications. Spring boot basically handles the infrastructure of the project so that you as the developer can keep your focus on the application. The following are things you can take advantage of in the spring boot platform have a java method execute in a database transaction without dealing with a transaction API, make use of a local Java method be a remote procedure without dealing with remote APIs, make use of a local Java method as a management operation without dealing with JMX APIs and make use of local Java method as a message handler without dealing with JMS APIs.

Spring boot framework consists of features arranged into around 20 modules and these modules are grouped into Core container, Data Access/Integration, Web, AOP(Aspect Orientated Programming), Instrumentation and Test the following diagram will explain visually.

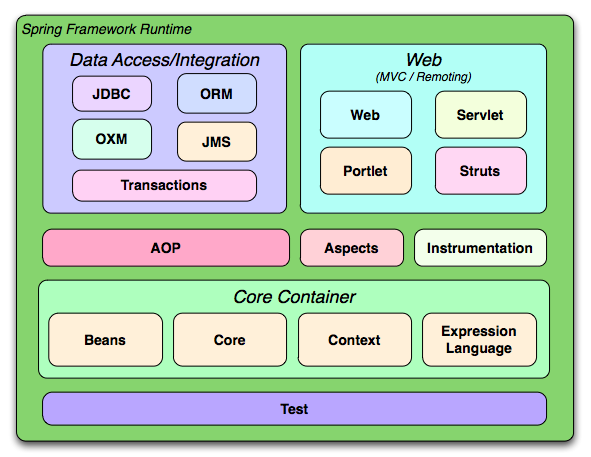


Figure : Spring boot Framework overview

## 1.1.1 Core Container

The Core, Beans, Context, and Expression Language modules make up the Core Container.

The core components of the framework, such as IoC and Dependency Injection, are provided by the Core and Beans modules. A complex application of the factory design is the BeanFactory. It does away with the requirement for programmatic singletons and enables you to separate the specification and setting of dependencies from the logic of your real program.

The Context module offers a way to access objects in a framework-style manner that is comparable to a JNDI registry, and it builds on the strong foundation offered by the Core and Beans modules. The Context module offers support for internationalization (using, for example, resource bundles), event propagation, resource loading, and the transparent generation of the Beans module's features.

For accessing and modifying an object graph at runtime, the Expression Language module offers a robust expression language. It is an expansion of the JSP 2.1 specification's unified expression language (unified EL). Invoking methods, accessing the context of arrays, collections, and indexers, logical and arithmetic operators, named variables, and retrieving objects by name via Spring's IoC container are all supported by the language. Along with common list aggregations, it also provides list projection and selection.

1.1.2 Data Access/Integration

The JDBC, ORM, OXM, JMS, and Transaction modules make up the layer for data access and integration.

The JDBC module offers a JDBC-abstraction layer that eliminates the need for tiresome JDBC coding and error code parsing particular to each database vendor.

The popular object-relational mapping APIs JPA, JDO, Hibernate, and iBatis all have integration layers that are provided by the ORM module. All these O/R-mapping frameworks may be combined with every other Spring feature, including the previously mentioned straightforward declarative transaction management feature, by using the ORM package.

For JAXB, Castor, XMLBeans, JiBX, and XStream, the OXM module offers an abstraction layer that supports implementations of Object/XML mapping.

There are functionalities for creating and consuming messages in the Java Messaging Service (JMS) module. The Transaction module allows declarative and programmatic transaction management for all of your POJOs and classes that implement specific interfaces (plain old Java objects).

## 1.1.3 Web

The Web, Web-Servlet, Web-Struts, and Web-Portlet modules make up the Web layer.

Using servlet listeners and a web-based application context, Spring's Web module may initialize the IoC container and provide basic web-oriented integration features like multipart file upload functionality. It also includes the components of Spring's remoting support connected to the web.

Spring's model-view-controller (MVC) implementation for web applications is contained in the Web-Servlet module. The Spring MVC framework, which combines with all the other Spring Framework capabilities, offers a clear division between domain model code and web forms.

The support classes for integrating a traditional Struts web tier within a Spring application can be found in the Web-Struts module. Keep in mind that as of Spring 3.0, this support has been deprecated. Think about moving your application to a Spring MVC solution or Struts 2.0 with its Spring integration.

The Web-Portlet module mimics the capabilities of the Web-Servlet module and offers the MVC implementation for usage in a portlet context.

## 1.1.4 AOP (Aspect Orientated Programming) and Instrumentation

The AOP module in Spring offers an aspect-oriented programming implementation that complies with the AOP Alliance. You may construct things like method interceptors and pointcuts to neatly decouple code that implements functionality that ought to be separated. You can also embed behavioural data into your code using source-level metadata functionality, much as how.NET attributes work.

Integration with AspectJ is provided through the standalone Aspects module.

For use in some application servers, the Instrumentation module offers support for class instrumentation and class loader implementations.

## 1.1.5 Test

The JUnit or TestNG testing of Spring components is supported by the Test module. Spring Application Contexts are consistently loaded, and those contexts are cached. Additionally, it offers mock objects so you may test your code independently.

# Spring Boot with Spring Framework

Faster development and ease of use are two of Spring Boot's main advantages over Spring Framework by itself. The increased freedom you have from interacting directly with Spring Framework is theoretically sacrificed to achieve this.

