Fernkurs Spring Boot 3

Dieser Spring Boot Fernkurs bietet eine kompakte und professionelle Einführung in die Entwicklung von Services mit dem Spring Boot 2.x Framework.

Information

Kurscode: FSBRDies ist ein Fernkurs ohne zeitliche und örtliche Präsenz..

Ablauf Fernkurs

Dieser Fernkurs deckt die gleichen Themen wie der Standard Spring Boot 3 Kurs ab. Sie erhalten die gleichen Unterlagen elektronisch und arbeiten den Kurs im Selbststudium nach eigenem Tempo und ermessen durch. Für die Übungen erhalten Sie von uns ein Ubuntu VM Ware Image. Für Fragen stehen wir per EMail zur Verfügung und erhalten damit direkten professionellen Support.Der publizierte Termin dient rein organisatorischen Zwecken.

Einleitung

Das Spring Framework bietet die geniale Infrastruktur für die Entwicklung von Enterprise Java Anwendungen.Mit Spring Boot sind diese Anwendungen eigenständig lauffähig per Konvention oder Konfiguration und dies ohne XML-Konfiguration.Spring Boot besteht aus dem Spring Framework, dem Embedded HTTP Server wie z.B. Tomcat und vielen vordefinierten Konfigurationen.Durch den Standalone Running Ansatz werden Spring Boot Anwendungen gerne für REST Services via Docker Runtime eingesetzt und betrieben.REST Services sind vielfältig in Java implementierbar und basieren auf dem JSON (Javascript Object Notation) Protokoll und dem Servlet API und damit Spring MVC.Generische Ansätze vereinfachen die Architektur und bieten einheitliche Endpoints.Mit dem Spring Framework Version 5.x bietet Spring Webflux den Support für Non-Blocking REST Services basierend auf reaktiven Streams und Mono, Flux.Dieser Kurs zeigt Ihnen pragmatisch auf wie man Spring Boot REST Services basieren auf Spring MVC und dem Servlet API programmiert, testet und ausführt.Weiter betrachten wir die Entwicklung der asynchronen und damit reaktiven REST Services und vergleichen beide Techniken hinsichtlich der Vor- und Nachteile.

Ihr Nutzen

Die Kursteilnehmer sind in der LageSpring Boot Anwendungen zu verstehen und programmieren.REST Services mit dem Spring Boot Framework und Spring MVC entwickeln.Datenbank Integration via Spring Data und JPA.Unit Testen von Spring Boot Anwendungen.Sie kennen generische Ansätze für die einheitliche REST Service Entwicklung.Reaktive Non-Blocking REST Services mit Spring Webflux entwickeln, testen und anwenden.

Verwandte Kurse

- Angular Material Design
- Spring Boot
- Spring Boot / Angular Material Design

Voraussetzungen

Gute Grundkenntnisse von Java analog den Kursen JEGL (Java Einführung) und JPF2 (Java Vertiefung).

Teilnehmerkreis

Java Entwickler oder Projektleiter, welche das Spring Boot Framework in ihren Projekten einsetzen.

Unterlagen

- Tutorial

- Code Walks
- Internet / Intranet

Inhalt

- Einführung
  - Was ist Spring
  - Was ist Spring Boot
  - Was sind REST Services
  - Javascript Object Notation (JSON)
  - Maven vs. Gradle
  - Was ist YAML
  - Spring Boot Setup
- Jumpstart
- Spring Boot Konfiguration
  - Autoconfig, Dependency Injection & Profiles
  - YAML und Property Dateien
- Spring REST Services
  - Rest Controller
  - Request Mapping
  - Http Methods GET, POST, PUT, DELETE
  - RESTful Services
  - Rest Versionierung
  - Generic Rest Controller
  - JSON Request/Response
  - HTTP Message Converters
  - Data Transfer Object (DTO)
  - @Bean vs @Autowired
  - @Scope Rules
  - @Component vs @Controller vs @Service
  - Service Components
  - Interceptor, ExceptionHandler, @ControllerAdvice
- Spring Data und JPA
  - Overview Java Persistence API
  - JPA Repositories
  - Spel Spring Expression Language
  - Query Methods
  - Custom Repositories mit QueryDSL
- Spring Backend
  - Scheduling and Tasks
  - fixedRate, fixedRateString, fixedDelay, fixedDelayString, initialDelay, initialDelayString, cron
  - Scheduling Pool
  - @EnableScheduling, @Scheduled
  - @EnableAsync, @Async
  - Configuration and Condition
  - JMS Messaging mit ActiveMQ
- Spring Caching
  - Caching Strategien
  - Default Cache Manager
  - Enable Caching (@Cachable, @CacheEvict, @CachePut, @CacheConfig)
  - Caffeine Cache Manager
  - Self Autowired
- Context / Events
  - ApplicationContext
  - Custom Context
  - Repository Context
  - ApplicationEvents
- Spring Test
  - Spring Unit Tests
  - Spring Boot Test Slices
  - Integrationstests mit @DataJpaTest

  - Integrationstests mit @SpringBootTest
  - Mocking mit @MockBean
  - WebLayer Tests mit @WebMcvTest und MockMvc (Mockito)
- Logging
  - Logback Extensions
  - logback-spring.xml
- SOAP Web Services
  - Contract First (XML Schema)
  - JAXB Generator
  - SOAP Web Service Endpoint
  - SOAP Web Service Configuration Beans
  - SOAP Request Test
- Spring Webflux
  - Reactive REST Services
  - Blocking vs Non-Blocking I/O
  - Reactive API, Mono und Flux
  - Reactive JumpStart
  - Backpressure
  - Reactive Spring Data
  - Reactive Repositories und MongoDB
  - Webflux.fn Functional Endpoints
- Spring Security
  - Auto Configuration mit @EnableWebSecurity
  - WebSecurityConfigurerAdapter
  - Method Level Security mit @PreAuthorize
  - OAuth2, OpenID Connect (OIDC)
- Actuator Endpoints
  - Spring Actuator
  - /health, /info, /metrics, ...
  - Spring Actuator Info Endpoint
  - Custom Info Endpoint InfoContributor
- Spring Admin Server
  - Setup Spring Admin Server
  - @EnableAdminServer
  - Spring Boot Admin Client
  - Secured Spring Admin Server
- Diverses
  - Spring Cloud
  - OpenFeign Client
  - Spring Batch
  - Spring Integration
  - ...

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz