Web Application Development Using Spring, Hibernate and JPA Eğitimi

Açıklama

Bu eğitim katılımcıların, JPA (Java Persistence API), Spring ve Hibernate açık kaynak kodu çerçevelerinin yanı sıra Web Hizmetleri ve Ajax hakkında kapsamlı bilgiler edinmelerini sağlar. Hem Spring3/Hibernate3 hem de Spring4/Hibernate4 için uygun olan bu eğitim, ana Spring ve Hibernate yeteneklerinin yanı sıra Spring tarafından sağlanan entegrasyon yeteneklerini de kapsar.

Bu eğitim, Spring4'ün desteklediği birçok yeni ve güçlü yeteneği kullanmaya yönelik teknikler hakkında genel bilgiler verir. Üç ana yapılandırma tarzını (@Configuration, @Component, XML) ve bunların kullanımı için gerekli yönergeleri kapsar. Eğitim ayrıca, JDBC desteği, Hibernate, Spring bildirim işlemleri gibi kalıcı çerçeveler ve Spring'in JEE Web teknolojileriyle entegrasyonu gibi gelişmiş yetenekleri de kapsar.

Eğitim, Hibernate açık kaynak nesne/ilişkisel kalıcılığının ve Java sorgu hizmetinin tüm önemli yeteneklerini kapsar. Eğitim ayrıca Java'da kalıcı sınıflar geliştirmenin yanı sıra ilişkilendirmeler/ilişkiler, kalıtım, çok biçimlilik, kompozisyon ve koleksiyonların kullanımı da kapsar. JPA Bildirimleri ve JPQL (Java Persistence Query Language dahil olmak üzere JPA'nın (Java Persistence API) temelleri de eğitimde yer alır.

Öğrenmeyi güçlendirmek ve gerçek yetkinliği geliştirmek üzere kapsamlı uygulama örnekleri eğitimle bütünleştirilmiştir. Katılımcılar eğitimde, tüm önemli geliştirme ortamları için geçerli bir bilgi birikimi temeli oluşturan Eclipse IDE'yi kullanan Spring/Hibernate uygulamaları oluştururlar.

Bu eğitimde neler öğreneceksiniz?

- Spring ve Dependency Injection (DI) / Kontrolün Tersine Çevrilmesi ile ilgili temel ilkeler

- Uygulama nesnelerini (beans) birbirine bağlamak ve yapılandırmak için Spring Core modülünü ve DI'yi kullanma
- Farklı meta veri türlerini (XML, @Component ve @Configuration) kullanma
- Core modülünün tüm yetenekleri
- Spring'i Hibernate veya JPA gibi teknolojilere entegre etmek için ORM (Object-Relational Mapping) modülünü kullanma
- Spring işlem desteği
- Spring'in Java EE Web uygulamalarına entegrasyonu
- Hibernate'in özellikleri ve sağladığı avantajlar
- Hibernate çerçevesini kullanarak uygulama geliştirme
- Hibernate protokollerini kullanarak kodları yapılandırma
- Kalıcı nesneleri veritabanına eşlemek için Hibernate Eşleme'yi kullanma
- Koleksiyonlar ve ilişkilendirmelerle çalışma
- Hibernate'in sürüm belirleme desteğini kullanma
- Kalıtım hiyerarşilerini Hibernate kullanarak eşleme
- Hibernate Queries, HQL ve Criteria ile çalışma
- Hibernate işlem desteği
- Hibernate ve Java Persistence API (JPA) arasındaki ilişki
- JPA2 kullanarak tasarım yapma ve kodlama
- AJAX entegrasyonunu kullanma
- Web hizmetlerini kullanma

Ön Koşullar

Java SE programlama deneyimi ve nesneye dayalı tasarım ilkeleri hakkında bilgi sahibi olunması gereklidir. Temel XML, HTML ve JavaScript temel bilgileri faydalıdır ancak gerekli değildir.

