**Auto Configuration**

Spring Boot Auto Configuration automatically configures your Spring application based on the JAR dependencies you added in the project. For example, if MySQL database is on your class path, but you have not configured any database connection, then Spring Boot autoconfigures an in-memory database. For this purpose, you need to add @EnableAutoConfiguration annotation or @SpringBootApplication annotation to your main class file. Then, your Spring Boot application will be automatically configured.

SpringBootApplication

Annotation since version 1.2.0

The entry point of the spring boot application is the class contains @SpringBootApplication annotation and the main method. This class should have the main method to run the Spring Boot application.

If you added @SpringBootApplication annotation to the class, you do not need to add the @EnableAutoConfiguration, @ComponentScan and @SpringBootConfiguration annotation. The @SpringBootApplication annotation includes all other annotations.

**Spring Component**

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

Spring Component annotation is used to denote a class as Component. It means that [Spring framework](https://www.journaldev.com/16922/spring-framework) will autodetect these classes for [dependency injection](https://www.journaldev.com/2410/spring-dependency-injection) when annotation-based configuration and classpath scanning is used.

In layman terms, a Component is responsible for some operations. [Spring framework](https://www.journaldev.com/16922/spring-framework) provides three other specific annotations to be used when marking a class as Component.

1. Service: Denotes that the class provides some services. Our utility classes can be marked as Service classes.
2. Repository: This annotation indicates that the class deals with CRUD operations, usually it’s used with [DAO](https://www.journaldev.com/16813/dao-design-pattern) implementations that deal with database tables.
3. Controller: Mostly used with [web applications](https://www.journaldev.com/14476/spring-mvc-example) or [REST web services](https://www.journaldev.com/2552/spring-rest-example-tutorial-spring-restful-web-services) to specify that the class is a front controller and responsible to handle user request and return appropriate response.

Note that all these four annotations are in package org.springframework.stereotype and part of spring-context jar.

Most of the time our component classes will fall under one of its three specialized annotations, so you may not use @Component annotation a lot.

**Spring Component Example**

<https://www.journaldev.com/21429/spring-component>

# What is a difference between ApplicationContext, AnnotationConfigApplicationContext, and AnnotationConfigWebApplicationContext in Spring?

Answer: <https://qr.ae/pNvYOk>

# **SpringApplication.run()**

You need to run Application.run() because this method starts whole Spring Framework. Code below integrates your main() with Spring Boot.