Spring Boot

Spring Boot is an open source Java-based framework used to create a Micro Service. It is developed by Pivotal Team. It is easy to create a stand-alone and production ready spring applications using Spring Boot. Spring Boot contains a comprehensive infrastructure support for developing a micro service and enables you to develop enterprise-ready applications that you can **“just run”**.

You can get started with minimum configurations without the need for an entire Spring configuration setup.

### **Advantages**

Spring Boot offers the following advantages to its developers −

* Easy to understand and develop spring applications
* Increases productivity
* Reduces the development time

## **What is Micro Service?**

Micro Service is an architecture that allows the developers to develop and deploy services independently. Each service running has its own process, and this achieves the lightweight model to support business applications.

### **Advantages**

Micro services offer the following advantages to its developers −

* Easy deployment
* Simple scalability
* Compatible with Containers
* Minimum configuration
* Lesser production time

## **Why Spring Boot?**

You can choose Spring Boot because of the features and benefits it offers as given here −

* It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
* It provides a powerful batch processing and manages REST endpoints.
* In Spring Boot, everything is auto configured; no manual configurations are needed.
* It offers annotation-based spring application
* Eases dependency management
* It includes Embedded Servlet Container

## **How does it work?**

Spring Boot automatically configures your application based on the dependencies you have added to the project by using **@EnableAutoConfiguration** annotation. For example, if MySQL database is on your classpath, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

The entry point of the spring boot application is the class contains **@SpringBootApplication** annotation and the main method.

Spring Boot automatically scans all the components included in the project by using **@ComponentScan** annotation.

# @SpringBootApplication vs @EnableAutoConfiguration annotations in Spring Boot

Even though both **@SpringBootApplication** and **@EnableAutoConfiguration** can be used to enable the **auto-configuration feature of Spring Boot** there is a subtle difference between them. The @SpringBootApplication does much more than what @EnableAutoConfiguration do.

It's a combination of three annotations:

**@Configuration** which is used in Java-based configuration on Spring framework.

**@ComponentScan** to enable component scanning of components you write e.g. [@Controller](http://javarevisited.blogspot.sg/2017/11/difference-between-component-service.html) classes.

**@EnableAutoConfgiuration** itself.