

Java Development met Spring

Doelgroep Cursus Java Development met Spring

Ervaren Java developers die het Spring Framework willen gebruiken voor Java Desktop, Web of Enterprise applicaties.

Voorkennis Cursus Java Development met Spring

Ervaring met programmeren in Java en object oriëntatie is vereist om deel te kunnen nemen aan deze cursus. Basis kennis van web applicaties en XML is bevorderlijk voor een goede begripsvorming.

Uitvoering Training Java Development met Spring

De concepten worden behandeld aan de hand van presentatie slides en demo's. De theorie wordt afgewisseld met oefeningen. De cursustijden zijn van 9.30 tot 16.30.

Certificering Java Development met Spring

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Java Development met Spring.

Duur: 5 dagen

Prijs: € 2999

[Open Rooster](#)



**Java Development
with Spring**



Inhoud Java Development met Spring

De cursus Java Development met Spring behandelt de concepten, componenten en architectuur van het Spring Framework. De cursus wordt uitgevoerd in de nieuwste Spring versie met Spring Boot. In de cursus staan annotations centraal bij de configuratie van de Spring Applicaties.

Spring Intro

De cursus Java Development met Spring gaat van start met een overzicht van de verschillende onderdelen van Spring en de soorten applicaties waarin Spring gebruikt kan worden.

Spring Boot

Vervolgens wordt ingegaan op de essenties van Spring Boot waarmee de configuratie van Spring applicaties enorm is vereenvoudigd. Aan de orde komen de voornaamste Spring Boot annotations, de opinionated defaults en de ingebouwde web servers.

Dependency Injection

Ruime aandacht wordt besteed aan het concept van Dependency Injection of Inversion of Control dat een centrale rol speelt in het framework. Verschillende varianten van een dependency injection zoals setter injection en constructor injection worden besproken.

Aspect Oriented Programming

De andere pijler van het framework, Aspect Oriëntatie, komt eveneens uitgebreid aan de orde. De concepten van Aspect Orientation zoals Aspects, Joinpoints, Pointcuts, Advice en Weaving worden toegelicht.

Data Access

Vervolgens worden de verschillende opties om de gegevens van Spring Java applicaties in databases op te slaan behandeld. Er wordt aandacht besteed aan het gebruik van JDBC met een JdbcTemplate, alsmede op het gebruik van Object Relational Mapping frameworks zoals Hibernate met een HibernateTemplate of JPA via annotaties. In dit verband worden ook Spring Transacties besproken.

Spring MVC

Web Applications met het Spring MVC Framework zijn ook onderdeel van het programma van de cursus Java Development met Spring evenals de creatie en het gebruik van Spring Rest Web Services. Hierbij wordt de rol van controllers, views, page parameters en command objects besproken.

Spring Security

Tot slot wordt aandacht besteed aan Spring en Security en wordt het vereenvoudigd opzetten van een Spring configuratie met Spring Boot besproken. De modules Spring met JMS en Spring met JMX zijn optioneel.

Modules Java Development met Spring

Module 1 : Spring Introduction	Module 2 : Spring Boot	Module 3 : Dependency Injection
Spring Mission Statement Spring Addressing Layers Spring Characteristics Spring Jars Core Container Packages Spring AOP Packages Data Access Packages Spring Web Packages Spring Versions Spring Projects Spring Configuration	What is Spring Boot? Advantages Spring Boot Spring Boot Flavors Key Spring Boot Components Spring Boot Starter Starter Dependencies Spring Boot Autoconfigurator @SpringBootApplication Spring Boot CLI Spring Boot Internals Spring Boot Actuator	Benefits of DI BeanFactory Bean Configuration File Application Context Setter Injection Constructor Injection Beans Life Cycle Injection Simple Values Autowiring Aware Interfaces ResourceLoader
Module 4 : Application Configuration	Module 5 : Aspect Orientation	Module 6 : Spring Persistence
XML Configuration with Namespaces Best Practices XML Configuration Bean Definition Inheritance Spring XML Namespaces Component Scanning Configuration with Annotations Context Namespace Spring Expression Language Autowiring Spring Java Configuration Spring Expression Language PropertyPlaceholderConfiguration	The need for AOP Crosscutting Concerns AOP Key Terms ProxyFactoryBean Spring AOP Configuration Aspects and Advices Join Points and Pointcut AspectJ Configuration Advice Arguments Programmatic Proxies Regexp Pointcut NameMatchMethod	Java Persistence Traditional Persistence Transparent Persistence Shared Persistence Concepts DAO Design Pattern DAO Pattern JDBC Integration with IoC DAO Portability Spring DAO Concepts Transaction Management Spring Exceptions Exception Translation
Module 7 : Spring JDBC	Module 8 : Spring ORM	Module 9 : Transactions
JDBC Architecture Executing Statements JDBC Drivers and URL's Spring JDBC Data Access Spring DAO with JdbcTemplate Data Source Injection Querying using JdbcTemplate RowMapper Querying and Populating Objects Updating with JdbcTemplate ResultSetExtractor Callbacks NamedParameterJdbcTemplate	Spring and Hibernate Integration Mapping Classes HibernateTemplate Implementation HibernateTemplate HibernateTemplate execute Hibernate Annotations Spring and JPA Using JPA API Persistence Unit Configuration Container Managed Transactions Externalizing DB Properties Entity Manager from JNDI JPA Java Configuration	Using Spring Transactions Transaction Strategy TransactionDefinition TransactionStatus Injecting Implementations Declarative Transactions Transactional Methods Transaction Configuration Rollback Rules Using @Transactional Transaction Propagation Programmatic Transactions Using TransactionTemplate
Module 10 : Spring REST	Module 11 : Spring and Security	Module 12 : Spring JMS
@RestController HttpEntity and ResponseEntity Default Content Types Default Status Codes @ResponseStatus and HttpStatus XML and JSON Multiple Representations Filtering with @JsonView RestTemplate Sending HTTP Requests Translating Entities Reading Responses Error Handlers	Spring Security Security Concerns Spring Authentication Spring Security Modules Spring Security Elements Defining Allowed Users Form Authentication Authentication Managers Security Interceptor Security Filters Access Decision Manager Web Authorization Method Authorization	Messaging Characteristics Messaging Models JMS Architectural Components Spring JMS JMS with Spring Spring JMS Options Spring JmsTemplate Send and Convert Execute Method ProducerCallback Receive and Convert Message Listener Containers MessageListenerAdapter