Course duration

- 4 days

Course Benefits

- Create Spring Boot projects.
- Use databases and JPA in Spring Boot.
- Create RESTful services with Spring Boot.
- Deploy services that use Netflix Eureka, Hystrix, and Ribbon to create resilient and scalable services.

Available Delivery Methods

Public Class
Public expert-led online training from the convenience of your home, office or anywhere with an internet connection. Guaranteed to run .

Private Class
Private classes are delivered for groups at your offices or a location of your choice.

Course Outline

1. Introduction to the Spring Framework
    1. What is the Spring Framework?
    2. Spring Philosophies
    3. Why Spring?
    4. Spring Modules
    5. Requirements and Supported Environments
    6. Using Spring with Servers
    7. Role of Spring Container
    8. Spring Example
    9. Avoiding Dependency on Spring
    10. Additional Spring Projects/Frameworks
    11. Summary
2. Spring Annotation Configuration
    1. Spring Containers
    2. Annotation-based Spring Bean Definition
    3. Scanning for Annotation Components

2. MongoDB Features
3. MongoDB's Logo
4. Positioning of MongoDB
5. MongoDB Applications
6. MongoDB Data Model
7. MongoDB Limitations
8. MongoDB Use Cases
9. MongoDB Query Language (QL)
10. The CRUD Operations
11. The
12. find
13. Method
14. The
15. findOne
16. Method
17. A MongoDB QL Example
18. Data Inserts
19. MongoDB vs Apache CouchDB
20. Summary

9. Working with Data in MongoDB
    1. Reading Data in MongoDB
    2. The Query Interface
    3. Query Syntax is Driver-Specific
    4. Projections
    5. Query and Projection Operators
    6. MongoDB Query to SQL Select Comparison
    7. Cursors
    8. Cursor Expiration
    9. Writing Data in MongoDB
    10. An Insert Operation Example
    11. The Update Operation
    12. An Update Operation Example
    13. A Remove Operation Example
    14. Limiting Return Data
    15. Data Sorting
    16. Aggregating Data
    17. Aggregation Stages
    18. Accumulators
    19. An Example of an Aggregation Pipe-line
    20. Map-Reduce
    21. Summary
10. Spring Data with MongoDB
    1. Why MongoDB?
    2. MongoDB in Spring Boot
    3. Pom.xml
    4. Application Properties
    5. MongoRepository

9. Eureka Server Configuration
10. Eureka Client/Service
11. Eureka Client Properties
12. Spring Cloud DiscoveryClient Interface
13. ServiceInstance JSON
14. ServiceInstance Interface
15. What about Services
16. Eureka and the AWS Ecosystem
17. Summary

17. Load-Balancing with Netflix Ribbon
    1. Load Balancing in Microservices
    2. Netflix Ribbon
    3. Server-side load balance
    4. Client-side Load Balance
    5. Architecture
    6. Load Balance Rules
    7. RoundRobinRule
    8. AvailabilityFilteringRule
    9. WeightedResponseTimeRule
    10. RandomRule
    11. ZoneAvoidanceRule
    12. IPing Interface (Failover)
    13. Using Ribbon
    14. YAML Configuration
    15. Configuration Class
    16. Client Class
    17. Client Class Implementation
    18. Integration with Eureka (Service Discovery)
    19. Using Ribbon in the Amazon AWS Cloud
    20. Summary
18. Application Hardening with Netflix Hystrix
    1. Netflix Hystrix
    2. Design Principles
    3. Design Principles (continued)
    4. Cascading Failures
    5. Bulkhead Pattern
    6. Circuit Breaker Pattern
    7. Thread Pooling
    8. Request Caching
    9. Request Collapsing
    10. Fail-Fast
    11. Fallback
    12. Using Hystrix
    13. Circuit Breaker Configuration
    14. Fallback Configuration
    15. Collapser Configuration
    16. Rest Controller and Handler

17. Collapser Service (Part 1)
18. How the Collapser Works
19. Hystrix Monitor
20. Enable Monitoring
21. Turbine
22. The Monitor
23. Monitor details
24. Summary

19. Edge Components with Netflix Zuul
    1. Zuul is the Gatekeeper
    2. Request Handling
    3. Filters
    4. Filter Architecture
    5. Filter Properties
    6. filterType()
    7. filterOrder()
    8. shouldFilter()
    9. Run()
    10. Cancel Request
    11. Dynamic Filter Loading
    12. Filter Communications
    13. Routing with Eureka and Ribbon
    14. Summary
20. Distributed Tracing with Zipkin
    1. Zipkin
    2. Zipkin Features
    3. Architecture
    4. The Collector
    5. Storage
    6. API
    7. GUI Console
    8. Zipkin Console Homepage
    9. View a Trace
    10. Trace Details
    11. Dependencies
    12. Dependency Details
    13. Zipkin in Spring Boot
    14. Zipkin Configuration
    15. Summary

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Microservices class:

- Experience with Java development.