JAVA PERSISTENCE API

JPA

Especificación ORM

Persistencia en Bases de Datos

EMPEZAR

DEFINICIÓN

JPA es una implementación para frameworks Object Relational Mapping (ORM), que permite interactuar con una base de datos por medio de objetos.

JPA es el encargado de convertir los objetos Java en instrucciones para el gestor de base de datos.

Regresar

CARACTERÍSTICAS

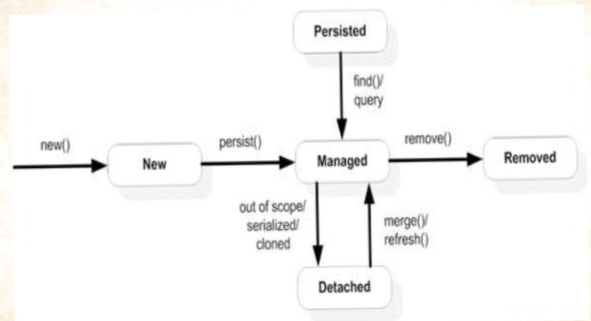
JPA

Java Persistence API

HIBERNATE

```java
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Department
```



1

Persistencia

Mantener los datos para luego utilizarlos

2

Identidad

Cada objeto relacionado con un registro en la BD

3

Transaccionalidad

INSERT, UPDATE, DELETE

Regresar

VENTAJAS

- El desarrollo más rápido.
- Utilizamos Entidades en lugar de SQL nativo (No siempre).
- 100% orientado a objetos.
- Menor cantidad de errores.
- Permite un mejor mantenimiento en las aplicaciones.

Regresar

```java
/**
package com.devpredator.jpaexample.entity;

import java.time.LocalDateTime;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

/**
 * @author DevPredator
 *
 */
@Entity
@Table(name = "genero")
public class Genero {
```

ENTITY

Un Entity de JPA, representa una tabla en la base de datos.

Cada instancia de un Entity, representa una fila en la tabla.

Regresar

JPQL

Se utiliza para crear consultas contra entidades hacia una base de datos relacional.

Regresar

```java
@Override
public List<Genero> consultar() {
    EntityManager em = ENTITY_MANAGER_FACTORY.createEntityManager();

    TypedQuery<Genero> query = em.createQuery("FROM Genero", Genero.class);

    List<Genero> generos = null;

    try {
        generos = query.getResultList();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        em.close();
    }

    return generos;
}
```

Regresar

Son implementaciones de JPA

POM.XML

```
dencies>
dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>4.11</version>
   <scope>test</scope>
/dependency>

-- https://mvnrepository.com/artifact/org.hibernate/hibernate-cor
dependency>
   <groupId>org.hibernate</groupId>
   <artifactId>hibernate-core</artifactId>
   <version>5.4.22.Final</version>
/dependency>

-- https://mvnrepository.com/artifact/mysql/mysql-connector-java
dependency>
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
   <version>8.0.21</version>
/dependency>


endencies>
```

PERSISTENCE.XML

```
<persistence version="2.0"
    xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">

    <!-- Define a name used to get an entity manager. Define that you will
    complete transactions with the DB  -->
    <persistence-unit name="persistenceMySQL" transaction-type="RESOURCE_LOCAL">

        <!-- Define the class for Hibernate which implements JPA -->
        <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
        <!-- Define the object that should be persisted in the database -->
        <class>com.devpredator.jpaexample.entity.Genero</class>
        <properties>
            <!-- Driver for DB database -->
            <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver" />
            <!-- URL for DB -->
            <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/tienda_musical?useUnicode=true&a
            <!-- Username -->
            <property name="javax.persistence.jdbc.user" value="root" />
            <!-- Password -->
            <property name="javax.persistence.jdbc.password" value="root" />

                    <!--Hibernate properties-->
            <property name="hibernate.show_sql" value="false"/>
            <property name="hibernate.format_sql" value="false"/>
            <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL5Dialect"/>
        </properties>
    </persistence-unit>
</persistence>
```

```java
public class GeneroDAOImpl implements GeneroDAO {

    private static final EntityManagerFactory ENTITY_MANAGER_FACTORY = Persistence
            .createEntityManagerFactory("persistenceMySQL");

    @Override
    public void guardar(Genero genero) {
        EntityManager em = ENTITY_MANAGER_FACTORY.createEntityManager();

        EntityTransaction et = em.getTransaction();
        et.begin();

        try {
            em.persist(genero);
            et.commit();
        } catch (Exception e) {
            // If there is an exception rollback changes
            if (et != null) {
                et.rollback();
            }
            e.printStackTrace();
        } finally {
            em.close();
        }
    }
```

Sincroniza transacciones de un entity con la BD