



# MODULE - 1

## SPRING - SPRING BOOT - SPRING MVC - SPRING AOP - SPRING SECURITY

### Course Outline

#### Chapter 1- Spring Framework Basics

- 1 - Setting up a Spring Project
- 2 - Understanding Tight Coupling using the Binary Search Algorithm Example
- 3 - Making the Binary Search Algorithm Example Loosely Coupled
- 4 - Using Spring to Manage Dependencies - @Component, @Autowired
- 5 - What is happening in the background?
- 6 - Dynamic auto wiring and Troubleshooting - @Primary
- 7 - Constructor and Setter Injection
- 8 - Spring Modules
- 9 - Spring Projects
- 10 - Why is Spring Popular?

#### Chapter 2- Spring Framework in Depth

- 1 - Dependency Injection - A few more examples
- 2 - Autowiring in Depth - by Name and @Primary
- 3 - Autowiring in Depth - @Qualifier annotation
- 4 - Scope of a Bean - Prototype and Singleton
- 5A - Complex scenarios with Scope of a Spring Bean - Mix of Prototype and Singleton
- 5B - Difference Between Spring Singleton and GOF Singleton
- 6 - Using Component Scan to scan for beans
- 7 - Lifecycle of a Bean - @PostConstruct and @PreDestroy
- 8 - Container and Dependency Injection (CDI) - @Named, @Inject
- 9 - Removing Spring Boot in Basic Application
- 10 - Fixing minor stuff - Add Logback and Close Application Context
- 11 - Defining Spring Application Context using XML - Part 1
- 12 - Defining Spring Application Context using XML - Part 2
- 13 - Mixing XML Context with Component Scan for Beans defined with Annotations

- 14 - IOC Container vs Application Context vs Bean Factory
- 15 - @Component vs @Service vs @Repository vs @Controller
- 16 - Read values from an external properties file

## Chapter 3- JUNIT and MOCKITO Basics

- 1 - What is JUnit and Unit Testing?
- 2 - First JUnit Project and Green Bar
- 3 - First Code and First Unit Test
- 4 - Other assert methods
- 5 - Important annotations
- 6 - Setting up an example using <http://start.spring.io>.
- 7 - Using Stubs - Disadvantages
- 8 - Your first mock with Mockito
- 9 - Using Mockito Annotations - @Mock, @InjectMocks, @RunWith
- 10 - Mocking List interface

## Chapter 4- UNIT Testing with Spring Framework

- 1 - Spring Unit Testing with a Java Context
- 2 - Spring Unit Testing with an XML Context
- 3 - Spring Unit Testing with Mockito

## Chapter 5- Spring Boot Basics

- 1 - Introduction to Spring Boot - Goals and Important Features
- 2 - Developing Spring Applications before Spring Boot
- 3 - Using Spring Initializr to create a Spring Boot Application
- 4 - Creating a Simple REST Controller
- 5 - What is the Spring Boot Auto Configuration?
- 6 - Spring Boot vs Spring
- 7 - Spring Boot Starter Projects - Starter Web and Starter JPA
- 8 - Overview of different Spring Boot Starter Projects
- 9 - Spring Boot Actuator
- 10 - Spring Boot Developer Tools

## Chapter 6- Spring AOP

- 1 - Setting up the AOP Example - Part 1
- 2 - Setting up the AOP Example - Part 2
- 3 - Defining @Before advice
- 4 - Understand AOP Terminology - Pointcut, Advice, Aspect, JoinPoint, Weaving and Weaver
- 5 - Using @After, @AfterReturning, @AfterThrowing advice
- 6 - Using @Around advice to implement performance tracing
- 7 - Best Practice - Use common Pointcut Configuration
- 8 - Quick summary of other Pointcuts
- 9 - Creating Custom Annotation and an Aspect for Tracking Time

