# Spring Boot Interview Questions

## 🔹 Basic Level

1. 1. What is Spring Boot?

Spring Boot is a framework that simplifies the development of Spring applications by providing:  
- Auto-configuration  
- Embedded servers (like Tomcat)  
- Opinionated starter dependencies  
- Production-ready features (e.g., health checks, metrics)

1. 2. What are the advantages of Spring Boot?

- Reduces boilerplate code  
- No need for external server deployment  
- Auto-configuration based on classpath  
- Starter templates for dependencies  
- Production-ready features (Actuator)

1. 3. What are Starters in Spring Boot?

Starters are pre-defined dependencies that simplify build configuration. Example:  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>

1. 4. How does Spring Boot auto-configuration work?

Uses @EnableAutoConfiguration and classpath scanning to automatically configure beans.

1. 5. How do you create a Spring Boot application?

Use Spring Initializr and annotate the main class with @SpringBootApplication:  
@SpringBootApplication  
public class App {  
 public static void main(String[] args) {  
 SpringApplication.run(App.class, args);  
 }  
}

1. 6. What is @SpringBootApplication?

A convenience annotation that includes:  
- @Configuration  
- @EnableAutoConfiguration  
- @ComponentScan

## 🔹 Intermediate Level

1. 7. What is Spring Boot Actuator?

Provides production-ready features like health checks, metrics, and environment info at endpoints like /actuator/health.

1. 8. How do you handle exceptions in Spring Boot?

Using @ControllerAdvice and @ExceptionHandler:  
@ControllerAdvice  
public class GlobalExceptionHandler {  
 @ExceptionHandler(Exception.class)  
 public ResponseEntity<String> handleException(Exception e) {  
 return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body(e.getMessage());  
 }  
}

1. 9. How to connect Spring Boot with a database?

Via application.properties:  
spring.datasource.url=jdbc:mysql://localhost:3306/db  
spring.datasource.username=root  
spring.datasource.password=pass  
spring.jpa.hibernate.ddl-auto=update

1. 10. What is the difference between @Component, @Service, @Repository, and @Controller?

@Component: Generic bean  
@Service: Business logic layer  
@Repository: Data access layer  
@Controller: Web controller (MVC)

1. 11. What are Profiles in Spring Boot?

Used to define environment-specific configurations using @Profile.

1. 12. How to configure properties in Spring Boot?

Use application.properties or application.yml and bind them using @Value or @ConfigurationProperties.

1. 13. How do you implement logging in Spring Boot?

Spring Boot uses Logback by default. Configure it via application.properties:  
logging.level.root=INFO  
logging.level.org.springframework=DEBUG

1. 14. How do you test a Spring Boot application?

Use @SpringBootTest for integration tests and @WebMvcTest, @DataJpaTest for slice tests.

## 🔹 Advanced Level

1. 15. What is Spring Boot DevTools?

Provides live reload, automatic restart, and better developer experience.

1. 16. Explain the Spring Boot Starter Parent.

It provides default configurations for Maven such as dependency versions, Java version, and plugin settings.

1. 17. What is the use of CommandLineRunner and ApplicationRunner?

Used to run code after the application context is loaded:  
@Bean  
public CommandLineRunner run() {  
 return args -> System.out.println("Application started");  
}

1. 18. How does Spring Boot support microservices?

Via Spring MVC for REST APIs, Spring Cloud Config for configuration, Eureka for discovery, Resilience4j, and Spring Cloud Gateway.

1. 19. What is the difference between Spring Boot and Spring MVC?

Spring Boot simplifies configuration and includes embedded servers, while Spring MVC is the web framework part.

1. 20. What is the use of @EnableAutoConfiguration?

It enables Spring Boot’s auto-configuration based on the classpath.

## 🔹 Bonus: Microservice-Specific Spring Boot Questions

1. 21. How do you create a REST API in Spring Boot?

@RestController  
@RequestMapping("/api")  
public class UserController {  
 @GetMapping("/users")  
 public List<User> getUsers() {  
 return userService.getAllUsers();  
 }  
}

1. 22. How do you implement security in Spring Boot?

Use Spring Security and configure with SecurityFilterChain or WebSecurityConfigurerAdapter (deprecated in newer versions).

1. 23. What is the difference between @RestController and @Controller?

@RestController = @Controller + @ResponseBody  
Used for REST APIs where the return is JSON/XML.

1. 24. What are common annotations used in Spring Boot?

@SpringBootApplication, @RestController, @RequestMapping, @GetMapping/@PostMapping,  
@Service, @Repository, @Autowired, @Entity, @Id,  
@EnableJpaRepositories, @ComponentScan, @Configuration