|  |
| --- |
| ***What is Spring Boot ?*** |
| *Spring Boot is approach to develop Spring based application with minimal configuration.* |
| *By using Spring boot we can develop web-applications, distributed applications, standalone apps, microservice based apps, etc. with auto-config.* |
| *Spring boot is very intelligent it will identify configuration required for our application and it will provide auto-config.* |

***Advantages of Spring Boot:-***

1. *Starter poms.*
2. *Dependancy version management.*
3. *Embedded Tomcat.*
4. *Actuator.*
5. *Security.*
6. *Auto-Configuration.*

***Version-wise features & Advantages?***

***What are transitive dependency in Spring Boot? 🡪All child dependencies re transitive dependencies.***

***How can we increase performance of our Spring Boot applications?***

*Removing unused / unnecessary dependencies of our application.*

***✅ What is Spring Boot?***

*🟩* ***Definition****:****Spring Boot*** *is an* ***opinionated framework*** *built on top of the Spring Framework that helps developers* ***create stand-alone, production-ready Spring applications*** *with minimal configuration.*

*It eliminates the need for writing boilerplate configuration by* ***auto-configuring*** *many components behind the scenes, and it comes with an* ***embedded server*** *(like Tomcat), so you don’t need to deploy WARs manually.*

***🟩 Why Spring Boot? (Need of It)***

* ***Traditional Spring applications*** *required heavy XML configuration and lots of setup.*
* *Managing* ***web.xml****,* ***DispatcherServlet****,* ***DataSource****, etc., was time-consuming.*
* *Every Spring module had to be manually configured and integrated (like Spring MVC + JPA + Security + AOP).*
* *Spring Boot solves this with:*
  + *✔️ Auto-Configuration*
  + *✔️ Opinionated Defaults*
  + *✔️ Embedded Web Server*
  + *✔️ Production-Ready Features*

*✅ “Spring Boot was created to simplify Spring application development and reduce developer effort.”*

***✅ Tricky / Indirect Interview Questions on Spring Boot***

*Interviewers may avoid directly saying “Spring Boot” to check your deeper understanding. Here are some tricky versions:*

| ***❓ Tricky Question*** | ***💡 What It's Actually Testing*** |
| --- | --- |
| *“How would you make a Spring app run without needing a server like Tomcat?”* | *Embedded server (Spring Boot)* |
| *“Can a Spring app start with just a main method?”* | *Spring Boot Application class* |
| *“How would you simplify Spring configuration for faster POC?”* | *Auto-Configuration* |
| *“What tool helps in rapid prototyping using Spring stack?”* | *Spring Boot* |
| *“How do you externalize configuration in modern Spring apps?”* | *application.properties or YAML* |
| *“How do you monitor a Spring-based microservice in production?”* | *Spring Boot Actuator* |

***🟩 Key Features of Spring Boot***

| ***Feature*** | ***Description*** |
| --- | --- |
| ***Auto Configuration*** | *Automatically configures Spring beans based on classpath dependencies.* |
| ***Embedded Server*** | *Comes with embedded Tomcat, Jetty, or Undertow.* |
| ***Starter Dependencies*** | *Provides pre-defined dependencies like spring-boot-starter-web.* |
| ***Spring Boot CLI*** | *Command-line tool to run and test Spring apps quickly.* |
| ***Actuator*** | *Provides production-ready endpoints to monitor and manage the app.* |
| ***No web.xml*** | *Servlet is configured automatically.* |
| ***Opinionated Defaults*** | *Offers sensible defaults to reduce setup time.* |

***🟩 Advantages of Spring Boot***

*✅* ***Developer Productivity*** *– Write less code, faster development.  
✅* ***Standalone Applications*** *– No need for external servers or WARs.  
✅* ***Auto Configuration*** *– Reduces manual setup.  
✅* ***Embedded Servers*** *– No need to deploy on Tomcat manually.  
✅* ***Actuator Support*** *– Built-in monitoring and metrics.  
✅* ***Easy Integration with Spring Cloud*** *– Perfect for microservices.  
✅* ***Spring Boot DevTools*** *– Enables hot reloading and productivity tools.  
✅* ***Starter Projects*** *– Reduce dependency management overhead.*

***🔴 Disadvantages of Spring Boot***

*⚠️* ***Hidden Configurations*** *– Auto-configuration may confuse beginners.  
⚠️* ***Memory Footprint*** *– Embedded servers can increase size and RAM usage.  
⚠️* ***Limited Fine-Grained Control*** *– Sometimes you need to override too much to customize things.  
⚠️* ***Overkill for Simple Apps*** *– Adds more than needed if project is very small.  
⚠️* ***Learning Curve with Actuator/Security*** *– Managing actuator endpoints and customizing security settings can be tricky.*

