

Spring Boot Eğitimi

Açıklama

Günümüz yazılım dünyasında, hızlı, ölçeklenebilir ve güvenli uygulamalar geliştirmek için Spring Boot en popüler çözümlerden biri haline gelmiştir. Bu kapsamlı eğitimde, Spring Boot'un temel bileşenlerinden microservice mimarisine kadar adım adım öğrenerek kurumsal düzeyde modern uygulamalar geliştirmeyi öğreneceksiniz.

- **Spring Boot Temelleri:** Proje yapısı, IOC, DI, Restful API, hata yönetimi
- **Veri Yönetimi:** Spring Data JPA, entity ilişkileri, DAO - DTO dönüşümleri
- **Güvenlik:** JWT Authentication, Spring Security
- **Test & Debugging:** Mockito, MockMVC, hata ayıklama teknikleri
- **Docker & Kafka:** Docker entegrasyonu, Kafka ile event-driven mimari
- **Microservice Mimarisi:** Spring Cloud, Eureka, Feign Client, Circuit Breaker, API Gateway

Bu eğitim, Spring Boot'un **yalnızca teorik yönünü değil, aynı zamanda pratik uygulamalarını da** kapsayarak gerçek dünya senaryolarına uygun çözümler geliştirme becerisi kazandıracaktır. **Hands-on** egzersizler ve **canlı demo uygulamaları** ile öğrendiklerinizi pekiştireceksiniz.

Kimler Katılmalı?

- Spring Boot ile ölçeklenebilir uygulamalar geliştirmek isteyen yazılım geliştiriciler
- Backend teknolojilerine hâkim olup Spring ekosistemine geçiş yapmak isteyenler
- Microservice mimarisi ve dağıtık sistemleri anlamak isteyen mühendisler
- Kurumsal düzeyde Spring Boot projeleri geliştirmeyi hedefleyen profesyoneller

Eğitim İçeriği

Spring Boot Projesi Oluşturma

- start.spring.io arayüzü
- Proje oluşturma parametleri

Spring Boot Proje Yapısı

- Spring Framework dünyasına giriş
 - Application context
 - IOC / DI kavramları
 - Spring Bean tanımları
- Proje paket yapısı
- Uygulama katmanları
- Katmanlar arası iletişim

Rest API

- Restful servis standartları
- Response ve Request yapıları
- Validation ve Custom Validation tanımları

OpenAPI ile RestAPI standardizasyonu

- Swagger UI
- Controller interface
- Request/Response generation

Hata ayıklama (Exception Handling)

- Hata ayıklama standartları
- Validation errors
- Error messages

Servis katmanı

- Service katmanı mimarisi
- Servisler arası iletişim ve bean injection

- Servis - veritabanı iletişimi

Spring JPA ve Data Entities

- Neden Lombok Kullanmıyoruz!?
- Spring Data JPA repository çeşitleri ve tanımlamaları
- Spring Data Entity tanımları
- Spring Data Entity ilişki tanımları
 - 1-to-1
 - 1-to-Many
 - Many-to-Many
- BaseEntity tanımlamaları

Dao - Dto dönüşümleri

- Conversion metodlar
- MapStruct

SpringBoot ile REST API tasarım desenleri

- Strategy Tasarım Deseni
- Singleton Tasarım Deseni
- Enum Type Factory Method Deseni

Veritabanı migration

- Flyway db migration kullanımı

Mockito Framework

- Mock kullanımı
- MockMVC ile controller testleri
- WireMock kullanımı

Spring Security

- InMemory Authentication
- Basic Authentication
- JWT Token Authentication ve Authorization

Spring Boot Uygulaması Docker Entegrasyonu

- Dockerfile oluşturulması
- Docker image oluşturulması ve çalıştırılması
- hub.docker.com'a image yüklenmesi ve çalıştırılması
- Dockercompose dosyasının oluşturulması

SpringBoot ile Apache Kafka entegrasyonu

- Event-Driven-Architecture tanımı
- SpringBoot - Kafka entegrasyonu

SpringBoot ile Cache entegrasyonu

- @Cacheable, @CacheEvict, @CachePut
- Springboot ile Redis kullanımı

Microservislere giriş

- Microservis ne değildir?

Microservisler arası iletişim

- Spring Cloud Eureka Server/Client ve Feign Client
- Resilience4J ile hata toleransı
- Circuit Breaker, Bulkhead ve Rate Limiting

Dağıtık Sistemler

- API Gateway
- Spring Actuator
- OpenTelemetry ile Merkezi loglama

Microservis konfigürasyonları

- Spring Cloud Config
- Centralized Configuration
- Vault Integration

Microservisler arası iletişim 2

- gRPC ile Microservice İletişimi

Ön Koşullar

Temel Java ve framework bilgisine sahip olmak