✅ **Spring Boot Must-Know Concepts (6+ Years Experience)**

(Refined, Structured, and Complete Roadmap)

---

1. Spring & Spring Boot Core

- **IoC (Inversion of Control)**
- **DI (Dependency Injection)**
- **Bean lifecycle & scopes (singleton, prototype, request, session)**
- **Autowiring (@Autowired, @Qualifier, constructor injection)**
- **Auto-configuration (how Spring Boot detects & configures beans)**
- **Spring Boot Starters (what happens internally, exclusions)**
- **Configuration Properties (@ConfigurationProperties, @Value)**
- **Profiles (dev/test/prod switching)**
- **Application Events & Listeners**

---

2. Spring Layers & Annotations

- **@Component, @Service, @Repository, @Controller, @RestController**
- **Bean creation: @Bean / @Configuration**
- **@EnableAutoConfiguration / @SpringBootApplication**
- **@EnableJpaRepositories, @EntityScan**
- **Validation**
    - @Valid
    - @Validated
    - Custom validators
- **@Transactional**
    - Propagation
    - Isolation
    - Read-only transactions
    - Rollback rules

---

3. Persistence (Spring Data JPA & Hibernate)

- **Entity mapping**
    - @Entity, @Table
    - @Id, @GeneratedValue

    - @OneToMany, @ManyToOne, @ManyToMany
    - Cascade types
    - Fetch types (Lazy/Eager)
- **JPQL, Native Queries**
- **Derived query methods**
- **Pagination & Sorting**
    - PageRequest
- **N+1 Problem (and solutions)**
    - Fetch join
    - @EntityGraph
    - BatchSize
- **JDBC Template**
- **Transaction management**
- **Optimistic/Pessimistic locking**

---

4. Hibernate Advanced Internals (Must Know for 6+ Years)

- **Session vs EntityManager**
- **Hibernate Cache Levels**
    - Level 1 Cache
    - Level 2 Cache
    - Query Cache
- **Dirty checking**
- **Flushing & Persistence Context**

---

5. Caching

- **Spring Cache abstraction**
    - @Cacheable
    - @CacheEvict
    - @CachePut
- Integrations:
    - Redis
    - caffeine
- Cache invalidation strategies

---

6. AOP (Aspect-Oriented Programming)

- **Cross Cutting Concerns**
- **@Aspect, @Around, @Before, @AfterReturning**
- **Pointcuts**
- **Use cases**
    - Logging
    - Exception handling
    - Security checks
    - Monitoring

---

7. API Development (Best Practices)

- Request/response DTO patterns
- DTO <-> Entity mapping (MapStruct, ModelMapper)
- Global Exception Handling
    - @ControllerAdvice
    - @ExceptionHandler
- Input validation
- REST best practices
- HTTP caching (ETags, cache headers)

---

8. Spring Security

- **Authentication vs Authorization**
- **SecurityFilterChain (new approach)**
- **UserDetailsService**
- **PasswordEncoder**
- **CSRF**
- **CORS**
- **JWT Authentication**
- **OAuth2 Basics**
    - OAuth2 Authorization Code flow
    - Resource server
    - Authorization server

- **Method-level security**
    - @PreAuthorize, @PostAuthorize

---

9. Messaging

Understand at least one:

- **Kafka**
    - Producers/Consumers
    - Consumer groups
    - Offsets
- **RabbitMQ**
    - Queues, exchanges, routing keys
- **Spring Cloud Stream**

(6-year engineers usually face Kafka questions)

---

10. Spring Cloud / Microservices Essentials

Even if not fully used, expected to know:

- **Service Discovery (Eureka/Consul)**
- **Config Server**
- **API Gateway (Spring Cloud Gateway)**
- **Resilience4j**
    - Circuit Breaker
    - Retry
    - Rate Limiter
- **OpenFeign**

---

11. Testing

- Unit testing
    - Mockito
    - JUnit5
- Integration testing
    - @SpringBootTest
    - @DataJpaTest
- Testcontainers (must know nowadays)

---

12. DevOps Awareness

(Not deep knowledge, but expected working exposure)

- Dockerfile, Docker Compose
- Kubernetes basics
  - Deployment
  - Service
  - ConfigMap
  - Secret
- CI/CD (Jenkins / GitHub Actions)
- Observability basics (must know)
  - Prometheus
  - Grafana
  - Logs (ELK / OpenSearch)

---

13. Performance & Optimization

- Database indexing basics
- Hibernate batch operations
- Connection pooling (HikariCP tuning)
- Memory leak patterns
- Profiling tools
  - JProfiler
  - Spring Boot Actuator metrics

---

14. Architectural Skills (6+ Years Must-Have)

- Clean architecture
- Hexagonal architecture
- SOLID principles
- Design patterns
  - Strategy
  - Factory
  - Builder
  - Template

    - Observer
- API versioning
- Monolith vs Microservices
- Event-driven architecture basics

---

15. Bonus: Real-World Integrations

Companies expect familiarity with:

- Swagger/OpenAPI
- Multipart uploads
- Scheduling
    - @Scheduled
    - Async processing
- Email, SMS integrations
- External API call handling
    - RestTemplate
    - WebClient (reactive)