===========

**Spring Boot**

===========

=> Spring Boot is one approach to develop Spring Based Applications with less configurations.

Spring Boot = Spring Framework - Xml Configurations

=> Spring Boot is not replacement for Spring Framework. Boot developed on top of Spring Framework

Note: All Spring Framework concepts can be used in Spring Boot also.

=======================

Spring Boot - Advantages

=======================

1) Less Configuration & No Xmls Configurations

2) Pom Starters to simplify dependencies configuration

Ex: web-starter, jpa-starter, security-starter, mail-starter

3) Auto Configuration

4) Embedded Servers (Ex: Tomcat, Jetty, Netty)

5) Actuators (Production Ready Features)

============================================================================================

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

============================================================================================

Spring Boot 1.0 released in 2014

Current version of Spring Boot is 4.x ===> Nov-2023

Note: Java 17 is mandatory to work with Spring Boot 4.x version

=============================

Spring Boot Application Creation

=============================

=> We can create boot application in 2 ways

1) Initializer website (start.spring.io)

2) Spring Starter Project in IDE

Note: If we try to create boot application using IDE then internally IDE will communicate with Intializr website to create the project.

Note: Internet Connection is mandatory for system to create Spring Boot Application.

=============================================

Options to choose While creating boot application

=============================================

Build Tool : Maven / Gradle

Language : Java / Groovy / Kotlin

groupId : It represents company name / Project Name

Ex: com.wipro

artifactId : It represents project name / module name

Ex: MyFirstApp / UserManagementApp / TicketBookingApp

version : 0.0.1 - SNAPSHOT

SNAPSHOT - Project under development

RELEASE / FINAL - Project development completed

packageName : It represents base package in the project

Ex: in.com

==================================

Spring Boot Application Folder Structure

==================================

src/main/java : To keep our project source code

- Application.java : It is start class of the spring boot (main class)

src/main/resources : To keep project configuration files

- application.properties / yml

src/test/java : To keep junit code (unit testing)

- ApplicationTest.java

src/test/resources : Unit testing related config files goes here ###TBD

Maven dependencies : Downloaded jars will be available here

pom.xml : Maven configuration file (dependencies)

=======================

Spring Boot Start Class

=======================

@SpringBootApplication

public class Application {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

=> It is entrypoint for boot application execution

=> @SpringBootApplication annotation is equal to below 3 annotations

@SpringBootConfiguration

@EnableAutoConfiguration

@ComponentScan

=> Spring Boot start class will act as Configuration class because of @SpringBootConfiguration annotation

=> In Spring boot application auto configuration feature will be available because of @EnableAutoConfiguration annotation

=> Component Scan will be performed in spring boot because of @ComponentScan annotation

Note: Package Naming convention will play important role in component scanning

basePackage : in.com

in.com.service -- will be scanned

in.com.dao -- will be scanned

com.com.util ---- will not get scanned