**What is a Framework?**

**Framework is a special type of software that is built on top of other technologies. It has the ability to generate the most commonly used logics automatically. It also provides abstraction on other technologies to simplify application/software development. For example: Spring, Struts, JSF Frameworks created on top of Servlet & JSP technologies. Spring JDBC module created on top of Spring and JDBC technology.**

**Note: JSP, Servlet, JDBC are the Java Technologies, Java is a programming language, whereas Spring, Struts, JSF are Frameworks.**

**What is a Spring Framework in Java?**

**Spring Framework is a special type of light weight open source software built on top of Servlet & JSP technologies. It is used to develop a complete Java application without letting developers worry about frequently used logics. It provides basic functionality support for developing all kinds of Java applications. In other words, Spring handles the basic infrastructure. So you as a developer needs to focus on your application’s business logic only.**

**Why is Spring Framework so popular ?**

**It has several dominating features in comparison to other frameworks in Java. It is a highly acceptable framework in java by the industry because of below features:**

**Light-Weight: Spring has dominated EJB in no time because of its light-weight feature. EJB was heavy-weight in comparison to Spring.**

**Open-Source: It is open source software, easily installable, so adapted by industry very quickly.**

**Modular Architecture: It has multiple modules, we can use each module independently as per our requirement.**

**Easy to develop Application: As it already has frequently used functionalities of an application built-in into it by default, the developer needs to focus on application’s business logic only.**

**Fast development: Spring framework modeled on the basis of real time use cases, so developing projects become very fast & easy.**

**Dependency Injection Feature:  Using**[**dependency injection**](https://javatechonline.com/spring-dependency-injection/)**feature, we just need to instruct the Spring container when to instantiate which object via xml or annotations, it creates objects for us. We don’t have to create objects manually. Dependency Injection also promotes**[**loose coupling**](https://javatechonline.com/oops-design-principles/#What_is_Loose_Coupling)**between components in an application.**

**Using interfaces instead of classes: It provides good practice such as programming using interfaces instead of classes.  It facilitates developers to develop enterprise applications using POJO and POJI model programming.**

**Easy test support: The Spring Test module provides support for an easy-to-test code.**

**Extensive Ecosystem: Spring has a massive ecosystem that helps in extending outside the core framework. It incorporates several projects and libraries, such as Spring Boot, Spring MVC, Spring Data, Spring Security, and Spring Cloud, which provide further functionality. The whole Spring ecosystem offers developers an ample amount of tools and options to improve productivity and sort out common business development challenges.**

**AOP Support: It reduces boilerplate code via**[**AOP**](https://javatechonline.com/how-to-implement-aop-in-spring-boot-application/)**(Aspect Oriented Programming) support. Boilerplate code is the code that repeats across multiple applications or in the multiple parts of the project with no change or very minimal change. For example, common connection preparation logic in JDBC comes under boilerplate code. By applying AOP, developers can flawlessly integrate features like logging, security, and transaction management into their applications without modifying the primary structure of the code base.**

**Integration Capabilities: Spring Framework offers support for integrating with other technologies and frameworks. It provides integration with popular frameworks such as Hibernate, JPA, JUnit, and much more. This interoperability enables developers to smoothly integrate Spring into the required software stack.**

**Backward Compatibility: Spring Framework has a powerful adherence to backward compatibility. It ensures that applications built on older versions of the framework can flawlessly migrate to newer versions without severe code changes. This behavior makes developers and organizations more confident to rely on the stability and sustainability of their Spring-based applications.**

**Community Support: A massive and active community of developers also contributes to the popularity of the Spring Framework. The community provides detailed documentation, tutorials, blogs, white papers, forums, and resources, making it easier for developers to learn the framework effectively. The**[**Spring community**](https://spring.io/community)**also aggressively contributes to the improvements, development of new features, and bug fixes, and ensures the framework stays relevant and up-to-date.**

**Now-a days companies get target to complete a project in very less time, but without any compromise on the quality. In such type of situations Spring Boot is the best option to use, which is built on top of the Spring Framework.**

**Advantages Of Spring Boot in Java**

**Those who are working in Java or even who just started learning Java programming, must have heard about the term Spring Boot. Those who are preparing for interviews either Freshers or experienced must also learn this term. Many a time interviewers are also asking ‘What is Spring Boot ?’. Nothing more to say how much important it is. Even it is also needless to say, how much importance this term has in a Java Developer’s life.**

**What is Spring Boot ?**

**Spring Boot is a combination of two words “Spring” and “Boot”. Here, the word ‘Spring’ represents a Spring Application. The word ‘Boot’ represents bootstrap. Hence, Spring Boot is an open-source java-based framework that helps in bootstrapping the Spring Application.**

