

Spring 5 ve JPA2'ye Giriş Eğitimi

Eğitim Hakkında

Bu Eğitim, katılımcılara bağımlılık yönetimi için Spring Boot'u nasıl kullanacaklarını ve Spring Data kullanarak JPA tabanlı depolar oluşturmayı öğretir. Katılımcılar üç ana yapılandırma stilini öğrenirler: Java tabanlı (@ Yapılandırma), açıklama tabanlı (@ Bileşen) ve geleneksel XML tabanlı yapılandırma.

Neler Öğreneceksiniz

Uygulama nesnelere (fasulye) birlikte yapılandırmak ve bağlamak için Yay Çekirdek modülünü ve DI'yi kullanmayı,
Farklı meta veri türlerini (XML, ek açıklamalar / @ Bileşen ve Java Yapılandırması / @ Yapılandırma) ve bunların nasıl ve ne zaman kullanılacağını,
Yaşam döngüsü olayları, fasulye kapsamı ve Spring API gibi Core modülünün tüm özelliklerini,
Bağımlılık yönetimini ve yapılandırmayı basitleştirmek için Spring Boot kullanmayı,
JPA ve JPA Mimarisinin faydalarını,
JPA tabanlı uygulamalar oluşturmayı,
Spring'i JPA ile entegre etmek için ORM (Object-Relational Mapping) modülüyle çalışmayı,
Kullanımı kolay Java açıklama desteği ve tx / aop XML yapılandırma öğeleri dahil olmak üzere Spring'in işlem desteğini,
Kalıcı nesnelere veritabanına eşlemek için JPA eşlemeyi,
JPA sorguları ve JPQL ile çalışmayı,
Koleksiyonları ve ilişkilendirmelerini, (Değer ve varlık türleri, tek yönlü, çift yönlü, 1-1, 1-N, NN
JPA'nın sürüm oluşturma desteğini kullanmayı,
JPA kullanarak devralma hiyerarşilerini eşlemeyi,
Spring / JPA'yı Java EE Web uygulamalarıyla entegre etmeyi,
Otomatik olarak oluşturulan sorgularla otomatik olarak JPA tabanlı depolar oluşturmak için Spring Data'yı kullanmayı öğrenebilirsiniz.

Eğitim İçeriği

Introduction to Spring

Overview of Spring Technology
Motivation for Spring, Spring Architecture
The Spring Framework
Spring Introduction
Declaring and Managing Beans
ApplicationContexts - The Spring Container
XML and @Component/@Named Config
Dependencies and Dependency Injection (DI)
Examining Dependencies
Dependency Inversion / Dependency Injection (DI)
DI in Spring - XML and @Autowired

Configuration in Depth

- Java Based Configuration (@Configuration)
- Overview, @Configuration, @Bean
- Dependency Injection
- Resolving Dependencies
- Integrating Configuration Types
- XML and @Component Pros/Cons
- @Configuration Pros/Cons
- Choosing a Configuration Style
- Integrating with @Import and <import>
- Bean Scope and Lifecycle
- Singleton, Prototype, and Other Scopes
- Configuring Scope
- Bean Lifecycle / Callbacks
- Externalizing Properties
- Properties Files
- @PropertySource, property-placeholder
- Using @Value
- SpEL
- Profiles
- Overview and Configuration
- Activating Profiles

Spring Boot Overview

- maven and Spring
- Spring Boot Structure
- Spring POMs with Boot Parents
- Spring Boot Starters
- Other Capabilities
- Spring Testing
- Testing and JUnit Overview
- Writing Tests - Test Classes, asserts, Naming Conventions
- Running Tests - IDE, maven, ...
- Test Fixtures - setup and teardown
- Spring TestContext Framework
- Overview
- Configuration
- Running Tests

Introduction to Java Persistence API (JPA2)

- Overview
- Persistence Layers, Object-Relational Mapping (ORM), JDBC
- JPA Overview
- Mapping with JPA
- Entities and @Entity, ids and @Id,
- Generated Id Values

- Basic Mapping Types
- Persistence Unit and EntityManager
- Persisting to the DB, the EntityManager API
- Persistence Units, Config, Persistence Context
- Retrieving Persistent Entities with find()
- More About Mappings
- Default Mappings, @Basic, @Column
- Field vs. Property Access
- Temporal (Date/Time) Mappings
- Java 8 Data/Time Mapping
- equals() and hashCode()
- Logging Options (Provider based)

Spring/JPA Integration

- Spring's DataSource Support
- Managing the EntityManager (EM)
- LocalContainerEntityManagerFactoryBean and Container-managed EMs
- JEE and JNDI Lookup of the EM
- Configuration and Vendor Adaptors
- Creating a JPA Repository/DAO Bean - @PersistenceUnit, @PersistenceContext

JPA Updates and Queries

- Inserting Updating, and Deleting Entities
- Querying and JPQL
- Entity Based Queries, SELECT, WHERE
- Query Interface, Executing Queries, Generic Queries (JPA 2)
- JPQL Operators, Expressions, and Parameters
- Named Queries
- Additional Query Capabilities - Projection and Aggregate Query, Embedded Objects

Transactions

- TX Overview and JPA Transactions
- Spring's Declarative TX Management (REQUIRED, etc.)
- @Transactional
- TX Scope and Propagation
- Pointcut-based Configuration of Transactions

The JPA Persistence Lifecycle

- The Persistence Lifecycle
- JPA Entity States (New, Managed, Detached, Removed), and Entity State Diagram
- Persistence Context - Lifespan, Propagation
- Synchronization to the DB
- Versioning and Optimistic Locking
- Overview, Detached Instances
- Versioning, @Version, Optimistic Locking
- Lifecycle Callbacks

@PrePersist, @PostPersist, etc.
Entity Listeners, @EntityListeners

Relationships

Relationships Overview: Object Relationships, Participants, Roles, Directionality, Cardinality
Relationship Mapping
Mapping Overview (1-1, 1-N, N-1, N-N)
Unidirectional and Bidirectional
@ManyToOne, @OneToMany, @ManyToMany, @OneToOne with Table Structures
Relationship Inverse - Owning Side
Collection Types (List, Set, etc)
Cascading, Lazy and Eager Loading
Queries Across Relationships (Inner Joins, Outer Joins, Fetch Joins)
Entity Inheritance Mapping
Overview
Single Table, Joined (Table per Subclass), Table per Concrete Class Mappings
Pros and Cons of Mapping Strategies

Spring Web Integration

Integrating Spring with Java EE Web Apps
ContextLoaderListener
WebApplicationContext
Using Spring beans in Web app controller logic
Open EntityManager in View
Lazy Loading Issue in Web Apps
Open EntityManager in View Pattern
Using Spring's OpenEntityManagerInViewFilter/Interceptor

Spring Data Introduction

Overview and Architecture
Configuring Spring Data
Repositories and JPA Repositories
Using CrudRepository
Using Spring Data
Naming Conventions for Querying
Creating more Complex Queries
Query Configuration

Additional Topics

Spring 5: Core Updates
JPA: Embedded Objects
JPA: Compound Primary Keys