

Spring Boot Development

Doelgroep Cursus Spring Boot Development

De cursus Spring Boot Development is bestemd voor ervaren Java Developers die Spring Boot willen gebruiken voor applicatie ontwikkeling.

Voorkennis Cursus Spring Boot Development

Ervaring met programmeren in Java en object oriëntatie is vereist om deel te kunnen nemen aan deze cursus. Voorkennis van het Spring Framework is bevorderlijk voor een goede begripsvorming.

Uitvoering Training Spring Boot Development

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

Certificering Spring Boot Development

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

Duur: 4 dagen

Prijs: € 2650

[Open Rooster](#)



Inhoud Cursus Spring Boot Development

In de cursus Spring Boot Development leer je op een snelle en efficiënte manier applicaties en microservices te ontwikkelen met Spring Boot. In de training worden essentiële kenmerken van Spring Boot behandeld zoals opiniated Spring Boot Starters, embedded servers, automatische configuratie, metrics en externalized configuratie.

Spring Intro

De cursus gaat van start met een overzicht de belangrijkste Spring principes zoals het laden van beans in de bean container en dependency injection.

Spring Boot

Vervolgens wordt behandeld hoe in Spring Boot voorgedefinieerde configuraties fungeren als startpunt voor een Spring Boot applicatie. Ook wordt ingegaan op andere hoofdcomponenten van Spring Boot zoals de Autoconfigurator, de Actuator en het Command Line Interface (CLI).

Dependency Injection

Uitgebreid wordt ingegaan op dependency injection met de bijbehorende annotaties zoals @Component, @Qualifier, @Repository and @Service. De interne werking van dependency injection met Java Reflection wordt toegelicht.

Application Configuration

Onderdeel van het cursus programma zijn ook de auto configuratie mogelijkheden in Spring Boot. Componenten kunnen met annotaties als @EnableAutoConfiguration mits geregistreerd in met @Configuration geannoteerde classes aan elkaar worden gelinkt.

Aspect Orientation

En ook wordt Aspect Orientation in Spring Boot aan de orde gesteld. Crosscutting concerns in een applicatie zoals security of profiling kunnen in zogeheten aspects worden opgenomen zonder de main program flow te verstoren.

Spring JDBC en Spring Data

Database benadering vanuit Spring Boot applicaties komt aan bod bij de behandeling van Spring JDBC en Spring Data. Diverse Spring templates die boilerplate code helpen voorkomen zoals jdbcTemplate en MongoTemplate evenals JPA repositories worden besproken.

Spring REST

Spring Boot is bij uitstek geschikt om met compacte code REST API's te benaderen. Ingegaan wordt op de verschillende annotaties die hierbij van belang zijn zoals @RestController, @ResponseStatus en @JsonView.

Spring Extensions

Tenslotte komen diverse Spring extensie projecten aan de orde zoals Spring Security en Spring Cloud.

Modules Cursus Spring Boot Development

Module 1 : Spring Core	Module 2 : Spring Boot	Module 3 : Dependency Injection
Spring Framework Overview Spring Configuration Spring Dependency Injection Non IoC versus IoC Application Context Beans Life Cycle XML Configuration Configuration with Annotations Component Scanning Spring Java Configuration Aware Interfaces	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	Non-IoC or Dependency Injection Benefits of Dependency Injection Constructor Dependency Injection Setter Dependency Injection Autowiring with @Autowired @Qualifier Annotation @Component Annotation @Repository and @Service Bean scopes Event Handling Internationalization
Module 4 : Application Configuration	Module 5 : Aspect Orientation	Module 6 : Spring JDBC
Configuration Classes @Configuration Annotation @Bean Annotation @Enable Annotations @EnableAutoConfiguration Autowiring and Component Scanning @EnableScheduling Wire External Values Spring Expression Language @Value Annotation @PropertySource Annotation	What is AOP? The need for AOP Crosscutting Concerns Traditional Approach Spring AOP AOP Concepts AOP Key Terms Aspects and Weaving Pointcuts and Joinpoints ProxyFactoryBean Spring AOP Configuration	Spring and JDBC JDBC Architecture 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
Module 7 : Spring Data	Module 8 : Spring REST	Module 9 : Spring Security
What is Spring Data? Spring Data Configuration CRUD Out of the Box JPA Repositories Persisting and Modifying Entities Spring Data Queries @Query Annotation Named and Async Queries Paging Results Transaction Handling @Transactional Annotation MongoDB Template Mapping and Inserting Documents	REST Web Services @RestController HttpEntity and ResponseEntity Default Content Types Default Status Codes @ResponseStatus and HttpStatus Working with XML and JSON Multiple Representations Filtering with @JsonView REST Clients RestTemplate Sending HTTP Requests Reading Responses	Spring Security Model Process Behind Security Interceptors Authentication Manager Configuring authentication Intercepting URLs Security at the method level Access Decision Manager Security Based on Roles Security Based on Identity Access Denied Handling Securing REST Services JSON Web Tokens OAuth2 Authentication
Module 10 : Spring Cloud		
What is Spring Cloud? Spring Cloud Config Eureka Service Spring Cloud Bus Spring Cloud Cluster Spring Cloud Security Spring Cloud Data Flow Spring Cloud Connectors Spring Cloud Task App Starters Spring Cloud Zookeeper Spring Cloud CLI Spring Cloud Gateway Spring Cloud Pipelines		