**Here is the official definition of spring.io website : “Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can just run”.**

**Let’s understand it word by word now. Spring Boot simplifies the process of creating a Spring based Application. Now what are the characteristics of the Spring based Application that Spring Boot creates? That application is stand alone, production ready and just runnable. We don’t have to worry about making jar/war and deploying it into any web server. These tasks and configurations are done by Spring Boot internally. Moreover, It has embedded web servers in the form of jar files that internally handle the process of running the application.**

**What are the Core Principles of Spring Boot?**

**Spring Boot is based on four core principles: autoconfiguration, opinionated, convention over configuration, and eliminates boilerplate code.**

**Autoconfiguration**

**Spring Boot provides a feature called autoconfiguration. In simple words, it automatically configures the application based on the dependencies that are provided to it. This means that developers do not need to spend time configuring the application themselves. Instead, Spring Boot will configure the application based on the reasonable default settings that are appropriate for most of the applications.**

**Opinionated**

**Spring Boot is an opinionated framework, which means that it provides opinions about the best way to configure and run the application. This simplifies the development process because developers do not need to make all the decisions themselves. Instead, they can rely on Spring Boot’s opinions to guide them in the right direction.**

**Convention over Configuration**

**Spring Boot provides acceptable default settings for configuration, which means that developers only need to provide configuration when they want to override the defaults. This simplifies the development process because developers do not need to spend time configuring the application themselves.**

**Eliminates Boilerplate Code**

**Spring Boot reduces the amount of boilerplate code that developers need to write by providing a set of defaults and abstractions. This means that developers can focus on writing business logic instead of writing boilerplate code. For example, unlike JDBC, we don’t need to write a lengthy code just to get a connection with the database. Instead, we just need to provide values of database properties in 3-4 lines via the properties file to get a database connection.**

**Why is Spring Boot so Popular?**

**If we compare with other frameworks in the Java programming language, Spring Boot became very popular in a very short duration. For example, below are some points that make Spring Boot a popular framework to develop a production-ready application.**

**1. Spring Boot speeds up the project development.**

**2. It provides an easier and faster way to configure, set up & run both web-based as well as simple applications.**

**3. No web servers like (Tomcat, Jetty, Undertow) are required to configure separately. Therefore, no need to deploy any war file manually.**

**4. Reduces XML configurations and promotes annotations.**

**5. Combines more than one annotations of Spring framework & replaces them by introducing a single annotation. For example,**[**@SpringBootApplication is a combination of three annotations**](https://javatechonline.com/spring-boot-annotations-with-examples/#SpringBootApplication_Configuration_ComponentScan_EnableAutoConfiguration)**: @EnableAutoConfiguration, @Configuration, @ComponentScan**

**6. Reduces the efforts of developer to deliver an enterprise-level application.**

**7. Provides ready-made functionalities that are the most common and applicable to any project. Hence, it eliminates boiler plate code.**

**8. Creates stand-alone application that can be run using jar file.**

**9. Automatically configures Spring and third party libraries whenever possible.**

**10. Provides production-ready features such as health checks, metrics and externalized configurations.**

**11. Provides easy integration mechanism with many third party tools.**

**12. Spring Boot offers a great support to develop**[**Microservices in Java**](https://javatechonline.com/microservices-in-java/)**.**

**13. Spring Boot Application easily integrates with its ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security, Spring Web MVC etc.**

**14. It integrates with databases very easily. Sometimes, no need to provide database specific properties like driver, dialects etc., as it automatically picks these property values from database URL.**

**15. It provides developers ready-made starter projects to quickly work on functionalities rather than wasting time in setting up initial configurations.**

**16. Spring Boot also provides command line interface(CLI) to develop and test applications quickly.**

**17. It offers support of most commonly used build tools like Maven & Gradle.**

**How Does Spring Boot Work?**

**Spring Boot works by offering a set of starter dependencies that we can use to quickly bootstrap a Spring-based application. These starter dependencies incorporate all the required libraries and configurations necessary to get an application up and running. Developers can use these starter dependencies to create a new project and get started quickly.**

**Spring Boot also includes a variety of features that assist developers to build and deploy applications more easily. For example, it offers a built-in web server (Tomcat, Jetty, or Undertow) that can be used to deploy applications without the need for an external server. It also provides various tools and plugins that simplify the development process, such as the Spring Boot CLI, which permits developers to create, run, and test applications from the command line as well.**

