Microservices Development using Spring Boot

Objective : To help participants from Java JEE server-side programming background to learn and develop their skills in microservices.

Entry Criteria : Participants meeting the required pre-requisites for each of the module.

Exit Criteria : Participants successfully completing the graded case studies / assessments for each of the modules.

Learning Plan and Duration :

#	Learning Plan	Duration in Hrs
1	Spring	16 Hrs
2	Spring Boot	16 Hrs
3	REST API Development using Spring Boot	24 Hrs
4	Microservices Development using Spring Boot	32 Hrs

Module 1.Spring

Duration : 2 Days – 16 Hrs

Pre-requisites: Participants should have a good hands on experience in Java JEE server side programming.

Course Outline

Introduction to Spring
- The Spring application context
- XML configuration
- Working with existing singletons and factories
- Working with multiple configuration files

Understanding the Bean Life Cycle
- XML namespaces
- Initialization, use, and destruction phases
- Bean Post Processor
- Externalizing constant values into properties files
- Bean scopes

Simplifying Application Configuration
- Bean definition inheritance
- Inner beans
- p and util namespaces
- Dependency injection of collections
- Spring Expression Language (SpEL)

Annotation-Based Dependency Injection

- Autowiring and component scanning
- Stereotype annotations
- Java-based configuration
- Mixing configuration styles
- When to use XML, annotations, and Java configuration

Aspect-Oriented Programming

- What problems does AOP solve?
- Differences between Spring AOP and AspectJ
- Defining pointcut expressions
- How to use the types of advice: around, before, after

Integrating Spring with Hibernate

- Quick introduction to ORM with Hibernate
- Benefits of using Spring with Hibernate
- Hibernate configuration in Spring
- Exception handling

Spring in a Web Application

- Introduction to Spring MVC
- Defining Spring MVC controllers using annotations
- Spring MVC in the view layer
- MVC namespace

Module 2.SpringBoot

Duration : 2 Days – 16 Hrs

Pre-requisites: Participants should have a good hands on experience in Spring.

Introduction

Spring Configuration Overview

- Spring Review - XML and @Component
- Dependency Injection
- Java-based Configuration (@Configuration)
- Integrating Configuration of Different Types

Introduction to Spring Boot

- Brief maven Overview
- Intro to Spring Boot - What is Spring Boot and What It Does
- Spring Boot Hello World / SpringApplication

Configuration and Customization

- Working with Properties - YAML and .properties
- Logging and its Configuration
- Auto-configuration Overview

Spring Boot Database Support

- Basic Auto-configuration - Datasource and Pooling
- Configuration Properties
- Spring Boot's JPA Support - spring-boot-starter-data-jpa
- Spring Boot Data (with Data-JPA in Detail)
- Using Spring Boot Data - CrudRepository/JpaRepository
- Defining Queries with Spring Boot Data

Spring Boot Web/REST

- DispatcherServlet Review
- Web Starters and Configuration spring-boot-starter-web
- Using Embedded Servers (Tomcat, Netty)
- Deploying to an External Server
- Actuator Overview and Capabilities
- Actuator Endpoints

Spring SECURITY

- Auto-configuration/simple security
- Security with JDBC/LDAP

Spring Boot Integration with Spring Batch

- Spring Batch Configuration
- Start job,Execute task,Finish job

Spring Boot Integration with Apache Kafka

- Even-driven programming
- Apache Kafka Architecture
- Spring Boot with Apache Kafka

Module 3.REST API Development Using Spring Boot

Duration : 3 Days – 24 Hrs

Pre-requisites: Participants should have a good hands on experience in Spring and Spring Boot

REST API Design - Concepts

- Introduction to REST
- REST and HTTP
- Designing Resource URI's
- RESTful URI types
- HTTP Methods
- Idempotence in HTTP Methods
- Rest Resource
- HATEOAS
- The Richardson Maturity Model
- Jax-RS vs Spring MVC Rest

Spring Boot RESTful Introduction

- Spring Restful Overview
- Initializing a RESTful Services Project with Spring Boot
- Creating a Hello World Service
- Enhancing the Hello World Service to return a Bean

Restful Web Services with Spring Boot

- Enhancing the Hello World Service with a Path Variable
- Creating User Bean and User Service
- Implementing GET Methods for User Resource
- Implementing POST Method to create User Resource
- Enhancing POST Method to return correct HTTP Status Code and Location
- Implementing DELETE Method to delete a User Resource

Restful Web Services with Exception Handling, Internationalization & Content Negotiation

- Implementing Exception Handling - 404 Resource Not Found
- Implementing Generic Exception Handling for all Resources
- Internationalization & Content Negotiation

RESTful development with HATEOAS, Swagger, Filtering & Versioning

- Introduction to Swagger Documentation Format
- Enhancing Swagger Documentation with Custom Annotations
- Monitoring APIs with Spring Boot Actuator
- Implementing Static & Dynamic Filtering for RESTful Service

- Versioning RESTful Services - Basic Approach with URIs
- Versioning RESTful Services - Header and Content Negotiation Approach

Overview of Connecting RESTful Service using Spring Data
- ORM overview (Hibernate/JPA)
- Creating User Entity and some test data
- Updating GET, POST, DELETE methods on User Resource to use JPA
- Creating Post Entity and Many to One Relationship with User Entity

Module 4: Microservices Development using Spring Boot

Duration : 4 Days – 32 Hrs

Pre-requisites: Participants should have a good hands on experience in Spring, Spring Boot and REST API Development

RESTful API Overview
- Spring RESTful API complete example

SOA and Microservices
- SOA Overview
- Microservice Overview
- Spring Boot and Microservices
- Spring Cloud Netflix components

Microservices and Cloud
- Cloud Services : IaaS, PaaS, Saas
- Cloud Native Applications

RESTful service with Service Registry
- Introduction to Naming Server
- Setting up Eureka Naming Server
- Connecting a service to Eureka

RESTful service with Webclient & Resilience4J
- Introduction to reactive programming
- RestTemplate vs Webclient
- Prevent failure cascades using Resilience4J, Rate limiter, Retry & Bulkhead

Using “Twelve-Factor App” Style Configuration

- Understand “Twelve-Factor App” configuration
- Read external configuration
- Support multiple environments using Spring profiles
- Spring Cloud Configuration Server
- Refresh configuration without restarting your application

Securing the RESTful services

- Implementing Basic Authentication with Spring Security
- Single sign-on with OAuth2

Distributed logging, Monitoring and Tracking

- Introduction to ELK stack & Zipkin
- Collecting and analyzing logs using ELK stack
- Distributed tracing using Zipkin

RESTful service with Webclient

- Introduction to reactive programming
- Spring Webflux with Webclient

Domain Driven Design

- Strategic Patterns (Domain, Sub-domain, Bounded Context)
- Tactical Patterns (Entity, Value Object, Aggregate, Repositories)
- Using Spring Boot Data - CrudRepository/JpaRepository

Saga & CQRS Patterns

- How to implement distributed transactions?
- what are events and commands?
- When and how to apply?

Dockerize Rest API App

- Containerization
- Setup the Dockerfile, Build the docker image

Getting Started with Kubernetes

- Creating Kubernetes Cluster
- Kubernetes Concepts - Pods, Replica Sets and Deployment
- Understanding Services in Kubernetes
- Understanding Kubernetes Architecture - Master Node and Nodes
- Build Docker Image and Push to Docker Hub for Hello World Rest API
- Deploy Spring Boot Sample Rest API to Kubernetes