# Spring Boot

Spring Boot is an open source Java-based framework. 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”**.

Official definition:

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

We take an opinionated view of the Spring platform and third-party libraries, so you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

”

### **Advantages**

Spring Boot offers the following advantages to its developers −

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

### **Goals**

Spring Boot is designed with the following goals −

* To avoid complex XML configuration in Spring
* To develop a production ready Spring application in an easier way
* To reduce the development time and run the application independently
* Offer an easier way of getting started with the application

## 1. What is starter template?

Spring Boot starters are templates that contain a **collection of all the relevant transitive dependencies** that are needed to start a particular functionality. For example, If you want to create a Spring WebMVC application then in a traditional setup, you would have included all required dependencies yourself. It leaves the chances of **version conflict** which ultimately result in more **runtime exceptions**.

With String boot, to create MVC application all you need to import is spring-boot-starter-web dependency.

|  |
| --- |
| pom.xml |
| <!-- Parent pom is mandatory to control versions of child dependencies -->  <parent>      <groupId>org.springframework.boot</groupId>      <artifactId>spring-boot-starter-parent</artifactId>      <version>2.0.4.RELEASE</version>      <relativePath />  </parent>    <!-- Spring web brings all required dependencies to build web application. -->  <dependency>      <groupId>org.springframework.boot</groupId>      <artifactId>spring-boot-starter-web</artifactId>  </dependency> |

## **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.

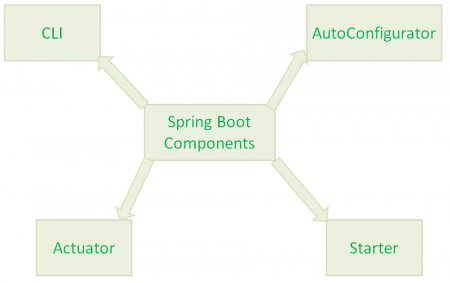
Spring Boot automatically scans all the components included in the project by using **@ComponentScan** annotation.

# Key Components and Internals of Spring Boot Framework

### Key Components of Spring Boot Framework

Spring Boot Framework has mainly four major Components.

* Spring Boot Starters
* Spring Boot AutoConfigurator
* Spring Boot CLI
* Spring Boot Actuator



### Spring Boot Starter

Spring Boot Starters is one of the major key features or components of Spring Boot Framework. The main responsibility of Spring Boot Starter is to combine a group of common or related dependencies into single dependencies.

#### Major Advantages of Spring Boot Starter

* Spring Boot Starter reduces defining many dependencies simplify project build dependencies.
* Spring Boot Starter simplifies project build dependencies.

### Spring Boot AutoConfigurator

To develop a Spring-based application requires lot of configuration (Either XML Configuration of Annotation Configuration). Then how to solve this problem.

The solution to this problem is Spring Boot AutoConfigurator. The main responsibility of Spring Boot AutoConfigurator is to reduce the Spring Configuration. If we develop Spring applications in Spring Boot,then We dont need to define single XML configuration and almost no or minimal Annotation configuration. Spring Boot AutoConfigurator component will take care of providing those information.

And also Spring Boot reduces defining of Annotation configuration. If we use @SpringBootApplication annotation at class level, then Spring Boot AutoConfigurator will automatically add all required annotations to Java Class ByteCode.

SpringBootApplicationAnnotation

### Spring Boot CLI

Spring Boot CLI(Command Line Interface) is a Spring Boot software to run and test Spring Boot applications from command prompt. When we run Spring Boot applications using CLI, then it internally uses Spring Boot Starter and Spring Boot AutoConfigurate components to resolve all dependencies and execute the application.

We can run even Spring Web Applications with simple Spring Boot CLI Commands.

Steps:

It is a tool which you can download from the official site of Spring Framework. Here, we are explaining steps.

Step 1: Download the CLI tool from official site as we are doing here. <https://docs.spring.io/spring-boot/docs/current/reference/html/getting-started-installing-spring-boot.html>

Step 2: After downloading, extract the zip file. It contains a bin folder, in which spring setup is stored. We can use it to execute Spring Boot application.

Step 3: CLI executes groovy files. So, first, we need to create a groovy file for Spring Boot application.

Step 4: Open terminal and cd into the bin location of cli folder.

Step 5: Create a groovy file.

C:\Niket\Java material\Spring\_Boot\_CLI\spring-2.1.1.RELEASE\bin>type nul > SpringBootCLIExample.groovy

Step 6: Edit this file and type

@RestController

class SpringBootCLIExample{

@RequestMapping("/hello")

String index(){

"<h1>Welcome to Spring boot CLI console</h1>"

}

}

Step 7: Run C:\Niket\Java material\Spring\_Boot\_CLI\spring-2.1.1.RELEASE\bin>spring run SpringBootCliExample.groovy

Note:

If getting java.lang.UnsupportedClassVersionError: org/springframework/boot/loader/JarLauncher : Unsupported major.minor version 52.0

Then set “set JAVA\_HOME=C:\Java\jdk1.8.0\_121”

Step 8: You can access the application by <http://localhost:8080/hello>

<https://docs.spring.io/spring-boot/docs/current/reference/html/cli-using-the-cli.html>

### Spring Boot Actuator

Spring Boot Actuator components gives many features, but two major features are

* Providing Management EndPoints to Spring Boot Applications.
* Spring Boot Applications Metrics.

# Spring Initializr

Spring Boot Initializr is used to quick start new Spring Boot Maven/Gradle projects within no time. It generates initial project structure and builds scripts to reduce Development time

Select Maven project and dependencies. Fill other details and click on generate project.

<https://start.spring.io/>

Spring Boot Initializr With Spring Boot CLI

C:\Niket\Java material\Spring\_Boot\_CLI\spring-2.1.1.RELEASE\bin> spring init

(It will create demo.zip in bin folder.)

<https://spring.io/guides/gs/spring-boot/>

## **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

Home work: