**Backend Java Developer Roadmap: Topic-by-Topic Breakdown**

Below is a **comprehensive list of essential topics** organized in the order that you should cover them on your Backend Java Developer journey (Java + Spring Boot). Each major area is broken down into core topics, providing you with a clear step-by-step path before we translate this into a week-by-week schedule.

**1. Java Fundamentals**

* Basic Syntax, Variables, Data Types, Operators
* Conditionals (If, Switch)
* Loops (For, While, Do-While)
* Arrays and String Operations
* Methods, Method Overloading & Overriding
* Type Casting
* Lifecycle of a Program
* Packages and Modules

**2. Object-Oriented Programming (OOP)**

* Classes & Objects, Attributes, Methods
* Access Specifiers (public, private, protected)
* Static Keyword, Nested Classes
* Inheritance, Polymorphism
* Encapsulation, Abstraction
* Interfaces, Abstract Classes, Enums
* Object Lifecycle
* Method Chaining
* Record Classes
* Pass by Value / Reference

**3. Advanced Java**

* Collections (List, Set, Map, Queue, Stack, Dequeue, Array vs ArrayList)
* Generics
* Lambda Expressions & Functional Interfaces
* Stream API & Functional Composition
* Exception Handling
* Annotations
* Concurrency (Threads, Virtual Threads, Synchronization, Java Memory Model, Volatile)
* Date and Time API
* File & I/O Operations
* Networking & Regular Expressions
* Logging Frameworks (Logback, Log4j2, SLF4J, TinyLog)

**4. Build Tools, Version Control, IDE**

* Maven, Gradle, Bazel
* Dependency Management
* Project Structure
* Git Basics, Branching, Pull Requests
* Repo Hosting (GitHub, GitLab, Bitbucket)
* IntelliJ IDEA or Eclipse Setup

**5. SQL & Databases**

* Introduction to Relational DBs (PostgreSQL, MySQL)
* SQL Syntax: SELECT, INSERT, UPDATE, DELETE
* WHERE, ORDER BY, GROUP BY, HAVING
* JOINs (INNER, LEFT, RIGHT, FULL OUTER, SELF, CROSS)
* Aggregate Functions (SUM, COUNT, AVG, MIN, MAX), Window Functions
* Indexes, Query Optimization
* Data Constraints (PK, FK, Unique, Not Null, Check)
* Transactions, ACID Properties, Isolation Levels
* DDL, DML, Views
* Subqueries (Scalar, Column, Row, Table, Nested, Correlated)
* Stored Procedures, Functions
* NoSQL Basics (MongoDB, Redis, Neo4j), CAP Theorem

**6. ORM & Data Access**

* JDBC Basics
* JPA Concepts
* Hibernate Fundamentals
* Mapping Entities & Relationships
* Entity Lifecycle
* Lazy vs Eager Loading
* Repository & Specification Pattern
* Transaction Management
* Migrations (Flyway, Liquibase)
* Spring Data JPA
* Spring Data MongoDB/JDBC

**7. Spring Core & Spring Boot**

* Introduction to Spring
* Terminology, Architecture, Why Spring
* Dependency Injection, IoC
* Bean Lifecycle & Scopes
* Configuration: XML, Java, Annotations
* Spring Boot Starters & Auto-configuration
* Embedded Servers (Tomcat/Jetty)
* Spring Boot Actuator
* Application Configuration (properties/yaml)
* Profiles & Logging

**8. Web Development with Spring MVC**

* MVC Architecture
* Controllers, Request Mapping
* REST API Development (Design, Implementation)
* Request/Response Handling
* Exception Handling
* JSON/XML Serialization
* HTTP Status Codes
* Content Negotiation
* JSP, Servlet Fundamentals

**9. Security & Authentication**

* Authentication Fundamentals (Basic, JWT, OAuth2, Cookie, Token, SAML, OpenID)
* Authorization: Roles & Permissions
* Spring Security Configuration
* Secure Password Handling
* CORS, CSRF, HTTPS, SSL/TLS
* OWASP Risks & Security Best Practices
* API Key Management, OAuth2 Flows

**10. Testing**

* Unit Testing (JUnit, TestNG)
* Mocking (Mockito, @MockBean, Stubs, Spies)
* Integration Testing (@SpringBootTest, TestContainers, Database Testing)
* MockMVC for Controller Tests
* REST Assured, API Testing
* Cucumber-JVM, Behavior & Load Testing (JMeter)
* Code Coverage, Test Automation

**11. Microservices & System Design**

* Monolithic vs Microservices
* Principles of Microservices
* Service Discovery (Eureka, Consul)
* API Gateway (Spring Cloud Gateway)
* Centralized Configuration (Spring Cloud Config)
* Circuit Breaker Patterns (Hystrix, Resilience4j)
* Messaging (RabbitMQ, Kafka)
* Distributed Tracing (Sleuth, Zipkin)
* Domain-Driven Design
* Event Sourcing, CQRS
* Twelve-Factor Apps
* Caching (Redis, Memcached)
* Load Balancing, Scaling, Sharding, Replication
* Graceful Degradation, Throttling, Backpressure

**12. DevOps & Deployment**

* Docker, Docker Compose
* Kubernetes Basics
* CI/CD Pipelines (Jenkins, GitHub Actions)
* Automated Deployment
* Monitoring & Observability (Actuator, Prometheus, Grafana)
* Logging & Alerting
* Infrastructure as Code (Helm, Terraform basics)

**13. Advanced Topics & Specialization**

* JVM Tuning & Internals
* Performance Optimization
* Real-time Data: WebSockets, Server Sent Events
* Time-Series DBs (InfluxDB, Cassandra)
* Search Engines (Elasticsearch, Solr)
* Graph Databases (Neo4j)
* System Design Interviews
* Big-O Analysis & DSA in Coding Interviews
* Project Portfolio & GitHub Documentation

**Next Steps**

**Want a Week-by-Week Schedule for all these topics?**

Just confirm, and you’ll get:

* Each week mapped to specific topics and sub-skills
* Clear learning objectives and suggested milestones
* Project and practice goals

**This order ensures you follow a logical progression from the very basics through to advanced, specialized backend engineering topics.**

Let me know if you want the week-by-week curriculum breakdown (recommended)!