## Chapter 7- Basic Java Web Application with JSP And Servlets

- 01 - Setting up Your First Java Web Application
- 02 - Theory 1 - Maven and Magic
- 03 - Theory 2 - What is a Servlet?
- 04 - Theory 3 - Web Application Request Flow
- 05 - Theory 4 - Understand Your First Servlet - LoginServlet
- 06 - Create LoginServlet From Scratch Again and Your First View
- 07 - Theory - Play Time - Let's Try Breaking Things
- 08 - Passing Request Parameters using Get Method
- 09 - Theory - Introduction and End to Scriptlets
- 10 - Disadvantages of Get Parameters
- 11 - Your First Post Request
- 12 - Your First Servlet doPost Method
- 13 - Let's Add a Password Field

## Chapter 8- Spring MVC

- 00 - Setting up Maven, Tomcat and Simple JEE Application
- 01 - Configure an application to use Spring MVC
- 02 - First Spring MVC Controller, @ResponseBody, @Controller
- 03 - Redirect to Login JSP - LoginController, @ResponseBody and View Resolver
- 04 - DispatcherServlet and Log4j
- 05 - Show userid and password on the welcome page - ModelMap and @RequestParam
- 06 - LoginService and Remove all JEE Servlets based code

- 07 - Spring Auto-wiring and Dependency Management - @Autowired and @Service
- 08 - Create TodoController and list-todos view. Make TodoService a @Service and inject it.
- 09 - Web Application Architecture
- 10 - More about Spring Framework
- 11 - Session vs Model vs Request - @SessionAttributes
- 12 - New Todo and redirect to a Controller
- 13 - JSTL
- 14 - Bootstrap - using Webjars
- 15 - Let's delete a Todo
- 16 - Use Bootstrap to format and add HTML5 Validations
- 17 - Introduce JSR 349 Validations using Hibernate Validator - First Command Bean.
- 18 - Let's update a Todo
- 19 - Let's add a Target Date for Todo - Use initBinder to Handle Date Fields
- 20 - Navigation bar and JSP Fragments

## Chapter 9- Spring Security

- 01 - Let's prepare for Spring Security
- 02 - Initial Setup for Spring Security
- 03 - Refactor and add Logout Functionality using Spring Security
- 04 - Exception Handling in Spring MVC - @ControllerAdvice, @ExceptionHandler & error-page in web dot xml
- 05 - Let's add Internationalization - i18n
- 06 - Basic Spring Rest Services - @RestController and Jackson-databind
- 07 - More Rest Services - @PathVariable

### Course Requirements

- You have some experience with Java Web Applications but did not completely understand the basics
- You want to learn the basics of Spring MVC and gain expertise in it
- We will use an In-Memory List as Data Store in this Beginner's Course

### Course Outcomes

- Basics of Spring Framework - Dependency Injection, IOC Container, Application Context and Bean Factory
- Spring Annotations - @Autowired, @Component, @Service, @Repository, @Configuration, @Primary....
- Spring MVC in depth - DispatcherServlet, Model, Controllers and ViewResolver
- Spring Boot Starters - Spring Boot Starter Web, Starter Data Jpa, Starter Test
- Basics of Spring Boot, Spring AOP, Spring JDBC and JPA

- Basics of Eclipse, Maven, JUnit and Mockito
- The basic concept of a Web application step by step using JSP Servlets and Spring MVC
- Unit testing with JUnit and Mockito using XML and Java Spring Application Contexts
- Basic Todo Management Application with Login/Logout
- Model, Controllers, ViewResolver and Filters
- Forms - DataBinding, Validation
- Annotation-based approach - @RequestParam, @PathVariable, @ModelAttribute, @SessionAttributes
- Bootstrap to style the page
- Spring Security
- Internationalization
- Exception Handling
- Basic REST Services



# MODULE - 2

## SPRING JPA – SPRING JDBC- HIBERNATE

### Course Outline

#### Chapter 1 - Introduction

- 01 - Master Hibernate and JPA with Spring Boot - Preview
- 02 - Master Hibernate and JPA with Spring Boot - Installing Basic Tools
- 03 - Quick introduction to JPA

