

Lot 5 : Advanced modules of Jakarta EE stack

Training @FPS on November 16th 2020

Adjété Tata DA SILVEIRA

Software engineer since 2011



LinkedIn : <https://www.linkedin.com/in/adjetetatadasilveira/>

Github : @ghostGuiggs

<https://www.hackages.io/>

Who are you ?

What do you do at SFPD ?

What do you expect from this training ?

What you can expect from this session

1. Up to date relevant knowledge, concepts
2. Hands-on and challenge driven-training
3. Pair programming
4. Sharing
5. Answers to Software Engineering Questions

Agenda

1. Enterprise Edition for Java Developers
2. Hibernate
3. JAXB
4. CDI
5. Questions

Technical Environment

- Git (Github Desktop)
- IDE IntelliJ, Eclipse, NetBeans
- Maven
- JDK 14
- Docker Desktop
- Terminal
 - Cygwin64
 - ConEmu
- Stay hydrated

Workshops: Fork the Github repository

<https://github.com/hackages/sfpd.training.lot5>

Default branch : starting-point

git remote add upstream <https://github.com/hackages/sfpd.training.lot5.git>

Keep the repo synced

<https://docs.github.com/en/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/syncing-a-fork>

Enterprise Edition for Java Developers

Introduction

- Java EE is a toolbox which contains API specifications
- Meet the demanding expectations of modern information systems
- APIs governed by the Java Community Process
 - <https://jcp.org/en/home/index>
 - JSR : Java Specification Request
- Used to be spearheaded by Oracle, now Eclipse Foundation

Java EE 8



Batch	Dependency Injection	JACC	JAXR	JSTL	Management
Bean Validation	Deployment	JASPIC	JMS	JTA	Servlet
CDI	EJB	JAX-RPC	JSF	JPA	Web Services
Common Annotations	EL	JAX-RS	JSON-P	JavaMail	Web Services Metadata
Concurrency EE	Interceptors	JAX-WS	JSP	Managed Beans	WebSocket
Connector	JSP Debugging	JAXB			
JSON-B	Security				

Credit : <https://blogs.oracle.com/java/java-ee-8-overview>

Jakarta EE Goals

- Faster and less costly applications development
- Cloud and MicroServices environments
- Support for HTML5 and HTTP2
- Annotation based configuration
- Dependency Injection for object creation
- Roughly 35 APIs included in Java Enterprise Edition : Jakarta EE

Hibernate

Agenda

1. Java Persistence API Specification
2. Hibernate
 - a. Relational Mapping
 - b. Hibernate Query Language
 - c. Criteria