But adopting Spring Boot is worth the sacrifice unless you want or want to develop a special configuration. You can still make advantage of the well-regarded annotation system in the Spring Framework, which makes it simple to add additional dependencies (not covered by Spring Starters) to your application. Additionally, you continue to have access to all the features of the Spring Framework, such as simple event handling, validation, data binding, type conversion, built-in security, and testing capability. In conclusion, Spring Boot can greatly speed up development if even just one Spring Starter can handle the breadth of your project.

#### Features that spring boot has over Spring framework

* Autoconfiguration
* Opinionated approach with configuration
* Creating standalone applications

#### Autoconfiguration

Applications are initialized with pre-set dependencies through autoconfiguration, so you don't have to explicitly setup them. Due to built-in autoconfiguration features, Java Spring Boot configures the underlying Spring Framework and third-party packages according to your settings (and based on best practices, which helps avoid errors). Even though you can change these defaults after initialization is finished, Java Spring Boot's autoconfiguration feature makes it easier for you to get started creating Spring-based apps quickly and lowers the likelihood of human error.

#### Opinionated approach with configuration

Based on the requirements of your project, Spring Boot takes an opinionated approach to adding and setting start-up dependencies. Instead of having you to make all those decisions and configure everything manually, Spring Boot decides which packages to install and which default values to use.

During the start-up process, when you select from a variety of starter dependencies known as Spring Starters that address common use cases, you can specify the requirements of your project. You can use Spring Boot Initializer without writing any code by completing a short web form.

#### Standalone applications

Spring Boot enables programmers to build ready-to-use applications. By integrating a web server like Tomcat or Netty inside your app during the start-up phase, it specifically enables you to construct standalone applications that function independently, without relying on an external web server. As a result, by just selecting the Run command, you may run your program on any platform.

## 1.2 AOP (Aspect Oriented Programming) with Spring Boot Framework

### Introduction

By offering an alternative perspective on program structure, aspect-oriented programming (AOP) enhances object-oriented programming (OOP). In OOP, the class is the primary unit of modularity, but in AOP, the aspect is the primary unit of modularity. Concerns that straddle numerous kinds and objects, like transaction management, can be modularized thanks to aspects. (In the AOP literature, such worries are frequently referred to as crosscutting worries.)

The AOP framework is one of many, Spring's elements but AOP framework is an essential element. AOP is not required to utilize the Spring IoC container, therefore you are free to forego using it if you like. However, AOP works well with Spring IoC to provide a powerful middleware solution.

# Index

AOP ……………………………………………………. Aspect Oriented Programming

JMS …………………………………………………….. Java Message Service

API ……………………………………………………… Application Programming Interface

JMX …………………………………………………….. Java Management Extensions

XML …………………………………………………….. Extensible Markup Language

JAXB ……………………………………………………. Java Architecture for XML Binding

Castor ………………………………………………….. Data Binding Framework for Java

XMLBeans ……………………………………………… A tool for access to XML in Java

JIBX …………………………………………….. A tool for binding XML data to java objects

XStream …………………………………………… Java library to serialize objects to XML

OXM ……………………………………………….. Spring Object XML Mappers

POJOs ………………………………………………………. Plain Old Java Object

JPA ………………………………………………………….. Java Persistence API

JDO …………………………………………………………. Java Data Objects

ORM ………………………………………………………… Object Relational Mapping

MVC ………………………………………………………… Model View Controller

JDBC ………………………………………………………… Java Database Connectivity

IoC …………………………………………………………… Inversion of Control

# Figures

[Figure 1: Spring boot Framework overview 2](#_Toc112499513)

# References

Anon., n.d. *AOP with Spring Framework.* [Online]   
Available at: https://www.tutorialspoint.com/spring/aop\_with\_spring.htm  
[Accessed 19 August 2022].

Education, I. C., 2020. *Java Spring Boot.* [Online]   
Available at: https://www.ibm.com/cloud/learn/java-spring-boot#:~:text=Java%20Spring%20Boot%20(Spring%20Boot,ability%20to%20create%20standalone%20applications  
[Accessed 19 August 2022].

GeeksForGeeks, 2019. *Introduction to Spring Boot.* [Online]   
Available at: https://www.geeksforgeeks.org/introduction-to-spring-boot/  
[Accessed 19 August 2022].

Jaiswel, S., 2011-2021. *Javatpoint.* [Online]   
Available at: https://www.javatpoint.com/  
[Accessed 26 08 2022].

Rod Johnson, J. H. K. D. C. S. R. H. A. A. T. R. D. D. D. K. M. P. T. T. E. V. P. T. B. H. A. C. J. L. C. L. M., 2004-2010. *Reference Documentation.* [Online]   
Available at: https://docs.spring.io/spring-framework/docs/3.0.x/spring-framework-reference/html/index.html  
[Accessed 26 08 2022].

Rojas, D. A., 2021. *Spring Core: Including Aspect-Oriented Programming in your skills.* [Online]   
Available at: https://medium.com/javarevisited/spring-core-including-aspect-oriented-programming-in-your-skills-2c37eaa75c2a  
[Accessed 19 August 2022].