### App Framework Specialization

### Advanced Spring & Microservices Program Document

**Duration : 4 Days**

**Timing : IST 08.30 am – 01.00 pm | 4 Hours per Day**

**Break Time : IST 10.30 am – 11.00 am**

**Training Mode : Online**

# Session Plan.

* Each day we will have 4 Hours of session which includes Lecture, demo & Important Labs
* Every day session would be handled as follows; this may vary based on day’s progress
  + **Session 1**: 90 Mins Lecture Demo followed by 20 Mins Lab and 10 Mins Q&A
  + **Session 2**: 90 Mins Lecture Demo followed by 20 Mins Lab and 10 Mins Q&A
* Post the session, All the participants will receive a Self-practiced hands-on task sheet for daily practice.
* During the session 2 Q & A, participant would be briefed about self-practiced tasks for the day
* On the day of starting first 15 Min will be spent on setting up objective, expectation and walk through of Essentials programs, followed by day’s session and from next day First 15 Min will be spent on clarification on previous day self-practiced tasks completion.
* Any issues during the session lab, will be clarified, but if any troubleshooting required due to Hardware or software issues will not be addressed as part of the program due time constraints.
* The will be MCQ based Assessment on the final day

Prerequisites

1. Sound Core Java Knowledge
2. Knowledge on SQL, Spring and JPA added advantage
3. Should have completed App framework essentials / Specialization training conducted earlier or equivalent experience
4. All participants must have laptop with good Internet
5. All required software loaded as given in the Lab setup requirement sheet – **Page 3**

**Day Wise Training plan**

**Day 1 (Spring core)**

* + - Spring Core summary
    - Overview of Spring Modules
    - IOC & Dependency Injection
    - Bean Life cycle
    - Spring AOP
    - Spring Test and best practices
    - Spring Boot
    - Getting Started with Spring boot
    - Spring boot starter Projects

**Hands-on**

First MVC application using Spring Boot

IOC & Dependency Injection Hands on

**Day 2 (Spring web, WS)**

* + - Spring web components
    - MVC Project using spring boot
    - Spring REST
    - RESTFUL web services using Spring Boot
    - Understanding exception handling in REST
    - Integrating with marshaller/unmarshaller (JSON/XML)

**Hands-on**

Create a view (jsp) to interact with Web controller

Create RESTFUL web services

Follow best practices discussed during the session.

**Day 3 (Spring database, JPA, data)**

* + - Spring JDBC
    - Spring transaction management
    - Spring JPA
    - Spring Data

**Hands -on**

Integrate the REST web services to use backend database to store data

Integrate with spring data and JPA to store and fetch data

**Day 4 (Microservices, other services)**

* + - Microservices
    - Using Eureka server, smart load Balancing
    - Logging with Sleuth
    - Monitoring with Zipkin
    - Filtering Requests using Zuul

**Hands-on**

Implementing load balancing

Using zuul for proxying

Tracing using sleuth and Zipkin

# Software Requirement for Lab Activities

1. JDK 11

<https://www.oracle.com/in/java/technologies/javase/javase-jdk11-downloads.html>

2. Spring Tool Suite 4.0 or above

<https://download.springsource.com/release/STS4/4.7.0.RELEASE/dist/e4.16/spring-tool-suite-4-4.7.0.RELEASE-e4.16.0-win32.win32.x86_64.self-extracting.jar>

<https://spring.io/tools>

3. MySQL RDBMS

4. Zipkin

Use this link   <https://zipkin.io/pages/quickstart> and navigate to the page

When the Page Opens  under java subheading choose **latest release**  link the executable jar downloads.

Once its downloaded open the command prompt and execute the jar file

java -jar filename.jar

Zipkin starts once its started open the browser the the url [http://localhost:9411/zipkin](http://localhost:9411/zipkin/)  should be able to see the Zipkin dashboard

5. Tomcat Server 8.0+

<https://tomcat.apache.org/download-80.cgi>

6. Maven

<https://mvnrepository.com/>

<https://maven.apache.org/download.cgi>

7. MongoDB

<https://www.mongodb.com/download-center/community>

8. Postman

[www.postman.com/download](http://www.postman.com/download)