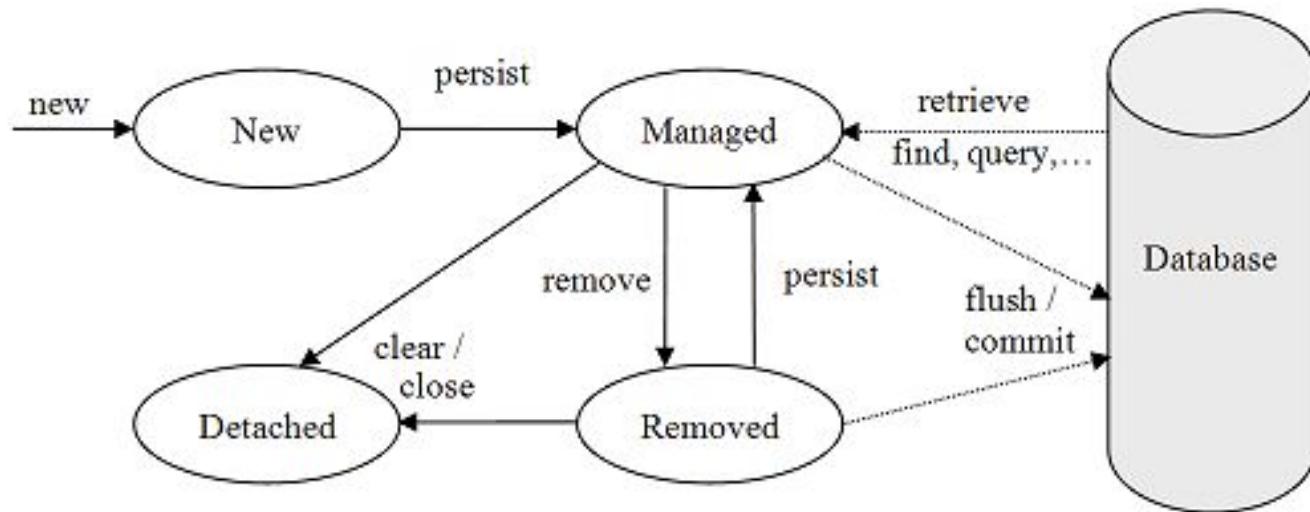
JPA API

- Supports object-relational mapping for Java applications with relational data
- Interact with a relational database without writing any SQL
 - JPQL
 - Criteria API
- EntityManager

Data Persistence

- JPA works with existing Java object model : Entity
- Represents a table within a database schema
- Entity objects lifecycle consists of 3 states

Data Persistence



Credit : <https://www.objectdb.com/java/jpa/persistence/managed>

Data Persistence : Transient state

- Objects in that state exist in memory
- DB has no knowledge of the object yet
- To persist this type, the session save method is invoked
- Then the object is assigned an identifier and no more in transient state

Data Persistence : Persistent state

- Objects in that state exist in DB
- Changes made to object are reflected in current transaction

Data Persistence : Detached state

- Objects in that state exist in DB but EntityManager lost its connection to them
- Persistent object whose session has been closed
- Can be reattached to session when needed

Basic Annotations

- `@Entity`
- `@Table`
- `@Column`
- `@Id`
- `@GeneratedValue`
- `@Transient`

JPA Sample

Introduction to Hibernate

- Provided by JBoss
- Current version 5
- Framework designed to simplify DB access
- Data is represented as POJO
- A session manager is used to access these objects

Introduction to Hibernate

- Goal : Increased productivity when data persistency is needed
- Makes persistent database access easier
- Open Source under the GNU license

Introduction to Hibernate

- Relies on Mapping and Configuration
- Database schema details are hidden by Hibernate
- Provides more time to concentrate on business logic
- Includes query and retrieval operations

Introduction to Hibernate

- Query results are objects instead of SQL result set
- Persistence is achieved by mapping objects to tables
- Provides APIs for restoring and retrieving POJO to and from DB
- DB changes only require changes to XML mapping
- Support most major RDBMS

Hibernate vs. JDBC comparison

TOOL	HIBERNATE JPA IMPLEMENTATION	JDBC
Purpose	Object-relational mapping (ORM)	Database Connectivity
Query Language	Hibernate Query Language (HQL) and the Java Persistence Query Language (JPQL)	Structured Query Language (SQL)
License	LGPL 2.1 and ASL 2.0	Oracle license
Packaging	Standalone JAR file along with dependent libraries	Part of Java SE
Caching	Built-in second-level caching	No caching
Competitors	TopLink, EclipseLink, OpenJPA and ActiveJDBC	ODBC, JDO
Maintainer	JBoss by Red Hat	Oracle
Release	Initial Hibernate release in 2001; JPA 1.0 released in 2006	Released in 1997 as part of Java 1.1

SessionFactory

- DB connections are done via core Hibernate classes (under the hood)
- A connection is opened when needed
- Each application should have one SessionFactory instance
- Threads service and client requests obtain session instances from this factory

SessionFactory

- ThreadSafe object
- Created at startup
- One SessionFactory per DB when multiple DB are concerned

Session Object

- Object used to get a physical connection with the DB
- Should be created and destroyed when an interaction is needed with the DB
- It wraps the JDBC resources
- Handles operations for instances of mapped entity classes using methods :
save , persist, get, create, read, openSession ...

Transaction Object

- Unit of work guaranteed to behave as if you have exclusive use of DB
- Captures the data at one point in time
- Can be started, committed, rolled back
- In Hibernate exists an underlying Transaction manager

HIBERNATE VERSUS JPA

Hibernate is an object-relational mapping framework that deals with data persistence.	The Java Persistence API is a Java specification for managing relational data in Java applications.
Hibernate is one the most advanced and popular JPA implementation providers.	JPA is only a specification that doesn't provide any implementation classes.
Session is the Hibernate specific API to handle persistence in applications.	EntityManager is the standard for JPA specification implementation.
It is an open-source Object relational mapping tool that simplifies the development of Java applications to make connecting to databases much easier than ever.	It is the standard API for persistence and object relational mapping that allows developers to perform database operations much efficiently.
It uses its own query language called HQL (Hibernate Query Language).	It uses a platform-independent object-oriented query language called the JPQL (Java Persistence Query Language).

