

Spring Data JPA

Audience Course Spring Data JPA

The course Spring Data JPA is intended for Java and Spring Boot developers who want to access relational databases from Spring Boot applications.

Prerequisites course Spring Data JPA

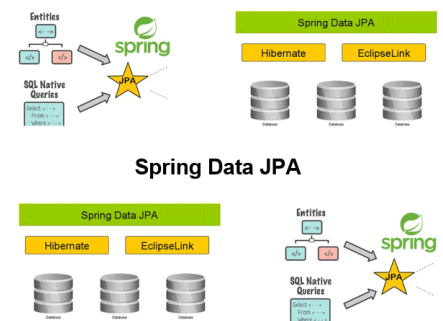
Thorough knowledge of Java and SQL and basic knowledge of Spring and Spring Boot. Experience with JPA and Hibernate is beneficial for understanding.

Realization Spring Data JPA Training

Demo sessions led by the trainer are alternated with presentations and practical exercises.

Certificate Spring Data JPA

After successfully completing the course, attendants will receive a certificate of participation in Spring Data JPA.

Duration: 3 days
Price: € 2250
[Open Schedule](#)


Content Course Spring Data JPA

The course Spring Data JPA explores the powerful Spring Data JPA abstraction on top of the Hibernate/JPA object relational mapping framework, which simplifies database interactions and query execution in Spring applications.

Persistence Intro

The course begins with an introduction to Java Persistence. Topics include JDBC, SQL, drivers, database configuration, and Object Relational Mapping. It dives into the Java Persistence API (JPA), use of the Entity Manager, Persistence Units, and mapping entities.

Spring Data JPA

This module introduces participants to Spring Data and Spring Data JPA. It covers configuration using property files, mapping tables and columns, and using primary keys. Key annotations like `@Id` and `@GeneratedValue` are introduced along with defining entity relationships such as One-to-One, One-to-Many, Many-to-One, and Many-to-Many.

CRUD Operations

This module focuses on implementing CRUD operations using `JpaRepository` and `CrudRepository`. Topics include Cascade Types, fetch strategies (EAGER vs LAZY), transaction behavior with `@Transactional`, and using the `PagingAndSortingRepository`.

Querying Data

Participants learn how to query data using method naming conventions like `findBy` and `countBy`. Both JPQL and native queries are covered. The module also includes the use of `@Query` and `@Modifying` annotations and building custom queries.

Entity Lifecycle

This module focuses on understanding the entity lifecycle in JPA. It covers save and delete methods, annotations such as `@PrePersist` and `@PostLoad`, flush modes, caching, transaction management, and optimistic locking using the `@Version` annotation.

Pagination and Sorting

Participants learn how to implement pagination and sorting using `Page`, `Slice`, and `PageRequest` interfaces. Sorting entities by specific attributes is also covered, along with using DTO projections (open, closed, and class-based).

Caching

This module covers caching strategies with JPA and Spring Cache. It explains the use of the `@Cacheable` annotation, cache configuration, optimizing fetch strategies using `@BatchSize` and `@EntityGraph`, and storing query results efficiently.

Transactions

Both declarative and programmatic transaction management are covered, including how to roll back transactions using the `rollbackFor` attribute.

Advanced Topics

The final module covers advanced topics such as domain events, event publishers and listeners.

Modules Course Spring Data JPA

Module 1: Persistence Intro	Module 2: Spring Data JPA	Module 3: CRUD Operations
Java Persistence JDBC and SQL Drivers and URL's Database Configuration Object Relational Mapping Java Persistence API Entity Manager Persistence Units Mapping Entities JPA Implementations Eclipse Link and Hibernate	What is Spring Data? Spring Data JPA Property Files Tables and Columns Primary Keys Auto Generation Strategies @Id and @GeneratedValue Persistence Context Defining Relationships One-to-One and One-to-Many Many-to-One and Many-to-Many	Implementing CRUD Operations @EnableJpaRepositories JpaRepository Cascade Types Fetch Strategies EAGER versus LAZY CrudRepository Transaction Management @Transactional Annotation @Transactional Attributes PagingAndSortingRepository
Module 4: Querying Data	Module 5: Entity Lifecycle	Module 6: Pagination and Sorting
Query Methods Method Naming Conventions Derived Query Methods findBy and countBy Java Persistence Query Language JPQL Queries @Query Annotation @Modifying Annotation Equality Conditions Custom Queries Native Queries	Understanding Entity Lifecycle Save Methods Delete Methods @PrePersist Annotation @PostLoad Annotation Flush Modes Caching Second Level Cache Managing Transactions Optimistic Locking @Version Annotation	Implementing Pagination Pageable and Sort Paginating All Entities Page and Slice Interface PageRequest Class Sorting All Entities Sort Order Typed Sort Returning DTO Projections Open and Closed Projections Class Based Projections
Module 7: Caching	Module 8: Transactions	Module 9: Advanced Topics
Spring Cache with JPA @Cacheable Annotation First Level Cache Second Level Cache Cache Configuration Storing Results Fetch Strategies Using @BatchSize @EntityGraph Annotation Optimizing Fetch Queries	ACID Properties Transaction Properties Transaction Propagation Transaction Management Injecting Transaction Manager @EnableTransactionManagement Declarative Transactions Programmatic Transactions Rolling Back Transactions rollBackFor Attribute	Domain Events Event Publishers Event Listeners @DomainEvents Annotation Auditing in Spring Data JPA @EnableJpaAuditing Annotation AuditingEntityListener Entity Graphs @EntityGraph Annotation @NamedEntityGraph Annotation