



An ISO 9001 : 2008 Certified Company

NARESH[®]

technologies

SOFTWARE TRAINING & DEVELOPMENT

☎ 040-2374 6666 | 23734842

✉ info@nareshit.com

🌐 www.nareshit.com

📘 nareshit

📧 nareshitech

📺 nareshit

Find Latest IT Jobs
SeshaJobs.com

Opp. Satyam Theatre, Durga Bhavani Plaza, Ameerpet, Hyd-16

SPRING BOOT

& Cloud with MicroServices

Spring BOOT

A) Spring Boot Core:

Boot Basic and Features

- Need Of Spring Boot
- Spring V/s Boot
- Components in Boot
- Boot Architecture
- Types of Files in Boot
- Advantages of Boot
- Boot Dependencies
- Maven Boot Plug-in
- Parent Project in Boot

First Application in Boot

- Basic Annotations in Boot
- Execution Flow
- Boot Starter class
- Application Properties
- Server Port Change
- Managing of Dependencies
- Common Problem in coding
- @SpringBootApplication

Customizing the Banner

- Spring Boot Banner
- Banner Example
- Text Banner
- Custom banner

Runners in Spring Boot

- Command Line Runner
- Application Runner
- Multiple Runners with Order
- Runner Configuration
- Inputs to Runner

Application Properties file

- Properties file Need
- XML/Properties
- Finding Key-values
- Option-args/Non option-args
- Default values

Application YAML File

- YAML structure
- YAML Locations and Order
- Snake YAML Execution Flow

POM structure for Boot

- Why POM, Role of POM Elements
- Parent Project
- Properties in POM
- Version Management

Profiles in Spring boot

- @Profile annotation
- Customer/Environment Specific coding
- Profile Properties
- Profiles YAML

Building Application (.jar,.war)

- What is Build plug-in
- Maven Clean and Install
- target folder and .jar and .war

Spring Boot starter class

- @SpringBootConfiguration
- @EnableAutoConfiguration
- @ComponentScan

Spring Boot stand alone Application

- Writing Stand alone Application
- Service Examples
- Printing Messages

for Online Training Call / WhatsApp : 8179191999

Spring Boot Configuration

- Writing Application Configuration
- Java and Annotation Configuration
- Auto detection of Configuration File

Lombok API

- @Data
- @Getter
- @Setter
- @ToString
- @EqualsAndHashCode
- @NoArgsConstructor
- @RequiredArgsConstructor
- @AllArgsConstructor
- @NonNull

Basic Annotations and Execution flow

- @Service
- @Component
- @Scope
- @Bean
- @Configuration
- @ConfigurationProperties
- @Order
- @Profile
- @Value
- @PropertySource

B) Spring Boot Advanced

Spring Boot Web Applications

- Design of Web Applications
- Layers in Web Application
- Spring Boot MVC Model
- Writing of Controllers

Embedded Server and Databases

- Need of Embedded Components
- Knowing Servers Configuration
- Working with Tomcat
- Working Jetty Servers
- Working with Undertow
- H2 Database and Console
- HSQLDB
- Apache Derby

Spring Boot JDBC

- Curd Repository
- Basic Operations
- Execution with Different Dbs
- Database Connection Properties

Spring Boot Data JPA

- JPA Repository, JPA DB Operations (CURD)
- findAll and findBy methods
- @Query Annotation for HQL
- JPA Properties Configuration

Spring Boot MongoDB

- Mongo Repository
- Document and Models Design
- JSON Format
- Basic Operation
- Embedded Mongo Db
- External Mongo Configuration
- Mongo DB Security

Spring Boot AJAX

- Synchronous v/s Asynchronous
- AJAX introduction
- JQuery AJAX call models
- Integration with RestController

Spring Boot Connection Pooling

- Types of Temporary Memories
- Need of Pooling
- Pooling v/s caching
- Hikari CP Configuration

Spring Boot Pagination and Sorting

- Pagination Introduction
- Google Example for Pagination
- Pagination And Sorting Repository
- Page and Pageable objects
- default values for Pagination
- Sorting Directions

Spring Boot Security

- Boot Security Design
- Authentication and Authorizations
- Roles and AntMatcher ULR Patterns
- In Memory Authentication
- Default Form Logins
- Custom Form Creation
- Jdbc Authentication
- JPA/Hibernate(ORM) Authentication
- OAuth2 Security Design
- OAuth2 Client Application Design

Rest Web Services and Operations

- Need of JSON
- JSON and It's Auto Conversion
- Testing Using POSTMAN
- Types of HTTP Methods
- PUT
- GET
- DELETE
- POST
- Parameters (HeaderParam, Request Param..)
- ResponseEntity Format
- @RequestBody and @ResponseBody Format