Difference Between .net

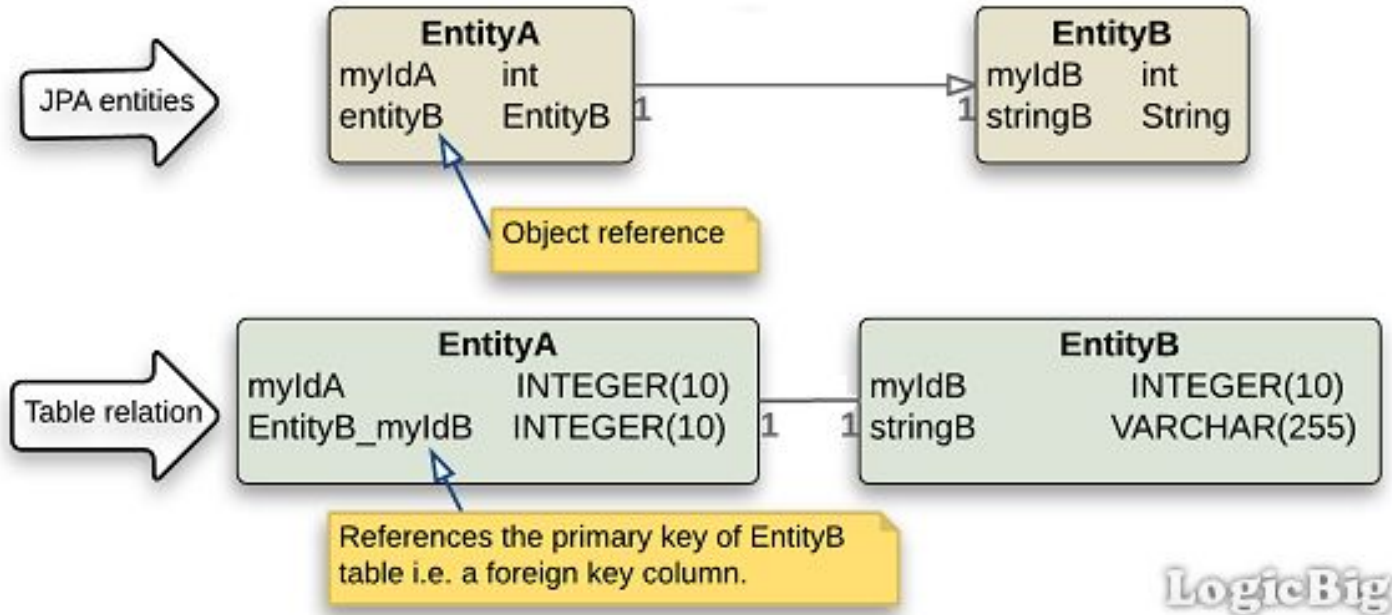
<http://www.differencebetween.net/technology/difference-between-hibernate-and-jpa/>

