

**Introduction to the Java Persistence API 2**

**Course Number:** JAV-162  
**Duration:** 3 days

**Overview**

Accelebrate's JPA 2 Training: Introduction to the Java™ Persistence API 2 class teaches attendees how to implement JPA for data persistence. JPA 2 is a standard feature of Java EE 6 and later, but can also be used with Java SE applications.

**Prerequisites**

All attendees must have substantial prior Java development and relational database experience.

**Materials**

All JPA training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* A recent version of Windows, macOS, or Linux with at least 8 GB RAM
* JDK 8 or later
* The Java tool the students are likely to use after the class (Eclipse or IntelliJ IDEA are recommended)
* Other free software - please contact us if you have purchased this class

**Objectives**

* Understand the JPA architecture
* Create JPA based applications
* Understand and use JPA to map persistent objects to the database
* Create JPA persistence units, and understand JPA persistence contexts
* Use the JPA EntityManager
* Work with queries and JPQL (Java Persistence Query Language), as well as the Criteria API (JPA 2)
* Understand and work with collections and associations
  + Value and Entity Types
  + Bidrectional and unidirectional
  + 1-1, 1-N, N-N
* Use versioning support for optimistic locking
* Map inheritance hierarchies using JPA
* Performance tune your JPA applications
* Understanding JPA transaction support
* Understand the relationship between JPA / EJB3
* Use JPA entities from session beans
* Be familiar with Spring support for JPA

**Outline**

* Introduction to Java Persistence API (JPA)
  + 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
  + Logging Options (Provider based)
* Updates and Queries
  + Inserting and Updating - Persisting new Entities, Updating an Instance, Removing an Instance
  + 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 query, Ordering, Aggregate Query, Build Update and Delete
  + Embedded Objects
    - @Embeddable, @Embedded
    - Defining and using Embedded Objects
  + Compound Primary Keys - @EmbeddedID, @IDClass, Defining Compound Keys
* The Persistence Lifecycle
  + Transaction Overview and Transactions in JPA
    - Transaction Overview
    - EntityTransaction API (including JTA and resource-local EntityManager)
  + 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, @Entity Listeners
* Entity 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, Table Structures
    - Relationship Inverse - Owning Side
    - Collection Types (List, Set, etc)
    - Cascading Over Relationships (including orphanRemoval - JPA 2)
    - @ManyToMany, @OneToOne
    - Lazy and Eager Loading
    - Queries Across Relationships (Inner Joins, Outer Joins, Fetch Joins)
  + Entity Inheritance Mapping
    - Overview
    - Single Table Mapping
    - Joined (Table per Subclass) Mapping
    - Table per Concrete Class Mapping
    - Pros and Cons
* Element Collections (JPA 2)
  + Overview, Collections of Value Objects, @ElementCollection, @CollectionTable
  + Using Element Collections
  + Collections of Embeddables
* Criteria API (JPA 2)
  + Overview of the Criteria API
  + Path Expressions, Building Queries (CriteriaBuilder, CriteriaQuery, Subquery, Predicate, Expression, Order, Selection, Join)
  + Executing Queries and Accessing Results
* Additional JPA Capabilities
  + XML Mapping Files
  + Bean Validation (JPA 2)
  + Best Practices
    - Primary Keys, Named Queries, Lazy/Eager Loading, Transactional Semantics, Encapsulation, Report Queries
* Integration
  + Data Access Objects (DAO) and Java SE Integration
    - DAO Overview
    - JpaUtil Class for EntityManager management in Java SE
    - Lifecycle Considerations
  + Integration with EJB
    - Using JPA with Session Beans
    - Container Managed (Injected) Entity Manger
    - JTA Transactions and Lifecycle Considerations
    - Extended Persistence Contexts
  + Using JPA with Java Web Apps
    - Using EntityManager in Web apps - request scoping
    - Lazy Loading - Open EntityManager in View Pattern
  + Integration with Spring
    - Injection of EntityManger, EntityManagerFactory
    - LocalEntityManager FactoryBean
    - JPA/Spring Based DAO
* Conclusion