Course Title : Spring Basics , Spring Boot & Micro Services
Duration : 120 sessions in 150 Sessions

Pre-requisites : Core Java (oops , exception handling, collections, ...)
JDBC basics
SQL Basics
Common Sense

Course Fee : 7000/- (without recordings)
12000/- (with recordings)

Course material : Initially the material, class room apps,
docs and etc.. will be given using
FB Group (natarazjavaarena)
==>Later some things will be shared
using Google drive

Faculty and admin details :

FB Group name : natarazjavaarena
FB group url : https://www.facebook.com/groups/[illegible]
To collect material : https://www.facebook.com/groups/[illegible]/files
(batch code : [illegible])
email id : natarazjavaarena@gmail.com
For new Batches info : https://nareshit.com/new-batches-hyderabad/
youtube url : https://www.youtube.com/[illegible]
Rajashekar (admin) : [illegible]

=>Spring/Spring boot frameworks are coming in the form of collection of modules (20+) modules
=> Examples modules are core, mvc, data jpa , data mongodb, security , cloud, scheduling , batch and etc...

=> spring boot framework is spring framework ...

Spring boot framework
spring
framework

Spring boot framework is the extension of spring framework

=>If we develop proj1 (same market) using java language let us assume we need to write 20K LOC
=>If we develop proj1 (same market) using java technologies let us assume we need to write 15K LOC
=>If we develop proj1 (same market) using spring let us assume we need to write 10K LOC
=>If we develop proj1 (same market) using spring boot let us assume we need to write 6K LOC (Lines of Code)

=> If any module is used in app /project development in spring style programming then it
is called spring module
eg:: spring core , spring mvc , spring batch , spring security and etc..

=> If any module is used in app /project development in spring boot style programming then it
is called spring boot module
eg:: spring boot core , spring boot mvc , spring boot security and etc..

=>All frameworks including spring and spring boot takes care of common logics generation
internally and dynamically to make the programmers to write only application specific logics

=> Frameworks provide abstraction on technologies and languages to simplify
the application development process

=> Abstraction means highlighting necessary details and hiding unnecessary
details from the programmers /users

Course plan

JPA :: Jakarta Persistence API

=>spring core (spring programming basics)
=>Spring boot core (spring boot programming basics)
=>spring boot data jpa (DB interaction and DB tables data manipulation (curd operations))
|---> for interacting with SQL Db s/w (oracle,mysql and etc..)
(fixed schema and fixed attributes)
=>spring boot mongodb (for NoSQL DB interaction and data manipulation)
|---> for interacting MongoDB s/w that deals with dynamically growing
unstructured data in the docs
=> spring boot Scheduling (For enabling scheduling on the jobs)
=> spring boot mailing (for mailing operations)
=> spring boot batch processing (reading , processing and writing huge amount data
batch by batch)
=>Spring boot mvc (For web application development)
=>spring boot Rest (for Distributed App development)
=>spring boot security (For securing web apps and Distributed apps)
=> spring boot oauth
=> Spring boot JWT (JSON web tokens)
=> spring boot okta

For advanced
security features implementation
like SSO (Single Sign On)

=> browser to s/w app interaction is called web application (or) online B2C
eg: browser to flipkart.com , browser to amazon.com
=> App to App interaction (B2B) is called Distributed App
eg: browser ---> flipkart app ---> phonepe app ---> Bank app
web application
Distributed App
Distributed App

=> Spring boot Actuators (provides set of readily available non-functional features
like threads info, processors info , outputs info , heap info..)

=> the main features of project like
opening account, closing a/c, withdraw,
deposit and etc.. of banking app are
called functional features
=> The supporting operations on the project
are called non-functional features
(threads info, heap info)

=> Spring boot Cloud (For developing Micro Service Architecture based
Projects)
=> Spring boot JMS (active MQ, Rabbit MQ and etc..)
(for enabling messaging based communication b/w
the software Apps or Projects in offline mode)
=> Spring boot apache kafka
(for more advanced messaging based communication)

=> Spring boot apps integration with UI (angular/ react JS)

=> Spring boot Mini Project (group)

=> 1 Mini Project
|---> Coding Standards
|---> Naming Conventions
|---> Design Patterns (class diagrams)
|---> tools support

=> Open topics (suggested by students)

=> We generally see projects in the market following
two architectures
a) Monolithic Arch (old and legacy)
b) Micro Services Arch (latest and hot cake)

=> In monolithic arch , all services will be combined into
single unit and will treated as single jar/war file

Bank App
Deposit
Service
Loan Service
Withdraw
Service
Login Service
SBI_BankApp.war/jar

Course Title :: Spring Basics, Spring Boot & Micro Services

Duration: 120 sessions to 130 Sesssions

Pre-requisites :: Core Java (oops, exception handling, collections,....)