Hibernate Samples

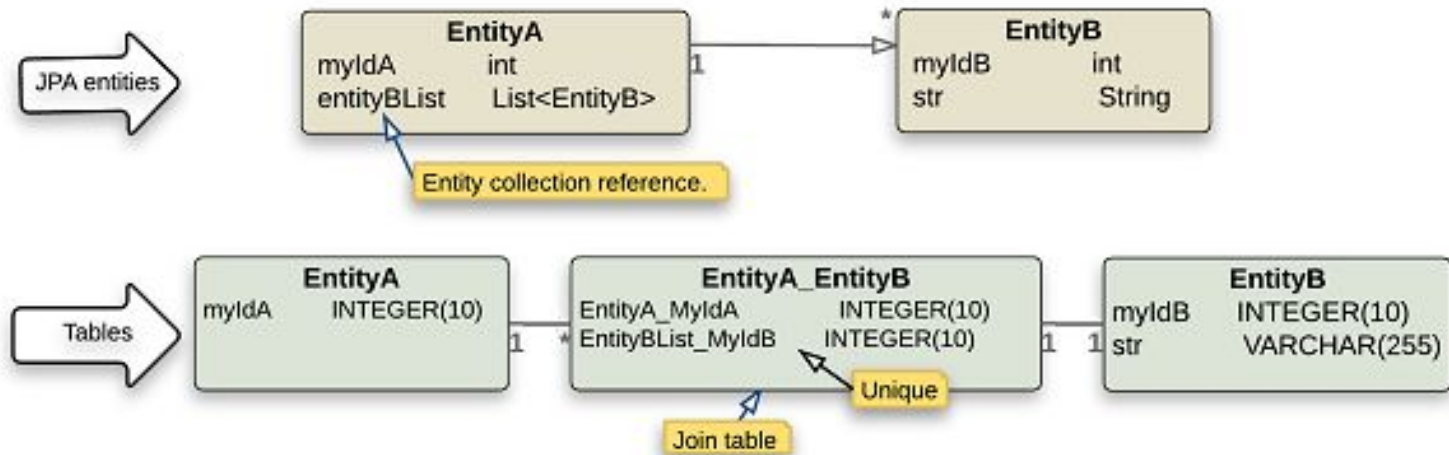
Relationships

- In the object model, entity associations map relationships between entities
- Cardinality : one-to-one, one-to-many, many-to-many
- Direction : Uni or Bidirectional
- Cascading : An action on one entity cascades to its associated entities
- Fetching : Eager or Lazily

One-to-One Entity Relationship

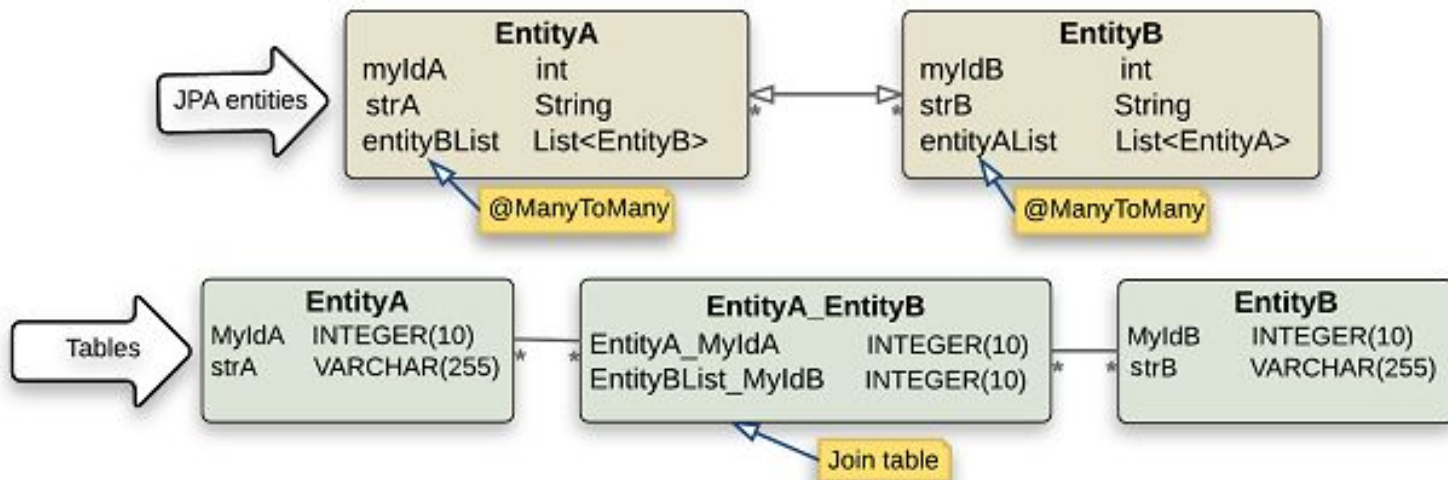


One-to-Many Relationship



LogicBig

Many-to-Many Bidirectional relationship



LogicBig

HQL

- An alternative to SQL, centered on Entities
- Object-oriented version of SQL
- No reference to “tables” or “fields” referenced by Classes and Attributes
- Adds a layer of abstraction for the programmer
- HQL Queries are translated to conventional SQL queries
- Helps avoid portability hassles

Workshop