Course Outline

Microservices Development with Java and Spring Boot

Course Summary

Objectives

After taking this course, students will be able to:

- Use Spring Boot to build standalone microservices and RESTful services
- Secure the transport layer via HTTPS
- Implement asynchronous messaging
- Discuss Netflix OSS to implement patterns for service discovery, load balancing, fault tolerance, and other key concerns for scalable distributed systems
- Monitor microservices with Sleuth and Zipkin
- Filter requests to your microservices using Zuul
- Students will finally look at implementing microservices with kubernetes

Topics

- Demystifying Microservices
- Building Microservices with Spring Boot
- Microservices Applied
- Microservices Use Case
- Reviewing a Sample Use Case Implementation
- Autoscaling Microservices
- Logging and Monitoring
- Containerizing your Microservice
- Deploying your Microservice with Kubernetes

Duration

Four Days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.

Course Outline

Microservices Development with Java and Spring Boot

Course Outline

I. ***Demystifying Microservices***
- A. Microservices Progression and Architecture
- B. Characteristics of microservices
- C. Microservices benefits
- D. Microservices Relationship with SOA & Twelve Factor apps
- E. Microservices Uses

II. ***Building Microservices with Spring Boot***
- A. Spring Boot to Build Microservices
  - Lab : Installing Spring Tool Suite, Maven and Java
  - Lab Solution : Install STS and Java - Visual
  - Lab Solution : Maven Installation
- B. POM and HATEOAS
- C. Spring Boot Configuration
- D. Securing Microservices
- E. Enabling cross-origin access for Microservices

III. ***Microservices Applied***
- A. Challenges around Microservices
- B. Communication and Orchestration of Microservices
- C. BPM and Workflows with Microservices
- D. Service Design Endpoints
- E. Service Version

IV. ***Microservices Use Case***
- A. Understanding the Application
- B. Why Microservices?
- C. Business Case
- D. Key Questions to be Answered
- E. Monolithic to Microservices
- F. Integration with other systems

V. ***Reviewing a Sample Use Case Implementation***
- A. Spring Cloud
- B. Config Server
- C. Feign as a declarative REST client
- D. Eureka Discovery
- E. Zuul
- F. Streams for reactive microservices

VI. ***Autoscaling Microservices***
- A. Autoscaling Microservices
- B. Components

VII. ***Logging and Monitoring***
- A. Logging and Monitoring
- B. Turbine

VIII. ***Containerizing your Microservice***
- A. Containerizing your Microservice
- B. Docker

IX. ***Deploying your Microservice with Kubernetes***
- A. Deploying your Microservice with Kubernetes

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.