JDBC basics

SQL Basics

Common Sense

Course Fee :: 7000/- (with out recordings)

12000/- (with recordings)

Course material: Initially the material, class room apps,

docs and etc.. will be given using

FBGroup(natarazjavaarena)

=>Later same things will be shared using Google drive

Faculty and admin details::

FB Group name :: natarazjavaarena

FB group url :: https://www.facebook.com/groups/388095825162910

To collect material :: https://www.facebook.com/groups/388095825162910/files (batch code ::NTSPBMS515)

email id :: natarazjavaarena@gmail.com

For new Batches Info :: https://nareshit.com/new-batches-hyderabad/

youtube url:: https://www.youtube.com/NareshIT

Rajeshwari (admin) ::: 7337313417

=>Spring/Spring boot frameworks are coming in the form of collection of modules (25+) modules

=> Example modules are core,mvc,data jpa, data mongodb, security, cloud, scheduling, batch and etc... => spring boot framework = spring framework++

Spring-boot framework

spring framework

Spring boot framework is the extension of spring framework

=>if we develop proj1 (super market) using java language let as assume we need to write 20K LOC =>if we develop proj1 (super market) using java technologies let as assume we need to write 15K LOC

=>if we develop proj1 (super market) using spring let as assume we need to write 10K LOC

=>if we develop proj1 (super market) using spring Boot let as assume we need to write 6K LOC (Lines of Code)

=> if any module is used in app/project development in spring style programming then it

is called spring module

eg:: spring core, spring mvc, spring batch, spring security and etc..

bone programming then it

=> if any module is used in app/project development in spring

is called springdule

eg: spring boot core, spring boot mvc, spring boot security and etc..

=>All frameworks including spring and spring boot takes care of common logics generation internally and dynamically to make the programmers to write only application specific logics

=> Frameworks provide abstraction on technologies and languages to simplify

the application development process

=> Abstraction means highlighting necessary details and hiding unnecessary details from the programmers/users

Course plan

==========

=>spring core (spring programming basics)

JPA :: Jakarta Persistence API

=>Spring boot core (spring boot programming basics)

=>spring boot data jpa (DB interaction and DB tables data manipulation (curd operations))

|---> for interacting with SQL DB s/ws (oracle, mysql and etc..)

(fixed schema and fixed attributes)

=>Spring boot mongoDB (for NoSQL DB interaction and data manipulation)

|----> for interacting MongoDB s/w that deals with dynamically growing unstructured data in the docs

=> spring boot Scheduling (For enabling scheduling on the jobs)

=> spring boot mailing (For mailing operations)

=> spring boot batch processing (reading, processing and write huge amount data

batch by batch)

=>Spring boot mvc (For web application development) => spring boot Rest (For Distributed App development)

=> browser to s/w app interaction is called web application/website(B2C) eg:: browser to flipkart.com, browser to nareshit.com => App to App interaction (B2B) is called Distributed App eg:: browser--> flipkart.com ----> phone App -----> web application Distributed App

=>spring boot security (For securing web apps and Distributed apps) => spring boot oauth

=> Spring boot JWT (JSON web Tokens) => spring boot okta

For advanced

security features implementation like SSO (Single Sign On)

=> Spring boot Actuators (provides set of readily available non-functional features like threads info, processes info, objects info, heap info..)

=> Spring boot Cloud

(For developing Micro Service Architecture based Projects)

=> Spring boot JMS (active MQ, Rabbit MQ and etc..) (For enabling messaging based communication b/w the software Apps or Projects in offline env..)

=> Spring boot apache kafka

(For more advanced messaging based communication)

=> Spring boot Apps integration with UI (angular/React JS)

=> Spring boot AI (Artificial Intelligence)

=> 4-5 Mini Project

Bank app

(SBI) Distributed App

=> the main features of project like opening account, closing a/c, withdraw, deposit and etc.. of banking app are called functional features

=> The supporting operations on the project are called non-functional features (threads info, heap info)

=> We generally see projects in the market following two architectures

a) Monolith Arch (old and legacy)

b) Micro Services Arch (latest and hot cake)

=> In monolith arch, all services will be combined into single unit and will related as single jar/war file

Bank App

|--> Coding Standards

|--> Naming Conventions

|--> Design Patterns (Best Practices)

|---> Tools support

=> Open topics (Suggested by Students)

Daily

Loans service

SBI BankApp.war/jar

Admin service

mutual funds

=> In Micro Services ARch, every service will be developed as separate Project having separate packing and all them will integrated using third party tools and softwares .Here

=> 10+ Realtime tools (maven, git, slf4j, junit, Mockito, SonarQube, jacoco, Jenkins, docker,JIRA and etc..) each project is called one MicroServices

=>10+ Common topics

--> Resume preparation

|---> Interview tips

Proj1.jar/war

Proj1 (MS#1) DailyTx Service

Proj2 (MS#2)

Loans service

Proj2.jar/war

|---> Knowing about IT Industry

|---> Building a Strong Software Career

Proj3 (M3#1)

|---> How to show gap as the experience

|---> Hot to arrange the projects and how to explain the projects

Proi4 (MS#4) admin service

Mutual funds Service

All MS projects

will be integrated

and etc..

Proj3.jar/war

using third party tools and sofftwares

Proj4.jar/war

In java Domain there are 3 categories of subjects to learn

Programming Languages

========================

a) Java Programming Language (Core java course)

b) Java Technologies (adv.java course)

c) Java Framework (Spring, spring boot and MicroService Course)

=>Programming languages are raw materials of programming providing basic features of the Application devleopment

=> These are directly installable softwares providing compilers and interpreters for checking the syntaxes (rules of programming) and semantics (structure of the programming) of the application Coding.

=>These are base for creating Technologies, frameworks, tools, Operating systems,DB s/ws and etc.. either directly or indirectly

eg:: Java, c,c++, C#, ada, small talk, java script and etc...

=> There are multiple JVM based languages which have got their own compilers and Programming syntaxes and semantics, but all them can use JVM/JRE to execute the compiled byte class (.class file). So these are called JVM based languages

.java

(best) java compiler

.scala

>.class (byte code) scala.compiler...> .class (byte code)- .kts(kotlin) kotlin compiler...>.class (byte code)

JRE/JVM

.groovy

groovy compiler

--->.class (byte code),

and etc.