# Java Development with Spring

### Audience Java Development with Spring Course

Experienced Java developers who want to use the Spring Framework for Java Desktop, Web and Enterprise applications.

### Prerequisites Course Java Development with Spring

Experience with Java programming and object orientation is required to participate in this course. Basic knowledge of Web Applications and XML is beneficial for a proper understanding.

### Realization Training Java Development with Spring

The concepts are treated by means of presentation slides and demos. The theory is interspersed with exercises. The course material is in English. The course times are from 9.30 up and to 16.30.

### Certification Java Development Spring

Participants receive an official certificate Java Development with Spring after successful completion of the course.

# Content Course Java Development with Spring

The course Java Development with Spring covers the concepts, components and architecture of the Spring Framework. The course is done with the latest Spring version with Spring Boot. In the course, annotations are central to the configuration of the Spring Applications.

### Spring Intro

The course Java Development with Spring starts with an overview of the different parts of Spring and the types of applications in which Spring can be used.

### Spring Boot

Next the essentials of Spring Boot, which greatly simplifies the configuration of Spring applications, are discussed. The main Spring Boot annotations, the opiniated defaults and the built-in web servers are treated.

### Dependency Injection

Attention is paid to the concept of Dependency Injection or Inversion of Control which plays a central role in the framework. Different variants of dependency injection such as setter injection and constructor injection are discussed.

### Aspect Oriented Programming

The other pillar of the framework, Aspect Orientation, is also covered in detail. The concepts of Aspect Orientation such as Aspects, Joinpoints, Pointcuts, Advice and Weaving are explained.

### Data Access

Next the different options for storing the data of Spring Java applications in databases are discussed. Attention is paid to the use of JDBC with a JdbcTemplate, as well as the use of Object Relational Mapping frameworks such as Hibernate with a HibernateTemplate or JPA via annotations. Spring Transactions are also treated in this context.

### Spring MVC

Web Applications with the Spring MVC Framework are part of the program of the course Java Development with Spring as well as the creation and use of Spring Rest Web Services. The role of controllers, views, page parameters and command objects is explained.

### Spring Security

Finally attention is paid to Spring and Security and the simplified setup of a Spring configuration with Spring Boot is treated. The modules Spring with JMS and Spring with JMX are optional.

# Modules Course Java Development with Spring

| Module 1 : Spring Introduction | Module 2 : Spring Boot | Module 3 : Dependency Injection |
|---|---|---|
| Spring Mission Statement | What is Spring Boot? | Benefits of DI |
| Spring Addressing Layers | Advantages Spring Boot | BeanFactory |
| Spring Characteristics | Spring Boot Flavors | Bean Configuration File |
| Spring Jars | Key Spring Boot Components | Application Context |
| Core Container Packages | Spring Boot Starter | Setter Injection |
| Spring AOP Packages | Starter Dependencies | Constructor Injection |
| Data Access Packages | Spring Boot Autoconfigurator | Beans Life Cycle |
| Spring Web Packages | @SpringBootApplication | Injection Simple Values |
| Spring Versions | Spring Boot CLI | Autowiring |
| Spring Projects | Spring Boot Internals | Aware Interfaces |
| Spring Configuration | Spring Boot Actuator | ResourceLoader |

| Module 4 : Application Configuration | Module 5 : Aspect Orientation | Module 6 : Spring Persistence |
|---|---|---|
| XML Configuration with Namespaces | The need for AOP | Java Persistence |
| Best Practices XML Configuration | Crosscutting Concerns | Traditional Persistence |
| Bean Definition Inheritance | AOP Key Terms | Transparant Persistence |
| Spring XML Namespaces | ProxyFactoryBean | Shared Persistence Concepts |
| Component Scanning | Spring AOP Configuration | DAO Design Pattern |
| Configuration with Annotations | Aspects and Advices | DAO Pattern JDBC |
| Context Namespace | Join Points and Pointcut | Integration with IoC |
| Spring Expression Language | AspectJ Configuration | DAO Portability |
| Autowiring | Advice Arguments | Spring DAO Concepts |
| Spring Java Configuration | Programmatic Proxies | Transaction Management |
| Spring Expression Language | Regexp Pointcut | Spring Exceptions |
| PropertyPlaceHolder Configuration | NameMatchMethod | Exception Translation |

| Module 7 : Spring JDBC | Module 8 : Spring ORM | Module 9 : Transactions |
|---|---|---|
| JDBC Architecture | Spring and Hibernate Integration | Using Spring Transactions |
| Executing Statements | Mapping Classes | Transaction Strategy |
| JDBC Drivers and URL's | HibernateTemplate | TransactionDefinition |
| Spring JDBC Data Access | Implementation HibernateTemplate | TransactionStatus |
| Spring DAO with JdbcTemplate | HibernateTemplate execute | Injecting Implementations |
| Data Source Injection | Hibernate Annotations | Declarative Transactions |
| Querying using JdbcTemplate | Spring and JPA | Transactional Methods |
| RowMapper | Using JPA API | Transaction Configuration |
| Querying and Populating Objects | Persistence Unit Configuration | Rollback Rules |
| Updating with JdbcTemplate | Container Managed Transactions | Using @Transactional |
| ResultsetExtractor | Externalizing DB Properties | Transaction Propagation |
| Callbacks | Entity Manager from JNDI | Programmatic Transactions |
| NamedParameterJdbcTemplate | JPA Java Configuration | Using TransactionTemplate |

| Module 10 : Spring REST | Module 11 : Spring and Security | Module 12 : Spring JMS |
|---|---|---|
| @RestController | Spring Security | Messaging Characteristics |
| HttpEntity and ResponseEntity | Security Concerns | Messaging Models |
| Default Content Types | Spring Authentication | JMS Architectural Components |
| Default Status Codes | Spring Security Modules | Spring JMS |
| @ResponseStatus and HttpStatus | Spring Security Elements | JMS with Spring |
| XML and JSON | Defining Allowed Users | Spring JMS Options |
| Multiple Representations | Form Authentication | Spring JmsTemplate |
| Filtering with @JsonView | Authentication Managers | Send and Convert |
| RestTemplate | Security Interceptor | Execute Method |
| Sending HTTP Requests | Security Filters | ProducerCalback |
| Translating Entities | Access Decision Manager | Receive and Convert |
| Reading Responses | Web Authorization | Message Listener Containers |
| Error Handlers | Method Authorization | MessageListenerAdapter |