
Hibernate

— Primers programes —

Llibreries necessàries

- Amb la configuració realitzada fins ara no es pot programar en Java, és necessari afegir una sèrie de paquets al projecte. Per a això:
 - Accedim a <http://hibernate.org/orm/downloads/> i ens descarreguem l'última distribució estable de **Hibernate**. Descomprimim el zip i ens quedarà una carpeta amb un nom semblant a **hibernate-release-<version>**.
 - Creem una carpeta (en l'escriptori, per exemple) que anomenarem **Hibernate**. Ens col·loquem en la carpeta **hibernate-release-<version>/lib/provided** i copiem tots els arxius a la carpeta recentment creada en l'escriptori.
 - Descarreguem l'última versió de **slf4j** des de <https://www.slf4j.org/download.html>.
 - Descomprimim i localitzem els fitxers **slf4j-api-<version>.jar** i **slf4j-simple-<version>.jar**; i els copiem en la carpeta que hem anomenat **Hibernate**.

Llibreries necessàries



- Des d'Eclipse fem clic amb el botó dret del ratolí en el nostre projecte i premem **Build Path → Add Libraries → User Library**. Premem “**User Libraries**” i després “**New**”.
- Ens demanarà un nom per a la llibreria que volem agregar. Escrivim per exemple: **Hibernate.lib**.
- Premem el botó “**Add External JARS**”. Seleccionem tots els fitxers de la carpeta **Hibernate**.
- Ara ja estem en condicions de poder treballar.
- En primer lloc necessitarem crear un **Singleton**.
- El **Singleton** és un patró dissenyat per a restringir la creació d'objectes pertanyents a un classe. La seva intenció consisteix a garantir que una classe només tingui una instància i proporcionar un punt d'accés global a ella.
- El nostre **Singleton** serà una classe d'ajuda que accedeix a **SessionFactory** per a obtenir un objecte de sessió, hi ha una única **SessionFactory** per a tota l'aplicació. El nom de la classe és **HibernateUtil.java** i s'inclourà en el paquet primer del nostre projecte. El codi és:

Llibreries necessàries



```
package primer;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {

    //Per assegurar que la classe no puga ser instanciada novament, es regula l'abast del constructor amb l'atribut
    //private
    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try{
            return new Configuration().configure().buildSessionFactory(new
                StandardServiceRegistryBuilder().configure().build());
        }
        catch (Throwable ex) {
            System.err.println("Initial SessionFactory creation failed."+ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() { return sessionFactory; }
}
```

Llibreries necessàries

- Amb la classe anterior podem obtenir la sessió actual des de qualsevol part del nostre projecte.
- Ara afegim al nostre projecte una nova classe que anomenarem `Main.java`. Aquesta es crearà en el `default package` del projecte. Aquesta classe inserirà una fila en la taula `depart`.

Llibreries necessàries



```
import primer.*;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
public class Main {
    public static void main(String[] args){
        //En primer lloc s'obté la sessió creada pel Singleton.
        SessionFactory session = HibernateUtil.getSessionFactory();
        //Obrim la sessió i iniciem una transacció
        Session session = session.openSession(); Transaction tx = session.beginTransaction();
        System.out.println("Inserim una fila en depart");
        //Creem un nou objete Depart i donem valor als seus atributs
        Depart dep = new Depart();
        dep.setDeptNo((byte) 62); dep.setDnombre("MARKETING"); dep.setLoc("GUADALAJARA");
        //Desem a la base de dades i comprometem la informació
        session.save(dep);
        tx.commit();
        session.close(); System.exit(0);
    }
}
```

Activitat

Executa l'exemple exposat anteriorment i comprova que afegeix una nova fila en `depart`.

A1.- Realitza un programa Java (pots modificar el `Main` anterior) que afegeixi una nova fila en la taula `emple`. El nou empleat estarà assignat al departament 10. Per a assignar-li el departament serà necessari crear un objecte de tipus `Depart` i assignar com a número de departament el valor 10 amb el mètode `setDeptNo()`.

Ajudes web per hibernate:

- <https://www.codejava.net/frameworks/hibernate/hibernate-hello-world-tutorial-for-beginners-with-eclipse-and-mysql>
- <https://docs.jboss.org/hibernate/orm/3.5/reference/es-ES/html/queryhql.html>
- <https://datos.codeandcoke.com/apuntes:hibernate>
- <https://www.adictosaltrabajo.com/2007/06/13/hibernate-tools/>