#### Chapter 2- Journey from Spring JDBC to JPA

- 01 - Setting up a project with JDBC, JPA, H2 and Web Dependencies
- 02 - Launching up H2 Console
- 03 - Creating a Database Table in H2
- 04 - Populate data into Person Table
- 05 - Implement findAll persons Spring JDBC Query Method
- 06 - Execute the findAll method using CommandLineRunner
- 07 - A Quick Review - JDBC vs Spring JDBC
- 08 - What's in the background? Understanding Spring Boot Autoconfiguration
- 09 - Implementing findById Spring JDBC Query Method
- 10 - Implementing deleteById Spring JDBC Update Method
- 11 - Implementing insert and update Spring JDBC Update Methods
- 12 - Creating a custom Spring JDBC RowMapper
- 13 - Quick introduction to JPA
- 14 - Defining Person Entity
- 15 - Implementing the findById JPA Repository Method
- 16 - Implementing insert and update JPA Repository Methods
- 17 - Implementing deleteById JPA Repository Method
- 18 - Implementing findAll using JPQL Named Query

## Chapter 3– JPA/Hibernate in Depth

- 01 - Create a JPA Project with H2 and Spring Boot
- 02 - Create JPA Entity Course
- 03 - Create findById using JPA Entity Manager
- 04 - Configuring application. properties to enable H2 console and additional logging
- 05 - Writing Unit Test for findById method
- 06 - Writing a deleteById method to delete an Entity
- 07 - Writing Unit Test for deleteById method
- 08 - Writing a save method to update and insert an Entity
- 09 - Writing Unit Test for the save method
- 10 - Quick Review and Debugging Tips
- 11 - Playing with Entity Manager
- 12 - Entity Manager Methods - clear and detach
- 13 - Entity Manager Methods - refresh
- 14 - A Quick Review of Entity Manager
- 15 - JPQL - Basics
- 16 - JPA and Hibernate Annotations - @Table
- 17 - JPA and Hibernate Annotations - @Column
- 18 - JPA and Hibernate Annotations - @UpdateTimestamp and @CreationTimestamp
- 19 - JPA and Hibernate Annotations - @NamedQuery and @NamedQueries
- 20 - Native Queries - Basics

## Chapter 4– Establishing Relationships with JPA & Hibernate – One to One

- 01 - Entities and Relationships - An overview
- 02 - Defining Entities - Student, Passport and Review
- 03 - Introduction to One to One Relationship
- 04 - OneToOne Mapping - Insert Student with Passport
- 05 - OneToOne Mapping - Retrieving Student with Passport and Eager Fetch
- 06 - OneToOne Mapping - Lazy Fetch
- 07 - Session vs Transaction
- 08 - OneToOne Mapping - Bidirectional Relationship - Part 1
- 09 - OneToOne Mapping - Bidirectional Relationship - Part 2

## Chapter 5- Establishing Relationships with JPA & Hibernate – One to Many and Many to Many

- 00 - ManyToOne Mapping - Designing the database
- 01 - ManyToOne Mapping - Retrieving and inserting Reviews for Course
- 02 - ManyToOne Mapping - Generalizing Insert Reviews
- 03 - ManyToOne Mapping - Wrapping up
- 04 - ManyToMany Mapping - Table Design
- 05 - ManyToMany Mapping - Adding Annotations on Entities
- 06 - ManyToMany Mapping - Fixing two join tables problem
- 07 - ManyToMany Mapping - Customizing the Join Table
- 08 - ManyToMany Mapping - Insert Data and Write Join Query
- 09 - ManyToMany Mapping - Retrieve Data using JPA Relationships
- 10 - ManyToMany Mapping - Insert Student and Course

## Chapter 6- Inheritance Hierarchies with JPA And Hibernate

- 01 - Introduction to Inheritance Hierarchies and Mappings
- 02 - JPA Inheritance Hierarchies and Mappings - Setting up entities
- 03 - JPA Inheritance Hierarchies and Mappings - Setting up a Repository
- 04 - JPA Inheritance Hierarchies and Mappings - Single Table
- 05 - JPA Inheritance Hierarchies and Mappings - Table Per Class
- 06 - JPA Inheritance Hierarchies and Mappings - Joined
- 07 - JPA Inheritance Hierarchies and Mappings - Mapped Super Class
- 08 - JPA Inheritance Hierarchies and Mappings - How to Choose?

