**Spring Boot**

**1. What is Spring Boot?**  
**Answer:** Spring Boot is an extension of the Spring Framework that simplifies application development by providing default configurations, auto-configuration, and an embedded server (like Tomcat). It reduces boilerplate code and avoids complex XML configuration.

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

* Auto-configuration
* Standalone applications
* Embedded servers
* Opinionated defaults
* Production-ready metrics and health checks with **Spring Actuator**

**What is the difference between Spring and Spring Boot?**  
**Answer:**

| **Spring** | **Spring Boot** |
| --- | --- |
| Requires manual configuration | Auto-configuration |
| Needs external server | Comes with embedded server |
| Slower setup | Faster development setup |

**What is @SpringBootApplication?**  
**Answer:** It’s a convenience annotation that combines @Configuration, @EnableAutoConfiguration, and @ComponentScan.

**How do you create a Spring Boot application?**  
**Answer:**

* Use [Spring Initializr](https://start.spring.io/)
* Add @SpringBootApplication to the main class
* Run using SpringApplication.run(MyApp.class, args);

**What is Spring Boot Auto-Configuration?**  
**Answer:** It attempts to automatically configure your application based on the dependencies present in the classpath. For example, if Spring MVC is on the classpath, it auto-configures a DispatcherServlet.

**How does Spring Boot handle dependency management?**  
**Answer:** It uses **Spring Boot Starters**, which are sets of convenient dependency descriptors for various technologies (e.g., spring-boot-starter-web).

**What is Spring Boot Actuator?**  
**Answer:** A module that provides endpoints to monitor and manage the application. For example:

* /actuator/health - shows health info
* /actuator/metrics - exposes metrics

**What is the use of application.properties or application.yml?**  
**Answer:** These files are used to define configuration properties such as server port, database settings, and custom settings.

*server.port=8081*

*spring.datasource.url=jdbc:mysql://localhost:3306/mydb*

**How do you handle exceptions in Spring Boot?**  
**Answer:**

* Use @ControllerAdvice for global exception handling.
* Use @ExceptionHandler to handle specific exceptions.

**What are profiles in Spring Boot?**  
**Answer:** Profiles let you define different configurations for different environments (dev, test, prod).

Set with:

**How is security handled in Spring Boot?**  
**Answer:** Using **Spring Security**, which provides authentication, authorization, and protection against common attacks like CSRF.

**How does Spring Boot support microservices?**  
**Answer:**

* REST APIs using Spring Web
* Service discovery via Spring Cloud with Eureka
* Config management via Spring Cloud Config
* Circuit breakers using Resilience4j or Hystrix

**How do you connect a Spring Boot app to a database?**  
**Answer:** Add JDBC or JPA dependencies, configure datasource in application.properties, and use Spring Data repositories.

**SORT**

**Spring**

Build enterprise level applications

* Focus on business issue and let Spring handle rest

Overview

* Build simple classes (like a business service) with annotations (@service) that define what they are
  + This allows Spring to manage lifecyle of a class
* Allows for dependency injection which allows objects to define own dependencies that the Spring container injects into them
* Can build loosely coupled components that are ideal for [microservices](https://www.ibm.com/topics/microservices) and distributed network applications

**Spring Boot**

Java Spring Boot is an open-source tool that makes it easier to use Java-based frameworks to create microservices and web apps.

* Convention over configuration
* Standalone

Overview

* Large body of pre-written code
* Helps remove configuration, build and deploy steps

Setup

* Maven- Put dependencies in POM.xml
* Gradle- Build tool
* Uses pom.xml
* Add <parent> that has springboot as its parent (Define configuration here and child inherits)

Annotations

* @SpringBootApplication: Main
* @service
* @controller
* @RequestMapping (/api/hi)
* @autowired- Needs dependency injection

**Spring Initializer**

* Quick start
  + Metadata and dependencies you need
  + Java version and Language
* Core: Security
* Web: Web, Web Service Web Socket
* Cloud Messaging:
* Basicaly like NPM

**Spring Boot CLI**

* Run groovy scripts less boilerplate code

**Spring MVC**

* Map Routes to a controller (Java Class)
* Convert to a response

**A picture containing text, screenshot, font, diagram

Description automatically generated**

|  |  |  |
| --- | --- | --- |
|  | **Spring** | **Spring Boot** |
| What is it? | An open-source web application framework based on Java. | An extension or module built on the Spring framework. |
| What does it do? | Provides a flexible, completely configurable environment using tools and libraries of prebuilt code to create customized, loosely coupled web apps. | Provides the ability to create, standalone Spring applications that can just run immediately without the need for annotations, XML configuration, or writing lots of additional code. |
| When should I use it? | Use Spring when you want: | Use Spring Boot when you want: |
| Flexibility | Ease of use |
| An unopinionated approach.\* | An opinionated approach.\* |
| To remove dependencies from your custom code. | To get quality apps running quickly and reduce development time. |
| To implement a very unique configuration. | To avoid writing boilerplate code or configuring XML. |
| To develop enterprise applications. | To develop REST APIs. |
|  |  |
| What's its key feature? | Dependency injection | Autoconfiguration |
| Does it have embedded servers? | No. In Spring, you'll need to set up the servers explicitly. | Yes, Spring Boot comes with built-in HTTP servers like Tomcat and Jetty. |
| How is it configured? | The Spring framework provides flexibility, but its configuration has to be built manually. | Spring Boot configures Spring and other third-party frameworks automatically by the default "convention over configuration" principle. |
| Do I need to know how to work with XML? | In Spring, knowledge of XML configuration is required. | Spring Boot does not require XML configuration. |
| Are there CLI tools for dev/testing apps? | The Spring framework alone doesn't provide CLI tools for developing or testing apps. | As a Spring module, Spring Boot has a CLI tool for developing and testing Spring-based apps. |