Spring Boot with Swagger

- What is API Endpoints
- Swagger Introduction
- Design of SwaggerUI
- Coding for RestController
- Testing Swagger Operations

Spring Boot Profiles

- Need of Profiles
- Environments and Profiles
- Defining Profiles
- Execution Of profiles

Spring Boot DevTools

- What are DevTools
- Dependency for DevTools
- Working with DevTools

Spring Boot Task Scheduling

- Schedulers Introduction
- Task Management
- fixedDelay and fixedRate
- cron expression

Spring Boot Actuator

- Production REST Endpoints

- Sensitive and Insensitive end points
- Dependency for Actuator
- Executing endpoints
- finding logs
- Beans in container
- Health check
- Current Environment

Spring Boot Email

- Java MAIL API structure
- Mail Properties
- Boot Mail Design
- Simple Email
- Mime Email with Attachments

Spring Boot batch API

- Batch API Introduction
- Item Reader, Writer and Processor Design
- StepConfiguration
- JobConfiguration
- JobExecutionListener
- CSV to MySQL Batch Programming

MICROSERVICES

Monolithic Applications

- Introduction of Layers Application
- Modules Integration
- Implementation of Monolithic Applications

Drawbacks of Monolithic applications

- Problems in Design
- Load Balancing
- Parallel Coding and Testing

SOA Introduction

- What is SOA
- SOA as Design Pattern
- Components of SOA
- Consumer and Producers
- Registry and Discovery Component

Microservices Introduction

- Importance of microservices
- Services in Application
- Monolithic to Microservices
- Architecture and Implementation styles

Design of Microservices

- Eureka Server as Registry and Discovery

- Registry server
- Consumer Discovery

Intra Communication of Microservices

- Provider and Consumer Integration
- Sending/Receiving Data
- HTTP Methods Examples

Data passing types (JSON,XML)

- HTTP Parameters
- HTTP Body for Data passing
- Object to JSON Format
- Object to XML Format

Need of Load balancing

- What is Load balancing
- Types of Load Balancing
- Ribbon for Client Load balancing
- Zuul for Server Load Balancing

API creation for Microservices

- Defining Consumer for RestController
- Paths and Parameters
- Using API for Integration

Spring Cloud Architecture

- Components
- Service Registry
- API Gateway
- Config Server
- Config Client
- Load Balancer
- PCF
- Circuit Breaker
- Cloud Integrations

Discovery and Registry server

- Netflix Eureka Server Example
- Application Properties
- Dependencies Introduction
- Spring Boot version and Cloud version

Provider and Consumer Microservices

- RestTemplate Introduction
- Eureka server Registry
- Provider Creation and Register with Eureka
- Consumer Application and Register with Eureka
- Application Configuration for RestTemplate
- Consumer and Provider Integration

Config server and Config Client

- Need of Config Servers
- Local File System
- Git Config Servers
- Eureka with Config server
- Provider and Consumers with Config Client

API for Gateway Proxy

- What is API Gateway
- Netflix Zull Gateway example
- Zull project Properties
- Integration with Eureka and PC App

Loading Balancing

- Need Of load Balancing
- Netflix Ribbon Example
- Provider Load Balancing
- Ribbon with Eureka - PC APP

Fallback methods and Circuit Breaker

- Exception Handling in Cloud
- Fallback methods
- Hystrix Circuit Breakers
- Hystrix Dashboard

Securing API

- Security Design
- Securing End Points
- OAuth 2.0 Introduction
- OAuth2.0 Implementation

Zipkin and Sleuth

- Logging and Tracing in Microservices
- Log4J for Microservices
- Need of Sleuth
- Zipkin UI Tracer

Pivotal Cloud Foundry (PCF)

- What is PCF
- Register and Validate Account
- Installing PCF in system
- Push Microservice to PCF
- Service Dashboard
- PCF Basic Commands

Message Queue(MQ)

- Introductions to MQs
- Need of MQs
- Coding using Active MQ
- Rabbit MQ Example

Admin UI

- Admin UI usage
- Configure Dependency
- Actuator with Admin UI

Case study [Employee Module]

- Spring Boot Design and Coding
- Converting to Microservices
- Deploy in PFC
- Integrate with Spring Apps
- Integrate with Angular Apps

Spring Boot with Tools

- Eclipse, STS
- Maven
- Gradle
- Github
- start.spring.io
- JUnit With Mocking
- Logging
- Reference Documents
- Docker
- FAQs