Eğitim İçeriği

Introduction to Spring

- Overview of Spring Technology
    - Challenges for Modern Applications
    - Motivation for Spring
    - Spring Architecture
    - The Spring Framework API
- Spring Fundamentals
    - Managing Beans
    - Inversion of Control/IoC
    - Dependency Injection/DI
    - Configuration Metadata
        - Configuring Beans (XML)
- The Spring Container
    - Function of the Spring Container
    - ApplicationContext Overview
        - ClassPathXmlApplicationContext
        - FileSystemXmlApplicationContext
        - AnnotationConfigApplicationContext
    - API and Usage
- Dependencies and Dependency Injection (DI)
    - Examining Dependencies
    - Dependency Inversion
    - Dependency Injection (DI) in Spring
        - Basic Configuration and Usage

Wiring in Depth

- Value Injection
    - Configuring Value Properties
        - Property Conversions
    - Externalizing Values in Properties Files
- Constructor Injection
    - Constructor Injection Overview
    - Configuration – @Configuration and XML
    - p: and c: Namespaces for XML Configuration
- Qualifiers and Domain Specific Language (DSL)
    - Limitations of Autowiring

    - Qualifiers and DSL
    - Creating and Using an Annotation-Based DSL for Bean Configuration
    - Benefits of Qualifiers for Bean Configuration
- Profiles
    - Profiles Overview
    - Configuring Profiles (XML and @Configuration)
    - Activating Profiles
- Introduction to SpEL

Spring/Hibernate Integration

- Overview of Spring Database Support
- Configuring a DataSource
- Using Contextual Sessions
    - Spring/Hibernate Configuration with LocalSessionFactoryBean and SessionFactory
    - Creating Spring/Hibernate DAO Classes Using Contextual Sessions
    - Overview of Template Approach

Transaction (TX) Management

- Hibernate Transaction Management
    - Transaction Overview and Transactions in Hibernate
    - Hibernate Transaction API (in Managed and Non-Managed Environments)
- Intro to Spring Transaction Management
    - Spring Transaction Managers
    - Spring Declarative TX Management
    - Spring TX Scope and Propagation
    - Spring TX Attributes (REQUIRED, SUPPORTS, etc)
- XML Configuration of Spring Transactions
    - Specifying Advice, TX Attributes, and Methods
    - Linking Advice with Pointcuts
    - Benefits of XML Configuration of TX Behavior

Relationships

- Object Relationship Overview
- Mapping Collections of Value Objects

- Entity Relationships: 1-N, N-1, N-N, 1-1
- Mapping Entity Relationships
- Uni and Bi-Directional Relationships
- The Relationship “Inverse”
- Cascading Over Relationships
- Queries Across Relationships
    - Lazy Loading vs. Eager Loading
- Inheritance Mapping
    - Entity Inheritance with Hibernate
    - Table-per-Class Mapping
    - Table-per-Subclass Mapping
    - Table-per-Concrete Class Mapping

Hibernate Additional Topics

- Components and Multi-Table Mapping
- equals() and hashCode()
- Caching and Efficiency
- Design Considerations

Introduction to AJAX

- AJAX Architecture and Capabilities
    - Client Side
    - Server Side
- Circumventing the Page Reload Paradigm
- CSS, HTML and AJAX
- JavaScript and DOM
- XMLHttpRequest Object
    - readyState and responseXML Properties
- Making AJAX Asynchronous Calls
- AJAX Function Calling Conventions
- Response Handling with JavaScript
- Browser Compatibility Issues
- Server-Side and Client-Side AJAX

Adding AJAX to Existing Web Applications

- Issues Adding AJAX to Existing Apps
- Dealing with Asynchronous Responses
- Cross Browser Libraries and Frameworks
- Working with AJAX Toolkits
    - Dojo Toolkit
    - Prototype
    - DWR (Direct Web Remoting)
    - Google Web Toolkit
- AJAX and REST Design Compatibility Issues
- Security Issues