***🟩 Real-Time Use Case***

*🎯* ***Banking Web Portal*** *Using Spring Boot, you can create a* ***customer portal*** *that:*

* *Accepts new account requests (Spring Boot + REST)*
* *Stores user data (Spring Data JPA + MySQL)*
* *Sends emails for account creation (JavaMailSender)*
* *Tracks logs and performance (Actuator)*
* *Allows admin management via endpoints*

*@SpringBootApplication*

*public class BankingApp {*

*public static void main(String[] args) {*

*SpringApplication.run(BankingApp.class, args);*

*}*

*}*

*✅ One file, one main method, and it runs the entire application!*

***🟩 Summary for Interviews***

| ***🔑 Key Point*** | ***💬 Interview Line*** |
| --- | --- |
| *What* | *A framework to build Spring apps quickly with minimal config* |
| *Embedded Server* | *Runs without deploying to external server* |
| *Autoconfiguration* | *Beans and settings are auto-detected and initialized* |
| *Use case* | *Microservices, REST APIs, POCs, Enterprise apps* |
| *Popular Tools* | *spring-boot-starter, actuator, devtools* |

***✅ Q10. What is the difference between Spring Framework and Spring Boot?***

*🟩* ***Tricky version asked in interviews:***

* *“How do you simplify traditional Spring applications for faster development?”*
* *“Can Spring run without a deployment descriptor now?”*
* *“Which part of Spring helps you avoid writing boilerplate configuration?”*
* *“What’s the modern approach to setting up a Spring REST service quickly?”*

***🟩 Comparative Table: Spring Framework vs Spring Boot***

| ***🔸 Feature*** | ***🌿 Spring Framework*** | ***🚀 Spring Boot*** |
| --- | --- | --- |
| ***Definition*** | *A comprehensive dependency injection and aspect-oriented programming framework.* | *A rapid application development framework built on top of Spring.* |
| ***Configuration*** | *Heavy XML or Java-based configuration.* | *Auto-configuration with minimal setup.* |
| ***Boilerplate Code*** | *Requires a lot (e.g., web.xml, dispatcher-servlet.xml).* | *Almost zero boilerplate.* |
| ***Server Deployment*** | *Requires WAR file and external server (e.g., Tomcat).* | *Runs as standalone with* ***embedded servers*** *(Tomcat/Jetty).* |
| ***Main Class*** | *No single entry point (main() not used traditionally).* | *Has a main method using SpringApplication.run() to start app.* |
| ***Project Setup*** | *Manually manage dependencies using Maven/Gradle.* | *Uses* ***Spring Boot Starters*** *to manage dependencies easily.* |
| ***Monitoring/Management*** | *No built-in health checks or metrics.* | *Comes with* ***Spring Boot Actuator*** *for health, metrics, and monitoring.* |
| ***Learning Curve*** | *Steeper due to manual setup.* | *Easier for beginners and quicker for experts.* |
| ***Microservices Support*** | *Requires manual integration.* | *Designed to work seamlessly with* ***Spring Cloud****.* |
| ***DevTools*** | *Not included.* | *Includes hot reload, automatic restart via* ***Spring Boot DevTools****.* |
| ***Command Line Interface (CLI)*** | *Not available.* | *Comes with CLI to run Groovy-based Spring scripts.* |

***🟩 Summary in Simple Terms***

| ***Aspect*** | ***Spring*** | ***Spring Boot*** |
| --- | --- | --- |
| *Setup* | *Manual & verbose* | *Auto-configured* |
| *Server* | *External required* | *Embedded* |
| *Dev Time* | *Slower* | *Rapid* |
| *Monitoring* | *Manual* | *Built-in (Actuator)* |
| *Usage* | *Suitable for fine-grained enterprise control* | *Best for microservices, POCs, REST APIs* |

***🟩 Real-Time Use Case View***

| ***📌 Use Case*** | ***Prefer*** |
| --- | --- |
| *Legacy enterprise app with custom control* | *Spring* |
| *Modern REST API or microservice* | *Spring Boot* |
| *Lightweight app with full customization* | *Spring* |
| *Rapid prototyping, cloud deployment* | *Spring Boot* |

***✅ Interview Tip:***

*“Spring Boot* ***does not replace*** *Spring Framework. It* ***simplifies using it*** *by handling internal configurations so developers can focus on writing business logic instead of boilerplate setup.”*