SPRING BOOT – COMPLETE REVISION SHEET

1. Spring Boot Basics

- @SpringBootApplication = @Configuration + @EnableAutoConfiguration + @ComponentScan

- Auto-configuration: detects classpath and configures beans automatically.

- Starter dependencies simplify setup (e.g., starter-web, starter-data-jpa).

2. Dependency Injection (DI) & IOC

- @Component, @Service, @Repository used for stereotype annotations.

- @Autowired for injection (prefer constructor injection).

- @Qualifier resolves multiple bean conflicts.

- @Bean used inside configuration classes.

3. REST API Development

- @RestController vs @Controller

- @GetMapping, @PostMapping, @PutMapping, @DeleteMapping

- @PathVariable, @RequestParam, @RequestBody

- ResponseEntity for custom HTTP responses.

4. Validation (JSR 380)

- Use @Valid on request body.

- Common annotations: @NotNull, @Size, @Email, @Min.

- Bind errors using BindingResult or global handler.

5. Exception Handling

- @ControllerAdvice for global exceptions.

- @ExceptionHandler for specific exceptions.

- Custom JSON error response structure.

6. Spring Data JPA

- @Entity, @Id, @GeneratedValue

- JpaRepository<>, CrudRepository<>

- Derived query methods (findByName, findByEmail)
- JPQL (@Query), Native queries
- Pagination (Pageable) & Sorting

7. JPA & Hibernate Concepts

- Lazy vs Eager fetching
- Cascade Types: ALL, PERSIST, MERGE, REMOVE
- Relationships: @OneToOne, @OneToMany, @ManyToOne, @ManyToMany
- Entity Lifecycle: transient, persistent, detached, removed

8. Configuration & Profiles

- application.properties vs application.yml
- Profiles: application-dev.properties, application-prod.properties
- Activate profile using spring.profiles.active

9. Spring Boot Actuator

- Enables health, metrics, env, beans endpoints.
- /actuator/health, /actuator/info
- Can extend with custom endpoints.

10. Logging

- Uses SLF4J + Logback
- Log levels: TRACE, DEBUG, INFO, WARN, ERROR
- Custom logback-spring.xml for advanced config.

11. Spring Security Basics

- Authentication vs Authorization
- Filters & SecurityFilterChain (new config style)
- Role-based access: hasRole('ADMIN')
- JWT basic flow (Auth → Token → Validate token)

12. Filters & Interceptors

- Filters → servlet level

- Interceptors → Spring MVC

- Used for logging, auditing, pre/post processing

13. Caching

- @EnableCaching

- @Cacheable, @CacheEvict, @CachePut

- Supports Redis, ConcurrentMap, EhCache

14. Swagger / OpenAPI

- springdoc-openapi dependency

- Access API docs at /swagger-ui.html or /swagger-ui/index.html

- Use @Operation and @Schema for documentation

15. Spring Boot Application Flow

- SpringApplication.run() creates ApplicationContext

- Bean creation → Dependency injection → Auto-configuration → App startup

16. Testing

- @SpringBootTest for full integration test

- @WebMvcTest for controller tests

- MockMvc, TestRestTemplate

17. Microservices (Spring Cloud)

- Eureka for service discovery

- Spring Cloud Gateway for routing

- Config Server for centralized config

- Resilience4j for circuit breakers

18. Packaging & Deployment

- Build JAR: mvn clean package

- Run: java -jar app.jar

- Dockerizing Spring Boot using Dockerfile

19. Best Practices

- Use DTOs, not entities, in controllers

- Keep service layer for business logic

- Use ResponseEntity everywhere

- Use exception handler for clean error responses

- Prefer constructor injection