**What are the features of Spring Boot?**

**What are the advantages of Spring Boot?**

**1) Spring Boot Starters**

**Spring Boot provides us a choice to add ready made mini starter projects to simplify the process of development. We can add them either at the beginning or even in the middle of development. For example, Spring Web, Spring Data JPA, Spring Security, Spring Boot Dev Tools, MySQL Driver, Java Mail Sender, Eureka Server etc. are some of the starter projects. It also provides flexibility to add other third party libraries in the form of starter projects like**[**Lombok**](https://javatechonline.com/how-to-reduce-boilerplate-code-in-java-using-lombok-annotations/)**.**

**2) Reducing Boilerplate Code**

**Spring Boot helps to avoid all the manual work of writing boilerplate code, annotations, and complex XML configurations. It internally consumes Spring framework’s API and provides easy to use API that saves us from writing boilerplate code again & again. Moreover, It also consumes multiple Spring Framework’s annotations and provides a single annotation that saves us from manual configurations including XML based configurations. For example,**[***@SpringBootApplication***](https://javatechonline.com/spring-boot-annotations-with-examples/#SpringBootApplication_Configuration_ComponentScan_EnableAutoConfiguration)**annotation is a combination of three annotations provided by Spring framework. They are *@Configuration, @ComponentScan,* and *@EnableAutoConfiguration.*Here *@SpringBootApplication* annotation is specific to SpringBoot.**

**3) Embedded Servers**

**Spring Boot comes with embedded HTTP servers like Tomcat, Jetty and Undertow. These servers come in the form of jars. However, Tomcat is by default enabled by Spring Boot. We can enable Jetty and Undertow easily if required. In short, we can easily enable or disable these servers based on our requirements.**

**4) Easier to connect with Databases**

**It allows connecting easily with databases like Oracle, PostgreSQL, MySQL, MongoDB, Redis etc. Moreover, it also allows easy integration with queue messaging services like ActiveMQ, Rabbit MQ,  ElasticSearch and many more.**

**5) Supports development of Microservices**

**It works well with microservices. The Spring Boot artifacts can be deployed directly into Docker containers.**

**6) Default setup for Testing**

**Spring Boot facilitates the creation and testing of Java-based applications by providing a default setup for unit and integration tests.**

**7) Run Application as a jar file**

**It provisions to run the jars independently using the command *java -jar.*Hence*,*even we can run a web MVC application using the command*java -jar.***

**8) Spring Boot Profile to easy switch environment**

**It’s profile concept simplifies the process to run our application in different environments by just changing profile name & server details.**

**9) Spring Boot Actuator for Application Monitoring**

**Apart from the application development, Spring Boot offers us the capability to inspect the internal working of the application through Spring Boot Actuator. Spring Boot Actuator provides endpoints for managing and monitoring of our Spring Boot application.**

**10) Increases Productivity and minimizes Development time**

**Needless to say, Spring Boot helps us in increasing productivity and saving development time.**

**What are the disadvantages of Spring Boot?**

**1) One of the disadvantage that many developers face at the time of development is that it includes many additional dependencies which are not required. It increases the deployment file size. However, if we compare with the list of advantages, this disadvantage becomes negligible. If the size criteria are our priority, then this becomes really a big disadvantage.**

**2) Another disadvantage is that there is a challenge to update code of legacy Spring based projects. However, we can overcome this problem by using Spring Boot CLI (Command Line Interface). It will help us to convert legacy code.**

**How many ways are there to create a Spring Boot Application?**

**In order to create a Spring Boot Application, we have multiple ways.**

**1) Using Spring Boot CLI Tool  
2) Using Spring STS IDE  
3) Using Spring**[**Initializr Website**](https://start.spring.io/)

**Out of them, STS (Spring Tool Suite) IDE is the most commonly used way of developing a SpringBoot project in the industry. It has various features and is the most user friendly. Even it also saves our development time**

**What are the Starters in Spring Boot ?**

**In Spring Boot, Starters are the ready made small projects that we can include in our application. Officially they are called dependency descriptors. The starters contain a lot of the dependencies that you need to get a project up and running quickly and with a consistent, supported set of managed transitive dependencies. For example, if you want to get started with Spring Web Application, include the *spring-boot-starter-web* dependency in your project. However, if you are using STS as an IDE, you need to just search for ‘Web’ and select ‘Spring Web’. The STS will automatically add the ‘spring-boot-starter-web’ dependency in your pom.xml.**

**All official starters follow a similar naming pattern; spring-boot-starter-\*, where \* is a particular type of application. This naming structure is intended to help when you need to find a starter. As aforementioned, the Maven integration with many IDEs like STS lets you search dependencies by name.**