## Chapter 7- Queries with Entities Using JPQL

- 00 - JPQL - Courses without Students
- 01 - JPQL - Courses with at least 2 Students and order by
- 02 - JPQL - Courses like 100 s
- 03 - JPQL - Using Joins

## Chapter 8- Queries Using Java API - Criteria Queries

- 01 - Criteria Query - Retrieving all courses
- 02 - Criteria Query - Courses like 100 s
- 03 - Criteria Query - Courses without Students
- 04 - Criteria Query - Using Joins



## Chapter 9- Transaction Management

- 01 - Introduction to Transaction Management
- 02 - Transaction Management - ACID Properties
- 03 - Understanding Dirty, Phantom and Non-Repeatable Reads
- 04- Understand 4 Isolation Levels
- 05 - Choosing between Isolation Levels
- 06 - Implementing Transaction Management - 3 Things to Decide

## Chapter 10- Spring Data JPA & Spring Data Rest

- 01 - Introduction to Spring Data JPA
- 02 - Testing the Spring Data JPA Repository with findById.
- 03 - Spring Data JPA Repository - CRUD Method
- 04 - Sorting using Spring Data JPA Repository
- 05 - Pagination using Spring Data JPA Repository
- 06 - Custom Queries using Spring Data JPA Repository
- 07 - Spring Data REST

## Chapter 11- Caching with Hibernate & JPA

- 01 - Introduction to Caching
- 02 - Hibernate and JPA Caching - First Level Cache
- 03 - Hibernate and JPA Caching - Basics of Second Level Cache with EhCache
- 04 - Hibernate and JPA Caching - Second Level Cache Part 2

## Chapter 12- Hibernate & JPA Tips

- 01 - Hibernate Tips - Hibernate Soft Deletes - @SQLDelete and @Where
- 02 - Hibernate Soft Deletes - Part 2
- 03 - JPA Entity Life Cycle Methods
- 04 - Using Embedded and Embeddable with JPA
- 05 - Using Enums with JPA
- 06 - JPA Tip - Be cautious with toString method implementations
- 07 - JPA Tip - When do you use JPA?

## Chapter 13- Performance Tuning Tips with Hibernate & JPA

- 01 - Performance Tuning - Measure before Tuning
- 02 - Performance Tuning - Indexes
- 03 - Performance Tuning - Use Appropriate Caching
- 04 - Performance Tuning - Eager vs Lazy Fetch
- 05 - Performance Tuning - Avoid N+1 Problems

## Chapter 14- Few More FAQ

FAQ 1 - How to connect to a different database with Spring Boot?

FAQ 2 - Approach to design great applications with JPA?

FAQ 3 - Good Practices for developing JPA Applications

### Course Requirements

- You have an attitude to learn while having fun
- You have programming experience with Java, Spring

### Course Outcome

- You will learn the basics of JPA and Hibernate - Entities, Relationships, Inheritance Mappings and Annotations
- You will understand approaches to querying data using JPA and Hibernate - JPQL, Criteria API and Native Queries
- You will understand JPA and Hibernate Relationships in-depth - One to One, Many to One and Many to Many
- You will use a variety of Spring Boot Starters - Spring Boot Starter Web, Starter Data Jpa, Starter Test
- You will learn the basics of performance tuning your JPA application with Hibernate - Solve N+1 Queries Issue.
- You will learn the basics of caching - First Level Cache and Second Level Cache with EhCache
- You will understand the basics of Spring Data JPA and Spring Data REST



