# Spring Boot Training Course Outline

# Day 1:

**Demystifying Microservices**

* The evolution of microservices
* What are microservices?
* Monolith benefits
* Principles of microservices
* Characteristics of microservices
* Challenges for a successful microservice architecture
* Microservices examples
* Microservices benefits
* Microservices vs SOA
* Relationship with other architecture styles
* Microservice use cases

**Spring Framework Spring 5.x Overview**

* Spring In Context – Core Concepts
* Bean management through IOC
* Bean Creation
* Constructor Injection
* Setter Injection

**Spring Boot Internals**

* Using @EnableAutoConfiguration, @ComponentScan, @Configuration
* Auto-Configuration
* Use of @SpringBootAppilcation annotation
* External Configuration
* Profiles and Logging
* Packaging Spring Application

# Day 2:

**Data Access with Spring Boot**

* Spring Data JPA & Rest
* Spring Data JPA: The Data Tier
* @Repository Annotation
* Introduction to JPA
* Adding Spring Data JPA
* Creating a Spring Data JPA Repository
* Making Crud Operations with Repository
* Adding Entity Relationship and Extending Repository

**Spring web applications and REST based web services program.**

* REST Application with Spring Boot
* @RestController Annotation
* Stereotypes Using with REST Controllers
* CRUD Application
* Using Java Bean Model with REST Controllers
* Consuming a REST Application with REST Template
* Spring boot environment variables
* Profile-specific Properties
* Consume Spring REST Service from Spring MVC
* Testing using Postman

**Spring Boot with HATEOAS & Swagger**

* Hypermedia
* HATEOAS
* Swagger documentation

# Day 3:

**Microservices Security:**

* Why security matters in a microservice environment
* How Microservices can be secured
* understand Spring Security
* OAUTH 2
* CQRS
* API Gateways

**Spring Boot actuator**

* Observability
* Troubleshooting

**Docker Overview**

* Understanding Containers and Docker
* The differences between dedicated hosts, virtual machines, and Docker
* Docker installation
* The Docker command-line client
* Docker and the container ecosystem
* Dockerfile
* Build container from spring tool

**Reactive Spring**

* Project Reactor
* Reactive in Spring Boot
* Reactive types - Flux and Mono
* Subscribing to Reactive Streams
* Transforming reactive sequences with operators
* Handling reactive sequences and Elements
* Purely functional web with WebFlux

# Softwires Required:

**Machines with 16GB Ram must**

JDK 11: <https://www.oracle.com/technetwork/java/javase/downloads/index.html> POSTMAN: <https://www.getpostman.com/downloads/>

Spring Tool Suite: <https://spring.io/tools> Docker for Desktop:

windows[:https://download.docker.com/win/stable/Docker%20Desktop%20Installer.exe](https://download.docker.com/win/stable/Docker%20Desktop%20Installer.exe) Mac OS: <https://download.docker.com/mac/stable/Docker.dmg>

Good Internet access [Mandatory as all dependencies are downloaded from Internet]

Websites Access Required: <https://hub.docker.com/> <https://repo1.maven.org/maven2/> <https://mvnrepository.com/repos/central>