AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 09/23/2026

Sign Up

Jakarta Persistence (JPA) Training

3 days (21 hours)

Overview

Jakarta Persistence (JPA) is the leading Java specification for managing data persistence and object-relational mapping (ORM) between an application's objects and a relational database.

This Jakarta Persistence (JPA) training course will enable you to master data persistence in your Java applications through a hands-on implementation based on Hibernate ORM. You will learn how to create entities, configure their associations, and manage their lifecycle using the EntityManager and the persistence context.

You will be able to model complex relationships, manage transactions and concurrent access, and efficiently query your data using JPQL and the Criteria API. You will also learn to select the loading strategies best suited to your applications' needs.

The training will enable you to identify and correct common performance issues such as N+1 queries, unnecessary fetches, or inefficient mappings. You'll learn to analyze the SQL queries generated by Hibernate and optimize your data access using fetch strategies, batching, pagination, and caching mechanisms.

Finally, you'll learn how to migrate a legacy JPA application to Jakarta Persistence, including mastering the transition from javax.persistence.* to jakarta.persistence.* and the changes in Jakarta Persistence 3.2 to modernize your existing Java applications.

By the end of this course, you will be able to design a robust persistence layer, manage transactions and application performance, and effectively use Jakarta Persistence with Hibernate.

Like all our training courses, this one will introduce you to **the latest stable version** of the

Java Persistence API (JPA) technology (4.1.1) and its new features.

Objectives

- Understand the architecture and operation of Jakarta Persistence (JPA)
- Create and map entities and their associations to a relational database
- Master the Persistence Context, EntityManager, and the entity lifecycle
- Manage transactions and concurrent access to data
- Build queries using JPQL and the Criteria API
- Identify and resolve performance issues related to data mapping and loading
- Migrate an application using javax.persistence to Jakarta Persistence 3.2

Target Audience

- Java developers
- Backend developers
- Software engineers
- Developers using JPA or Hibernate in their applications
- Developers responsible for modernizing existing Java EE applications

Prerequisites

- Proficiency in Java programming and object-oriented programming
- Basic knowledge of SQL and relational databases
- Understanding of the principles of Java classes, objects, annotations, and collections
- Some experience with JDBC, JPA, or Hibernate is a plus but not required

Curriculum for our Jakarta Persistence (JPA) training course: Java data persistence

[Day 1 - Morning]

Understanding Jakarta Persistence and managing the entity lifecycle

- Understanding the role of Jakarta Persistence (JPA) in Java applications and its integration with Jakarta EE
- Distinguish between the Jakarta Persistence specification and its implementations, such as Hibernate ORM
- Configure a persistence unit, a data source, and the provider’s main properties
- Create entities, define their identifiers, and understand key generation strategies
- Master the Persistence Context, EntityManager, and the transient, managed, detached, and removed states

- Hands-on workshop: Configure Jakarta Persistence with Hibernate and perform your first CRUD operations on a relational database.

[Day 1 - Afternoon]

Master entity and association mapping

- Map Java properties to SQL tables, columns, constraints, and data types
- Configuring OneToOne, OneToMany, ManyToOne, and ManyToMany associations
- Mastering bidirectional relationships, cascading operations, and orphaned deletes
- Using embeddables, value collections, and composite primary keys
- Implement inheritance strategies and entity mapping across multiple relational structures
- Hands-on workshop: modeling a business domain with multiple entities, associations, and cascade rules.

[Day 2 - Morning]

Managing transactions, concurrency, and the persistence lifecycle

- Understand how transactions work and how they interact with the persistence context
- Using the EntityManager’s persist, merge, remove, find, refresh, flush, and detach operations
- Understand dirty checking and synchronization between Java objects and the database
- Master optimistic and pessimistic locking to manage concurrent access
- Use lifecycle callbacks and listeners to respond to persistence events
- Hands-on workshop: Implement transactional processing and resolve a scenario involving concurrent data modifications.

[Day 2 - Afternoon]

Querying data with JPQL and the Criteria API

- Writing queries with Jakarta Persistence Query Language (JPQL)
- Use parameters, joins, aggregations, projections, subqueries, and typed queries
- Build dynamic queries with the Criteria API
- Implement pagination and choose the appropriate query strategy for your needs
- Leverage the latest features of Jakarta Persistence 3.2, including getSingleResultOrNull() and updates to JPQL and the Criteria API
- Hands-on Workshop: Build Several JPQL and Criteria Queries and Compare Their Relevance based on different business needs.

[Day 3 - Morning]

Optimizing Performance with Jakarta Persistence and Hibernate

- Understanding LAZY and EAGER strategies and their impact on data access
- Identify and resolve the N+1 problem using fetch joins and appropriate loading strategies
- Analyze SQL queries generated by Hibernate and diagnose inefficient data access
- Optimize processing with batching, pagination, and appropriate caching mechanisms
- Identify ORM anti-patterns and resolve key issues related to detached sessions and entities
- Hands-on workshop: diagnose an application with excessive queries, then optimize its mapping and loading strategies.

[Day 3 - Afternoon]

Modernize and migrate to Jakarta Persistence 3.2

- Understand the evolution from JPA to Jakarta Persistence and the transition from javax.persistence.* to jakarta.persistence.*
- Update the dependencies, imports, configurations, and Hibernate versions of an existing application existing application
- Leverage modern Java types and prioritize `java.time` for managing timestamps
- Use the new features of Jakarta Persistence 3.2, such as Embeddable records, PersistenceConfiguration, and SchemaManager
- Test the compatibility of mappings, queries, and transactions after migration and prepare non-regression tests
- Hands-on workshop: Migrate an application using javax.persistence to Jakarta Persistence 3.2 with Hibernate and resolve any incompatibilities encountered.

Target Audience

This training is intended for both individuals and companies—large or small—seeking to train their teams in a new, advanced IT technology or to acquire specific business knowledge or modern methodologies.

Assessment upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor, supported by examples and

reflection sessions and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.