

Spring boot

An age of innovation

“The Spring Framework has been around for over a decade and has found a place as the de facto standard framework for developing Java applications. With such a long and storied history, some might think that Spring has settled, resting on its laurels, and is not doing anything new or exciting. Some might even say that Spring is legacy and that it’s time to look elsewhere for innovation.

Some would be wrong.

There are many exciting new things taking place in the Spring ecosystem, including work in the areas of cloud computing, big data, schema-less data persistence, reactive programming, and client-side application development. Perhaps the most exciting, most head-turning, most game-changing new thing to come to Spring in the past year or so is Spring Boot. Spring Boot offers a new paradigm for developing Spring applications with minimal friction.”

-Craig Walls



Spring Boot is an open source Java-based framework used for making application development easy, by avoiding some tedious development steps and creating essential boilerplate code and configuration. More than that Spring Boot is also used heavily for building micro services based applications. It is used for creating web as well as enterprise applications. With the advent of container related technologies such as Docker and Kubernetes, the importance of Spring Boot have increased.

History of Spring boot

Initially spring was launched in 2002 and various frameworks built on top of it such as Spring Boot have come to dominate the way java developers code. Rod Johnson along with Juergen Hoeller and Yann Caroff developed spring as an open source framework. In 2013 , to improved support for ‘containerless’ web application architectures. As earlier there was a problem with the web applications embedded in servlet container which comes with learning curve that we cannot assume new developers will have to overcome. To solve this on August 6,2013 Phil Webb went on to create Spring Boot and fix the problem of spring framework by creating another framework.

How Spring Boot is used to develop a microservice applications?

“Microservices, in a nutshell, allows us to break our large system into a number of independent collaborating components.”

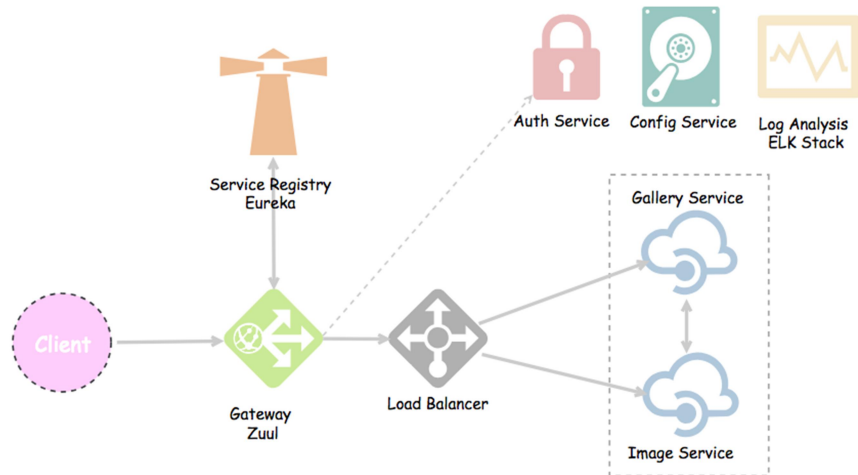


Image to describe a microservice

Spring Boot is used to create many micro service based applications. 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. It's very smart to know how to get them working together can quickly setup services with minimal configurations.

Advantages

Micro services offers the following advantages to its developers –

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

Value proposition of spring boot



Why Spring Boot is an Excellent Solution

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

Advantages of Spring Boot based applications

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 applications in an easier way
- To reduce the development time and run the application independently
- Offer an easier way of getting started with the application

Creating a Spring Boot Application

The best way to create a new Spring Boot application is to use <https://start.spring.io>. This is what is more commonly known as the Spring Initializer. It helps to cook up a working Spring Boot application in almost no time. On the page that opens, you need to fill in a few details. In the field **Group**, enter the high-level package name. This can be the domain. The artifact could be the name of the application. Click on **Switch to Full Version** to see all the details to be entered. Enter the **Group** as *com.progressivecoder.demo* and **Artifact** as *spring-boot-starter*. Note that we have selected the Maven Project at the top. Gradle can also be selected. In the languages dropdown, you could opt for Kotlin or Groovy. In this example, we have selected Java.

See the image below:

Generate a

Maven Project ▾

 with

Java ▾

 and Spring Boot

2.1.1 ▾

Project Metadata

Artifact coordinates

Group

com.progressivecoder.demo

Artifact

spring-boot-starter

Name

spring-boot-starter

Description

Demo project for Spring Boot

Package Name

com.progressivecoder.demo.springbootstarter

Packaging

Jar ▾

Java Version

8 ▾

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Too many options? [Switch back to the simple version.](#)

The next important step is to select the various packages or features in the application. This step is similar to selecting the dependencies in a typical Maven or Gradle project. In the dialog box, scroll down to see the entire list of curated dependencies available. Select a few important dependencies like Web, Rest Repositories, JPA, H2, and Actuator.

At this point, click on the big green button that says Generate Project. Alternatively, you can press Alt + Enter. The project will be downloaded to your computer.