# MODULE - 3

## WEBSERVICES – REST & SOAP

## MICROSERVICES – SPRING CLOUD

### Course Outline

#### Chapter 1- Introduction to Web Services

- 01 - Introduction to Web Services - An Overview
- 02 - What is a Web Service?
- 03 - Important How Questions related to Web Services
- 04 - Web Services - Key Terminology
- 05 - Introduction to SOAP Web Services
- 06 - Introduction to RESTful Web Services
- 07 - SOAP vs RESTful Web Services

#### Chapter 2- Soap Web Services with Spring and Spring Boot

- 01 - Initialize a Spring Web Services application with Spring Boot
- 02 - Overview of creating SOAP Web Service using Contract First Approach
- 03 - Define Request and Response XML Structure
- 04 - Define XML Schema Definition (XSD) for Request - GetCourseDetailsReque
- 05 - Define XML Schema Definition (XSD) for Response - GetCourseDetailsResp
- 06 - More about XML Schema Definition and Implementing XSD Best Practices
- 07 - Introduction to Java API for XML Binding (JAXB) and Configuring JAXB 2
- 08 - Configuring an Endpoint for GetCourseDetailsRequest
- 09 - Spring Web Services Configuration - Message Dispatcher Servlet
- 10 - Spring Web Services Configuration - Generating WSDL
- 11 - Using Wizdler to execute SOAP Requests
- 12 - Implementing a service - Course Details Service - backend within the memo
- 13 - Implementing SOAP Web Service for GetAllCourseDetailsRequest
- 14 - Quick introduction to different parts of a WSDL
- 15 - Implementing SOAP Web Service for DeleteCourseDetailsRequest
- 16 - Improving the DeleteCourseDetailsRequest - Using an Enum for Status
- 17 - Exception Handling and SOAP Fault Responses
- 18 - Implementing Security for SOAP Web Services with WS Security

## Chapter 3- Restful Web Services with Spring and Spring Boot

- 01 - Initializing a RESTful Services Project with Spring Boot
- 02 - Understanding the RESTful Services we would create in this course
- 03 - Creating a Hello World Service
- 04 - Enhancing the Hello World Service to return a Bean
- 05 - Quick Review of Spring Boot Auto Configuration and Dispatcher Servlet
- 06 - Enhancing the Hello World Service with a Path Variable
- 07 - Creating User Bean and User Service
- 08 - Implementing GET Methods for User Resource
- 09 - Implementing the POST Method to create User Resource
- 10 - Enhancing the POST Method to return correct HTTP Status Code and Location
- 11 - Implementing Exception Handling - 404 Resource Not Found
- 12 - Implementing Generic Exception Handling for all Resources
- 13 - Exercise - User Post Resource and Exception Handling
- 14 - Implementing DELETE Method to delete a User Resource
- 15 - Implementing Validations for RESTful Services
- 16 - Implementing HATEOAS for RESTful Services
- 17 - Overview of Advanced RESTful Service Features
- 18 - Internationalization for RESTful Services
- 18 - Part 2 - Internationalization
- 19 - Content Negotiation - Implementing Support for XML
- 20 - Configuring Auto Generation of Swagger Documentation
- 21 - Introduction to Swagger Documentation Format
- 22 - Enhancing Swagger Documentation with Custom Annotations
- 23 - Monitoring APIs with Spring Boot Actuator
- 24 - Implementing Static Filtering for RESTful Service
- 25 - Implementing Dynamic Filtering for RESTful Service
- 26 - Versioning RESTful Services - Basic Approach with URIs
- 27 - Versioning RESTful Services - Header and Content Negotiation Approach
- 28 - Implementing Basic Authentication with Spring Security

## Chapter 4- Introduction to JPA in 10 Steps

Introduction to JPA - An Overview

- 1 - Object-Relational Impedance Mismatch

- 2 - World before JPA - JDBC, Spring JDBC and myBatis
- 3 - Introduction to JPA
- 4 - Creating a JPA Project using Spring Initializr
- 5 - Defining a JPA Entity - User
- 6 - Defining a Service to manage the Entity - UserService and EntityManager
- 7 - Using a Command-Line Runner to save the User to the database.
- 8 - Magic of Spring Boot and In-Memory Database H2
- 9 - Introduction to Spring Data JPA
- 10 - More JPA Repository - findById and findAll