**List of Common Starters**

**Below is the list of some application starters provided by SpringBoot under the org.springframework.boot group:**

|  |  |
| --- | --- |
| **Starter Name** | **Purpose** |
| **spring-boot-starter** | **Core starter, including auto-configuration support, logging and YAML** |
| **spring-boot-starter-activemq** | **Starter for JMS messaging using Apache ActiveMQ** |
| **spring-boot-starter-aop** | **Starter for aspect-oriented programming with Spring AOP and AspectJ** |
| **spring-boot-starter-data-jpa** | **for using Spring Data JPA with Hibernate** |
| **spring-boot-starter-batch** | **for using Spring Batch** |
| **spring-boot-starter-cache** | **Starter for using Spring Framework’s caching support** |
| **spring-boot-starter-data-jdbc** | **Starter for using Spring Data JDBC** |
| **spring-boot-starter-data-ldap** | **for using Spring Data LDAP** |
| **spring-boot-starter-data-mongodb** | **for using MongoDB document-oriented database and Spring Data MongoDB** |
| **spring-boot-starter-validation** | **Starter for using Java Bean Validation with Hibernate Validator** |
| **spring-boot-starter-data-elasticsearch** | **Starter for using Elasticsearch search and analytics engine and Spring Data Elasticsearch** |
| **spring-boot-starter-data-rest** | **for exposing Spring Data repositories over REST using Spring Data REST** |
| **spring-boot-starter-freemarker** | **for building MVC web applications using FreeMarker views** |
| **spring-boot-starter-integration** | **Starter for using Spring Integration** |
| **spring-boot-starter-jersey** | **Starter for building RESTful web applications using JAX-RS and Jersey. An alternative to starter-web** |
| **spring-boot-starter-json** | **for reading and writing json** |
| **spring-boot-starter-mail** | **for using Java Mail and Spring Framework’s email sending support** |
| **spring-boot-starter-security** | **Starter for using Spring Security** |
| **spring-boot-starter-test** | **Starter for testing SpringBoot applications with libraries including JUnit Jupiter, Hamcrest and Mockito** |
| **spring-boot-starter-web** | **for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container** |
| **spring-boot-starter-webflux** | **for building WebFlux applications using Spring Framework’s Reactive Web support** |
| **spring-boot-starter-thymeleaf** | **Starter for building MVC web applications using Thymeleaf views** |

**Starter for Oracle Database (Since Spring Boot 3.0.2)**

|  |  |
| --- | --- |
| **oracle-spring-boot-starter-ucp** | **A starter for UCP (Universal Connection Pool) for Spring 3.0.2. This makes it easy to access the Oracle Database from a Spring Boot application.** |

**Starters to get Production Ready features**

**Spring Boot incorporates a number of additional features to help you monitor and manage your application in production. The actuator module offers all of SpringBoot’s production-ready features. The recommended way to enable the features is to add a dependency on the spring-boot-starter-actuator ‘Starter’.**

|  |  |
| --- | --- |
| **spring-boot-starter-actuator** | **Starter for using Actuator which provides production ready features to help you monitor and manage your application** |

**Starters for Swapping or excluding technical features**

**Finally, It also includes the following starters that can be used if you want to exclude or swap specific technical facets.**

|  |  |
| --- | --- |
| **spring-boot-starter-jetty** | **Starter for using Jetty as the embedded servlet container. An alternative to starter-tomcat** |
| **spring-boot-starter-undertow** | **for using Undertow as the embedded servlet container. An alternative to starter-tomcat** |
| **spring-boot-starter-logging** | **for logging using Logback. Default logging starter** |
| **spring-boot-starter-reactor-netty** | **Starter for using Reactor Netty as the embedded reactive HTTP server.** |
| **spring-boot-starter-tomcat** | **Starter for using Tomcat as the embedded servlet container. Default servlet container starter used by starter-web** |
| **spring-boot-starter-log4j2** | **Starter for using Log4j2 for logging. An alternative to starter-logging** |

**Moreover, the complete List of starters is on official**[**spring.io**](https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#getting-started.introducing-spring-boot)**website.**

**Conclusion**

**After going through all the theoretical & some example part of ‘What is Spring Boot & Advantages Of Spring Boot?’, finally, we should be able to answer All About Sprong Boot & Advantages Of Spring Boot?, and also What does it do at all. Therefore, we expect from you to further utilize your knowledge and implement them in your project accordingly. Furthermore, If there is any update in future, we will also update the article accordingly. Moreover, Feel free to provide your comments in comments section below.**