

Sela.



Spring

Spring Framework Programming



college@sela.co.il

03-6176666





Spring Framework Programming

Spring - Version: 2.2

 5 days Course

Description:

This is a 5 day hands-on training on Spring framework covering a wide spectrum of topics starting from the basic yet fundamental concepts such as dependency injection all the way up to the advanced topics such as security, AOP, and transaction. The training is structured in a way a student will learn these topics by actually doing lots of hands-on programming exercises.

Intended audience:

The course is for java professionals that would like to work in the spring framework

Prerequisites:

Java programming experience
Web application programming experience

Objectives:

Learn how Dependency Injection is used in Spring framework
Learn how to build enterprise quality web application using Spring MVC
Learn how to add security, AOP, transaction to the Spring applications
Learn how to test the applications
Learn how to build RESTful services to the application

Topics:



Introduction to J2EE

- Tiered model architectures – principles and goals
- J2EE definition and characteristics
- J2EE technologies in a multi-tier architecture

Introduction to Spring

- Spring Framework Need and Introduction
- Spring Building Blocks

Spring Overview and Tools

- What is Spring framework?
- Why Spring framework?
- Spring framework architecture
- Usage scenario
- Tools (we are going to use in this course)
- Spring Helloworld Refactoring

Spring Dependency Injection Basics

- What is and Why Dependency Injection (DI)?
- Dependency Injection variants
- DI classes in Spring framework
- DI parameter types
- Bean naming
- @Autowired annotation
- Auto-scanning



Spring Dependency Injection Annotation

- Annotation-based Dependency Injection
 - @Autowired
 - @Required
- Qualifier
 - @Qualifier
 - Custom qualifier
- JSR 330 (Dependency Injection for Java)
 - @Inject
- JSR 250 (Common Annotations)
 - @PostConstruct & @PreDestroy
 - @Resource
- Java-based Spring configuration (instead of XML configuration file)
 - @Configuration
 - @Bean
- @Component and further stereotyped annotations
 - @Service
 - @Repository
 - @Controller
- Auto scanning

Spring 3 and Maven

- Maven Repositories for Spring artifacts
- Creating Spring Maven projects

Spring MVC



- Spring 3 MVC Basics I
 - Introduction to Spring MVC
 - DispatcherServlet
 - Context configuration
 - Controller
 - Annotation-based controllers
 - Request mapping
 - URI template
 - Request mapping with other means
 - Handler method arguments
 - Handler methods return types
- Spring 3 MVC Basics II
 - Handler mapping customization
 - Resolving views
 - XML namespaces brief tutorial
 - Spring 3 `<mvc:xxx>` configuration
- Spring 3 MVC Form handling
 - 2-phase form submission handling
 - Command/form objects
 - `@ModelAttribute`
 - Data binding
 - Validation
 - Redirect
 - Form tags
- Spring 3 MVC Misc.
 - Exception handling
 - Locale handling
 - `@Value`
 - SpEL
- Spring 3 MVC Views
 - `ContentNegotiatingViewResolver`



- Tiles
- Velocity and Freemarker
- PDF or Excel
- JasperReports
- Spring 3 MVC REST
 - Annotation-based REST support
 - URI template
 - Content negotiation
 - Feed views
 - XML marshaling views
 - JSON mapping views
 - HTTP method conversion
 - RestTemplate
 - Exception handling
 - Message conversion
- Spring 3 MVC Ajax

Spring 3 AOP

- What is and Why AOP?
- AOP concepts and terminology
- @AspectJ support in Spring
- Types of advice
- Declaring advices
- Accessing Join point information
- Declaring a pointcut
- Defining and using common pointcuts
- Introduction

Spring 3 Database



- DAO support
- @Repository annotation
- Data access through JDBC
 - JdbcTemplate class
 - NamedParameterJdbcTemplate class
 - SimpleJdbcTemplate and SimpleJdbcDaoSupport classes
- Data access through ORM
 - Hibernate
 - JPA

Spring 3 Transaction

- Transaction management in Spring framework
- Global transaction vs. local transaction
- PlatformTransactionManager interface
- Declarative transaction management
 - XML-based
 - Annotation-based
- Transaction propagation

Spring 3 Testing

- Testing requirements
- Unit testing without using Spring DI
- Unit testing using Spring DI
- Spring TestContext framework
- Transaction-aware testing
- Spring MVC Testing
- Spring Testing Annotations