## Chapter 5- Connecting Restful Web Service to JPA

- 29 - Overview of Connecting RESTful Service to JPA
- 30 - Creating a User Entity and some test data
- 31 - Updating GET methods on User Resource to use JPA
- 32 - Updating POST and DELETE methods on User Resource to use JPA
- 33 - Creating Post Entity and Many to One Relationship with User Entity
- 34 - Implementing a GET service to retrieve all Posts of a User
- 35 - Implementing a POST service to create a Post for a User

## Chapter 6- Restful Web Services - Best Practices

- 36 - Richardson Maturity Model
- 37 - RESTful Web Services - Best Practices

## Chapter 7- Microservices with Spring Cloud

- 00 - 01 - Introduction to Microservices
- 00 - 02 - Challenges with Microservices
- 00 - 03 - Introduction to Spring Cloud
- 00 - 04 - Advantages of Microservices Architectures
- 00 - 05 - Microservice Components - Standardizing Ports and URL
- 01 - Part 1 - Intro to Limits Microservice and Spring Cloud Config Server
- 01 - Part 2 - Setting up Limits Microservice
- 02 - Creating a hard-coded limits service
- 03 - Enhance limits service to get configuration from application properties
- 04 - Setting up Spring Cloud Config Server

- 05 - Installing Git
- 06 - Creating Local Git Repository
- 07 - Connect Spring Cloud Config Server to Local Git Repository
- 08 - Configuration for Multiple Environments in the Git Repository
- 09 - Connect Limits Service to Spring Cloud Config Server
- 10 - Configuring Profiles for Limits Service
- 11 - A review of Spring Cloud Config Server
- 12 - Introduction to Currency Conversion and Currency Exchange Microservice
- 13 - Setting up Currency Exchange Microservice
- 14 - Create a simple hardcoded currency exchange service
- 15 - Setting up Dynamic Port in the Response
- 16 - Configure JPA and Initialized Data
- 17 - Create a JPA Repository
- 18 - Setting up Currency Conversion Microservice
- 19 - Creating a service for currency conversion
- 20 - Invoking Currency Exchange Microservice from Currency Conversion Micro
- 21 - Using Feign REST Client for Service Invocation
- 22 - Setting up client-side load balancing with Ribbon
- 23 - Running client-side load balancing with Ribbon
- 24 - Understand the need for a Naming Server
- 25 - Setting up Eureka Naming Server
- 26 - Connecting Currency Conversion Microservice to Eureka
- 27 - Connecting Currency Exchange Microservice to Eureka
- 28 - Distributing calls using Eureka and Ribbon
- 29 - A review of implementing Eureka, Ribbon and Feign
- 30 - Introduction to API Gateways
- 31 - Setting up Zuul API Gateway
- 32 - Implementing Zuul Logging Filter
- 33 - Executing a request through Zuul API Gateway
- 34 - Setting up Zuul API Gateway between microservice invocations
- 35 - Introduction to Distributed Tracing
- 36 - Implementing Spring Cloud Sleuth
- 37 - Introduction to Distributed Tracing with Zipkin
- 38 - Installing Rabbit MQ
- 39 - Setting up Distributed Tracing with Zipkin

- 40 - Connecting microservices to Zipkin
- 41 - Using Zipkin UI Dashboard to trace requests
- 42 - Understanding the need for Spring Cloud Bus
- 43 - Implementing Spring Cloud Bus
- 44 - Fault Tolerance with Hystrix

### Course Requirements

- You have an attitude to learn while having fun -)
- You have programming experience with Java, Spring

### Course Outcome

- You want to learn the basics of Web Services
- You want to learn the terminology associated with Web Services
- You want to learn to develop and design RESTful web services
- You want to learn to develop and design SOAP web services
- You want to learn how to design and develop RESTful web services with Spring Boot
- You want to learn how to design and develop Microservices with Spring Cloud