Web Services on the Client Side

- Consuming a Web Service
- Client Side Artifacts
- JAX-WS Clients
- Java/WSDL Mapping
- RESTful Web Services
- RESTful Web Services in Java

Configuration in Depth

- Annotation Driven Configuration
    - JSR 330 (@Named) and Spring (@Component) annotation styles
    - @Named/@Component, @Inject/@Autowired, @Repository, @Service
    - Configuring Beans and Autowiring with Annotations
    - Enabling Annotations – context:component-scan
    - Annotation Pros and Cons
- Java Based Configuration (@Configuration)
    - Overview of Code-Centric Configuration
    - @Configuration and @Bean
    - Dependency Injection
    - Resolving Dependencies on Other Beans
        - Injecting Configuration Classes
    - Pros and Cons
- Integrating Configuration Types
    - Choosing a Configuration Style
    - Integrating Configuration Styles

    - Importing: @Import
    - Scanning with @Configuration Style
- Bean Scope and Lifecycle
    - Bean Scope Defined
        - Singleton, Prototype and Other Scopes
    - Configuring Scope
    - Bean Creation Lifecycle
        - Lifecycle Callbacks
    - BeanPostProcessor and Event Handling

Introduction to Hibernate

- Issues with Persistence Layers and Object-Relational Mapping (ORM)
- Hibernate Overview and Benefits
- Hibernate Architecture
- Configuring Hibernate
    - hibernate.cfg.xml File
    - Connection Properties
    - Database Dialect
    - SessionFactory
    - Configuration and Session
- Mapping a Class
    - Persistent Entity Class
    - Hibernate Mapping File
    - Mapping the Entity Class
    - Primary keys: ID property, Generated ID
    - Hibernate Type System
- Working with Sessions and Persistent Objects
- Logging: hibernate.show_sql,
- log4j Overview and Configuration for Hibernate

Updates and Queries

- Inserting, Updating and Deleting Entities
- HQL – Hibernate Query Language Fundamentals
- The Query Interface
- Creating and Working with Queries
    - Named Queries

    - Projection Queries
    - Aggregate Queries

The Hibernate Persistence Lifecycle

- The Lifecycle of Managed Objects
- Persistent, Transient, and Detached Objects
- The Persistence (Session) Context (Lifespan, Relation to Managed Objects, Propagation)
- Contextual Sessions
- Synchronization to the Database
- The Session as Cache
- Optimistic Locking/Versioning
    - Detached Objects and Optimistic Locking
    - Versioning Overview and Using Versioning
    - Locking Objects

Spring Web Integration

- Integrating Spring with Java EE Web Apps
    - ContextLoaderListener
    - WebApplicationContext
    - Using Spring Beans in Wep App Controller Logic

Hibernate and JPA (Java Persistence API)

- Overview of the Java Persistence API (JPA)/EJB 3
- Relationship Between JPA and Hibernate
- Mapping Entities with JPA Annotations
- The EntityManager, Persistence Context and Persistence Unit
- Working with Transactions
    - EntityTransaction, Managed and Unmanaged Environments
- Inserts and Updates
- JPQL – Java Persistence Query Language
- Versioning
- Relationships

Java and AJAX

- Callback Functions/Methods
- Built-In Objects
- Parsing AJAX Responses
- XML and JSON Response Formats
- Servlet Code to Handle AJAX Requests
- Server Side Persistence
- AJAX Patterns and Best Practices
- Limitations of AJAX
- Debugging AJAX

Web Services on the Server Side

- Overview of Web Services
- Advantages of Web Services
- Creating a Web Service
- Deploying a Web Services
- Requirements for a JAX-WS Web Service Endpoint Implementation Class
- The Web Service Endpoint Implementation Class
- Working with WSDL Files
- Web Service Interoperability